



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

**Grau en Enginyeria Electrònica Industrial i Automàtica**

**EDITOR DE GRAFCETS**



**Memòria**

**Autor:** Pablo Mañas Pellejero  
**Director:** Joan Domingo Peña  
**Convocatòria:** Maig 2019





## Resum

Aquesta memòria tracta sobre la realització d'un programa informàtic que permet la creació de GRAFCETs juntament amb la seva edició, simulació i l'exportació en diferents formats. També es dona informació d'aquests models de representació gràfics amb la fi d'explicar com funcionen i perquè són utilitzats i a més es mostra l'estat actual dels programes que permeten l'edició dels GRAFCETs amb l'ajuda de diferents exemples juntament amb les seves característiques. Per tal d'explicar el funcionament del programa desenvolupat, s'afegeix al document un apartat que mostra els diferents elements d'aquest a mode d'aprenentatge.

Cal dir que per aconseguir l'objectiu principal s'ha utilitzat l'entorn de desenvolupament anomenat Microsoft Visual Studio 2017 i el llenguatge de programació empleat ha sigut el Visual Basic. Per explicar com funciona el programa desenvolupat s'afegeix un apartat amb aquesta informació detallada. La metodologia emprada a l'hora de programar ha sigut en forma d'espiral al voltant de les activitats de: determinació d'objectius, anàlisi de riscos, desenvolupament i prova, i, per últim, planificació.

Com a resultat s'ha obtingut un programa funcional que permet realitzar els diagrames corresponents amb unes característiques concretes que es poden observar. Com a conclusió, encara que algunes de les qualitats del programa (interaccions i disseny) es poden millorar, el programa final compleix els requisits que s'esperen d'un editor de GRAFCETs.

## Resumen

Esta memoria trata sobre la realización de un programa informático que permite la creación de GRAFCETs junto con su edición, simulación y la exportación en diferentes formatos. También se da información de estos modelos de representación gráficos con el fin de explicar cómo funcionan y porqué son utilizados y además se muestra el estado actual de los programas que permiten la edición de los GRAFCETs con la ayuda de diferentes ejemplos junto con sus características. Para explicar el funcionamiento del programa desarrollado, se añade al documento un apartado que muestra los diferentes elementos de éste a modo de aprendizaje.

Cabe decir que para conseguir el objetivo principal se ha utilizado el entorno de desarrollo llamado Microsoft Visual Studio 2017 y el lenguaje de programación empleado ha sido el Visual Basic. Para explicar cómo funciona el programa desarrollado se añade un apartado con esta información detallada. La metodología empleada a la hora de programar ha sido en forma de espiral alrededor de las actividades de: determinación de objetivos, análisis de riesgos, desarrollo y prueba, y, por último, planificación.

Como resultado se ha obtenido un programa funcional que permite realizar los diagramas correspondientes con unas características concretas que se pueden observar. Como conclusión, aunque algunas de las cualidades del programa (interacciones y diseño) se pueden mejorar, el programa final cumple los requisitos que se esperan de un editor de GRAFCETs.

## Abstract

This report is about the realization of a computer program that allows the creation of GRAFCETs along with its edition, its simulation and its export in different formats. Information about these graphic representation models is also given in order to explain how they work and why they are used. The current status of the programs that allow the edition of the GRAFCETs with different examples together with their characteristics is also shown. To explain the operation of the developed program, a section is added to this document that shows the different elements of it as a tutorial.

It must be said that to achieve the main objective, the development environment called Microsoft Visual Studio 2017 has been used and the programming language used has been Visual Basic. To explain how the developed program works, there is a section with this detailed information. The methodology used at the time of programming has been in a spiral around the activities of: determination of objectives, risk analysis, development and testing, and, finally, planning.

As a result, a functional program has been obtained that allows the corresponding diagrams to be made with specific characteristics that can be observed. As a conclusion, although some of the qualities of the program (interaction and design) can be improved, the final program meets the requirements that are expected of a GRAFCET editor.

## Glossari

GRAFCET: *Grphe Fonctionnel de Commande Etape Transition* (en francès). Traduït com diagrama de control amb etapes i transicions.



# Índex

<b>RESUM</b>	<b>2</b>
<b>RESUMEN</b>	<b>3</b>
<b>ABSTRACT</b>	<b>4</b>
<b>GLOSSARI</b>	<b>5</b>
<b>1. INTRODUCCIÓ</b>	<b>9</b>
1.1. Objectius del treball.....	9
1.2. Abast del treball.....	9
1.3. Motivació .....	10
<b>2. ¿QUÈ ÉS UN GRAFCET?</b>	<b>11</b>
2.1. Elements d'un GRAFCET .....	13
2.1.1. Etapes.....	13
2.1.2. Transicions .....	14
2.1.3. Línies d'evolució.....	15
2.2. Estructures bàsiques d'un GRAFCET .....	16
2.2.1. Seqüència lineal .....	16
2.2.2. Convergència i divergència en "O" .....	17
2.2.3. Convergència i divergència en "Y" .....	20
2.3. Macroetapes.....	21
<b>3. EDITOR DE GRAFCETS – PABLO MAÑAS</b>	<b>22</b>
3.1. Inici del programa .....	22
3.2. Barra d'elements .....	29
3.3. Esdeveniments de teclat, ratolí i edició d'elements .....	34
3.4. Enllaços .....	40
3.5. Simulació.....	42
3.6. Exportació en Excel.....	59
3.7. Funcionament de l'aplicació a nivell de programació.....	61
3.7.1. Menú Principal .....	61
3.7.2. Nou Document.....	108
3.7.3. Transicions .....	110
3.7.4. Accions .....	113
3.7.5. Accions associades.....	114



3.7.6.	Canviar etiquetes etapes.....	114
3.7.7.	Canviar etiquetes.....	115
3.7.8.	Canviar etiquetes marques .....	115
3.7.9.	Convergències i Divergències.....	116
3.7.10.	Enllaços .....	116
3.7.11.	Etiquetes.....	117
3.7.12.	Línies .....	117
3.7.13.	Simulació.....	119
3.7.14.	Temporitzador Classe .....	134
3.7.15.	Temporitzador Form .....	134
3.7.16.	Tipus d'enllaç .....	138
3.8.	Millores al programa.....	139
<b>4.</b>	<b>ESTAT DE L'ART</b> .....	<b>141</b>
4.1.	Dia 0.97.2 .....	141
4.2.	CADeSIMU v3 .....	145
4.3.	Omegon Fluid Technology Grafcet 2.0.9.2 .....	149
4.4.	Taula comparativa.....	154
<b>5.</b>	<b>METODOLOGIA DE TREBALL</b> .....	<b>158</b>
	<b>CONCLUSIONS</b> .....	<b>163</b>
	<b>PRESSUPOST I/O ANÀLISI ECONÒMICA</b> .....	<b>165</b>
	<b>BIBLIOGRAFIA</b> .....	<b>167</b>
	Recursos utilitzats per la programació: .....	167
	Recursos utilitzats per la part teòrica:.....	168
	Recursos utilitzats per l'estudi de l'art: .....	168



## 1. Introducció

Per iniciar el treball de fi de grau és necessari explicar quins són els objectius principals d'aquest, quins són els conjunts de temes sobre els quals es vol informar, és a dir, l'abast del propi i quina és la motivació a dur aquest treball. Per tant, a continuació es mostren amb més detall aquests conceptes.

### 1.1. Objectius del treball

El principal objectiu del treball és la realització d'un programa informàtic que permet la creació, edició, simulació i exportació en diferents formats (Excel i PDF) de GRAFCETS amb la fi de crear una eina útil i fàcilment utilitzable per a l'usuari interessat en aquest àmbit. Com és evident aquest objectiu relaciona conceptes de programació amb idees bàsiques del control automàtic. Per tal de que el lector i l'usuari sigui conscient de com funciona el programa realitzat, també es considera com a objectiu principal l'explicació dels diferents elements que componen el programa. Relacionat amb aquest concepte, és necessari afegir com funciona detalladament el codi de l'aplicació genera ja que es considera com a objectiu del treball explicar la manera en que s'han elaborat les diferents accions que permet el programa amb la fi que es puguin veure reflexades les idees en quant a programació. A més, a la part de l'annex es troba tot el codi font per si es vol veure minuciosament tot.

Altre objectiu menys important que es persegueix aquesta memòria és donar informació bàsica sobre els GRAFCETS que tant s'utilitzen al món de l'automàtica per tal d'explicar al lector com és el seu funcionament i per a què serveixen. També, es cerca estudiar diferents tipus d'editors de GRAFCETS ja existents al mercat, explicar les seves característiques i comparar-los entre sí. Per últim es vol mostrar la metodologia emprada a l'hora de fer el treball.

### 1.2. Abast del treball

Tot i que ja es menciona anteriorment en gran part a l'apartat anterior, l'abast del treball engloba tota la part de programació necessària per a la realització de l'editor en qüestió, juntament amb l'estudi dels diferents programes ja existents al mercat, la cerca d'informació sobre els GRAFCETS, l'exposició d'aquesta de forma detallada i la metodologia emprada en l'elaboració del treball.

### 1.3. Motivació

Per tal de justificar la raó per la qual s'ha decidit elaborar un editor de GRAFCETs , en aquest subapartat s'exposen els arguments de l'autor. El principal motiu ha sigut l'interès d'aprendre a programar de forma correcta i adequada i el fet de realitzar un programa com aquest ajuda en molts sentits. Altre raó és l'impuls de voler fer una eina útil per a que altres persones puguin emprar-la de forma fàcil en la seva vida d'estudiant o laboral. Per últim, la inclinació de l'estudiant per la part de control i automàtica també ha ajudat a la preparació d'aquest treball.

## 2. ¿Què és un GRAFCET?

Abans de resoldre la pregunta formulada al títol del capítol cal explicar primerament en què consisteix l'automatització de processos. S'entén com a procés a una successió de fases que transcorren en una operació industrial i l'automatització d'aquest permet que es faci de forma automàtica amb ajuda d'un automatisme que duu a terme una sèrie d'operacions o accions a partir d'unes expressions lògiques o entrades. Això és important, ja que un GRAFCET és la representació gràfica, millor dit, un diagrama funcional d'una sèrie d'activitats i operacions que es realitzen de forma seqüencial i, per tant, permet descriure el procés a automatitzar. Es considera doncs que el procés a automatitzar i l'automatisme són un mateix sistema i s'indiquen les accions que s'han de realitzar en aquest procediment en conjunt amb la informació de com dur-les a terme.

El GRAFCET (*GRA*phique *fon*ctionnel de *com*mande *É*tapes / *T*ransitions o gràfic funcional de control d'etapes i transicions) va tenir el seu origen a França el 1975 i es va publicar per primera vegada el 1977; va ser el fruit del treball de la AFCET (*Ass*ociation *Fra*nçaise pour la *Cy*bernétique *E*conomique et *T*echnique, Associació Francesa per la Cibernètica Econòmica i Tècnica, creada el 1975) en la seva tasca de trobar una eina útil per a la descripció d'automatismes. Durant un temps, GRAFCET va significar "gràfic de AFCET". La primera vegada que es va convertir en norma va ser en 1982, a França, sota l'agència francesa per a la normalització, AFNOR, com a norma NF C 03-190.

El 1988 la IEC (*Int*ernational *Ele*ctrotechnical *Com*mission, -organització per a la normalització, fundada el 1906, el seu primer president va ser Lord Kelvin-) el va declarar norma internacional amb el número IEC 848 i el va descriure com SFC, *Seq*quential *F*unction *Ch*art, Gràfic Funcional Seqüencial i no es va reconèixer el nom GRAFCET perquè les traduccions podien donar lloc a ambigüitats i a interpretacions errònies. El 1993 els conceptes del GRAFCET sota el nom de SFC es van incorporar com a part de la norma IEC 61131-3, l'última edició (la tercera) és de febrer de 2013. El 2002 hi va haver una revisió de la IEC 848 donant lloc a la IEC 60.848. La versió actualitzada és la IEC 60.848:2013 que defineix al GRAFCET com un llenguatge per especificar la part seqüencial d'un sistema de control. Especifica els símbols i regles de representació així com els tipus de variables (internes, entrada i sortida). El que especifica la norma IEC 60.848 difereix del llenguatge del llenguatge que descriu la norma IEC 61131-3, la SFC, en aspectes sintàctics i semàntics i en l'enfocament.

En qualsevol cas, el GRAFCET és una eina per a la descripció gràfica de seqüències, pensat especialment per al cas de la seqüència que ha de seguir el controlador d'un sistema automàtic. Realment, no es tracta d'un llenguatge de programació, encara que existeixen compiladors per a alguns PLC (*Pro*grammable *Lo*gic *Co*ntroller o Controlador lògic programable) que són capaços de traduir un GRAFCET que estigui prou detallat a un programa per a un model de PLC en concret.

No obstant això, no s'ha de considerar el GRAFCET com si fos un llenguatge de programació perquè la norma vigent, la IEC 61131-3 no ho admet com a tal; el descriu com un llenguatge organitzacional o metallenguatge, per descompondre una seqüència en diversos passos perfectament definits i en els elements que fan possible la transició entre ells, de manera que defineixin l'algoritme de control o llei de comandament.

Així doncs, mentre que el GRAFCET defineix en forma de blocs o passos les accions que s'han de realitzar i les condicions de transició entre aquests blocs, la mateixa norma IEC 61131-3, defineix quatre llenguatges de programació i un metallenguatge, la SFC (en el fons, el GRAFCET); els quatre llenguatges són per a escriure programes i el metallenguatge per descriure (veure figura 2.1). Només quan els blocs del GRAFCET estan prou detallats i atomitzats, és quan es pot assimilar a una forma de programar en sí mateixa; però, com la norma declara, aquest és un detall que no el converteix en un llenguatge de programació.

Metallenguatge	Llenguatges
<ul style="list-style-type: none"> <li>GRAFCET (Gràfics funcionals seqüencials)</li> </ul>	<ul style="list-style-type: none"> <li>Diagrama de contactes</li> </ul>
	<ul style="list-style-type: none"> <li>Llista de instruccions</li> </ul>
	<ul style="list-style-type: none"> <li>Blocs funcionals</li> </ul>
	<ul style="list-style-type: none"> <li>Programació estructurada</li> </ul>

**Figura 2.1.** Llenguatges i metallenguatge a la norma EN 61131-3 (Font: Pablo Mañas)

Programar autòmats va començar sent responsabilitat dels automatistes que utilitzaven controladors amb relés electromagnètics i, per tant, persones familiaritzades amb la interpretació d'esquemes elèctrics de control amb relés. Perquè aquestes persones poguessin (cap als anys 70), passar-se al camp dels controladors lògics programables i poguessin programar els esquemes de contactes en comptes de cablejar-los, els primers PLC ja van incorporar llenguatges de programació similars als sistemes cablejats que es van denominar *llenguatge de diagrama de contactes* que, posteriorment, es va passar a denominar *diagrama d'escala* (llenguatge *ladder*, *ladder diagram* o, senzillament, *ladder*) perquè els circuits que es programaven semblaven contactes disposats a escala. Aquest llenguatge en escala és el més antic i el més estès, encara que no l'únic, i recollit per la IEC 61131-3 com un dels quatre llenguatges normalitzats.

Amb el temps s'ha ofert als programadors utilitzar la programació simbòlica mitjançant instruccions mnemotècniques denominades *llista d'instruccions*, similars a les que s'utilitzen per a programar microprocessadors amb llenguatge ensamblador. Aquest tipus de programació era el que s'utilitzava en les consoles més senzilles, que només admetien una representació de text pel seu visualitzador; la impossibilitat o cost d'una representació gràfica dificultava molt la utilització del *ladder*. El conjunt d'instruccions utilitzat per programar un autòmat és generalment bastant extens i exigeix que el programador ho coneixi amb una certa profunditat si no vol haver d'estar consultant el manual de programació a cada moment. Aquest conjunt d'instruccions és similar a la majoria de PLC, encara que no igual, el que en el fons reverteix en la dificultat per transportar un programa d'un PLC d'un fabricant a un altre de diferent fabricant o, fins i tot, entre PLC d'un mateix fabricant. Programar amb *llista d'instruccions* és una fórmula de programació útil, i recollida per la IEC 61131-3 com un altre dels quatre llenguatges normalitzats, però que no està més estesa que la programació gràfica, que és molt més intuïtiva i que, des que s'utilitzen ordinadors com a terminals de programació, no suposa cap problema respecte a la capacitat de representació gràfica.

Una altra possibilitat per programar PLC recollida per la IEC 61131-3 com un altre dels quatre llenguatges normalitzats que, amb l'aparició dels llenguatges estructurats, es va obrir en el seu moment va ser la programació estructurada utilitzant condicions, bucles, etc. L'últim llenguatge que estableix la norma és mitjançant una representació de diagrames de blocs de funcions, com si de logigrames o esquemes de portes lògiques es tractés. Fa possible programar blocs sofisticats com, per exemple, reguladors PID només passant uns paràmetres.

## 2.1. Elements d'un GRAFCET

Ara que ja s'ha explicat en què consisteix i una mica la història dels GRAFCETS es poden donar a conèixer els elements que componen aquests diagrames funcionals en qüestió. Qualsevol GRAFCET es forma amb tres conjunts bàsics de símbols:

- **Etaques.**
- **Transicions.**
- **Línies d'evolució.**

### 2.1.1. Etaques

Les etapes són un conjunt de fases del procés a automatitzar en les quals s'associen accions a realitzar sobre ell mateix. Aquestes accions han de ser molt simples ja que hauran de ser determinades per les entrades i sortides del sistema.

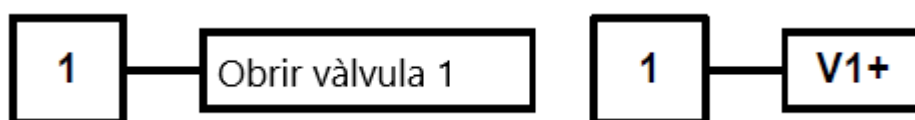
La representació gràfica de les etapes es fa amb un quadrat amb un número interior que l'identifica i, per tant, les diferencia de les altres. En el cas que sigui la primera etapa o, més ben dit, l'etapa inicial en la qual s'iniciarà el sistema per primer cop, la representació es fa amb un quadrat major que a l'interior conté un quadrat menor amb el número d'identificació que sol ser el 0. Ambdues representacions es poden observar a la figura 2.1.1.1.



**Figura 2.2.1.1.** Etapes en un GRAFCET. A l'esquerra etapa inicial (Font: Pablo Mañas)

Les accions associades que s'han mencionat anteriorment es representen per rectangles units a l'etapa en qüestió. Cal dir que a una etapa es poden associar les accions que siguin necessàries. La forma en que es descriuen aquestes es pot realitzar amb dos nivells de complexitat.

- Nivell 1. Es descriu l'acció de forma funcional.
- Nivell 2. Es detalla l'acció merament de forma operativa.



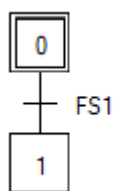
**Figura 2.1.1.2.** Nivells de descripció d'accions i representació d'accions. A la dreta nivell 2 (Font: Pere Ponsa i Ramon Vilanova [14]) (modificada per Pablo Mañas)

Amb la fi de relacionar les accions amb els símbols es necessari realitzar una taula on es detalli cada acció amb la seva descripció. En el cas de la figura anterior la taula hauria d'incloure una columna on l'acció V1+ es relacionés amb Obrir vàlvula 1.

## 2.1.2. Transicions

Les transicions, en resum, permeten la presa de decisions i l'avanç entre les etapes del procés. Es representen mitjançant un línia perpendicular a la línia que uneix dues etapes entre sí. A la dreta de la primera línia esmentada s'escriu la condició que s'haurà de complir per a que l'etapa anterior a aquesta deixi de fer la seva acció o accions i l'etapa següent es posi en funcionament tal i com es mostra a la figura 2.1.2.1. Cal afegir que existeix la possibilitat de que varies etapes s'uneixin a una sola transició.

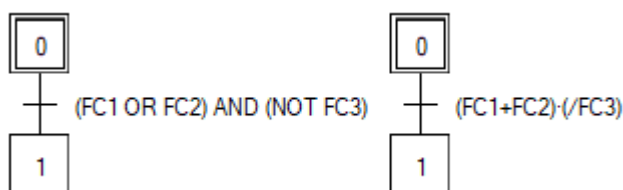




**Figura 2.1.2.1.** Transició entre dues etapes; si l’etapa inicial està activa i es compleix FS1, s’activa l’etapa 1. (Font: Pablo Mañas)

La condició o condicions d’activament depenen de variables externes que poden ser sensors o entrades o variables internes al sistema com comptadors o temporitzadors. Aquestes s’avaluen de forma lògica, és a dir, com a vertaderes o falses i, en funció dels resultats, permeten o no l’activació de les etapes següents. Com en el cas de les accions del subapartat anterior, es poden descriure les condicions, també anomenades com a receptivitats, en els dos nivells de complexitat.

En el cas que la transició depengui d’una expressió lògica entre dues o més variables del sistema, aquestes s’expressen utilitzant les funcions lògiques AND, OR i NOT, que a l’hora de representar-se es simplifiquen pels signes: ·, + i /, respectivament, tal i com es mostra a la figura següent.

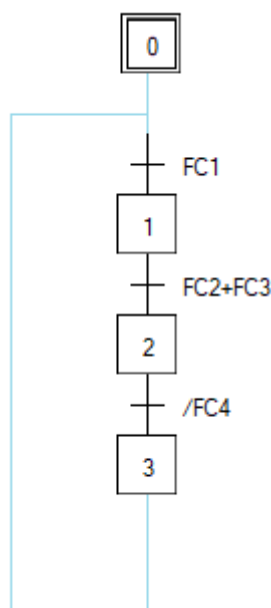


**Figura 2.1.2.2.** Expressió lògica expressada de dues formes; la segona correspondria a la notació adequada (Font: Pablo Mañas)

### 2.1.3. Línies d’evolució

L’últim element principal dels GRAFCETS són les línies d’evolució que permeten l’enllaç entre les etapes i les transicions i viceversa. Mai es poden unir dues transicions o dues etapes entre sí i, per tant, no és necessari l’ús de fletxes per indicar la direcció de les línies tot i que es poden utilitzar com a ajuda.

A la figura que es mostra a continuació (figura 2.1.3.1) es poden observar les diferents línies d’evolució en color blau, en aquest cas, sense fletxes de direccionalitat.



**Figura 2.1.3.1.** Exemple GRAFCET amb línies d'evolució (color blau) (Font: Pablo Mañas)

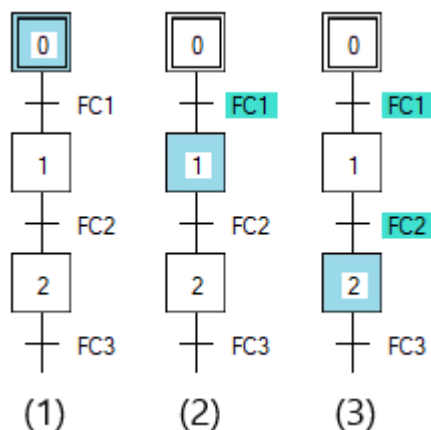
## 2.2. Estructures bàsiques d'un GRAFCET

Ara que ja s'han presentat els elements que componen un GRAFCET, a continuació, es mostren les estructures bàsiques que es poden trobar en els diagrames. Aquestes construccions combinades entre sí podran donar lloc a estructures més complexes i, per tant, a GRAFCETs amb una complexitat elevada. Per tal d'explicar aquest apartat, es mostraran exemples per veure com evolucionen les etapes a través dels tipus de sistemes formats. Aquestes configuracions elementals són les següents:

- **Seqüència lineal.**
- **Convergència i divergència en "O".**
- **Convergència i divergència en "Y".**

### 2.2.1. Seqüència lineal

Es tracta d'encadenar una sèrie d'etapes una rere l'altra amb transicions entre mig que permetin l'avanç del GRAFCET. En aquesta tipologia només pot estar una etapa activa al mateix temps. A la imatge següent (figura 2.2.1.1) es pot veure detalladament aquesta estructura i com avança en funció de l'activació de les transicions. Per fer-ho es mostra l'etapa activa amb un fons blau, tot i que, normalment es fa amb un punt vermell.



**Figura 2.2.1.1.** Exemple seqüència lineal (Font: Pablo Mañas)

Així doncs, tal i com es veu a la figura anterior quan s’inicia el GRAFCET s’activa l’etapa 0 (1). Després, quan la variable FC1 s’activa, l’etapa inicial es desactiva i dona pas a la iniciació de l’etapa1 (2). Per últim, quan FC2 s’activa, l’etapa anterior, o sigui, l’etapa 1 deixa d’estar activa i s’inicia l’etapa 2 (3). Cal afegir que en cada etapa podria haver una acció associada o varies que es durien a terme quan les seves etapes corresponents s’activessin, però al ser un exemple acadèmic no s’ha tingut en compte.

## 2.2.2. Convergència i divergència en “O”

Aquesta estructura es construeix amb la fi de representar tasques alternatives i per això s’anomena amb la conjunció disjuntiva “O”. La tipologia consisteix en una divergència en “O” que parteix inicialment d’una etapa i es divideix en diferents alternatives possibles (poden ser més de dos) les quals solen ser del tipus d’estructura lineal. Per últim, es troba una convergència també del tipus “O” que uneix els diversos camins creats anteriorment per la divergència inicial.

Cadascuna de les alternatives creades per la divergència han de ser excloents, és a dir, no poden haver dos o més camins actius al mateix temps, únicament 1 ha de funcionar. Per ajudar a comprendre com es forma aquesta estructura i com funciona a les següents captures (figura 2.2.2.1, figura 2.2.2.2, figura 2.2.2.3 i figura 2.2.2.4) s’observa un exemple.

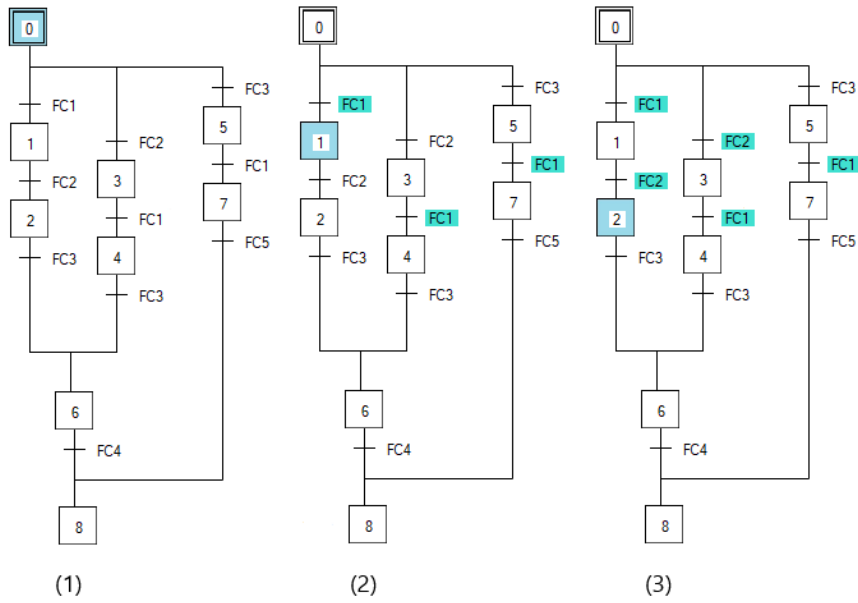


Figura 2.2.2.1. Exemple convergència i divergència en “O” (camí 1). (Font: Pablo Mañas)

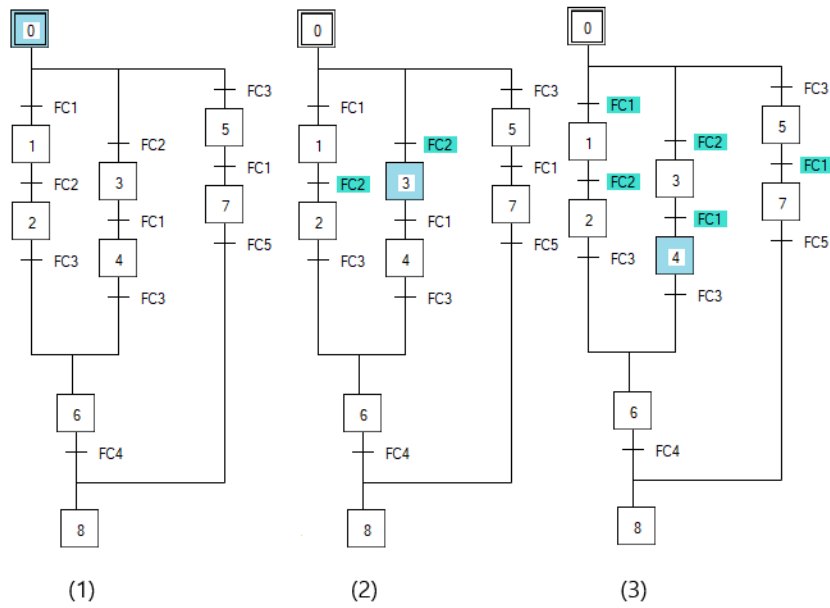


Figura 2.2.2.2. Exemple convergència i divergència en “O” (camí 2). (Font: Pablo Mañas)

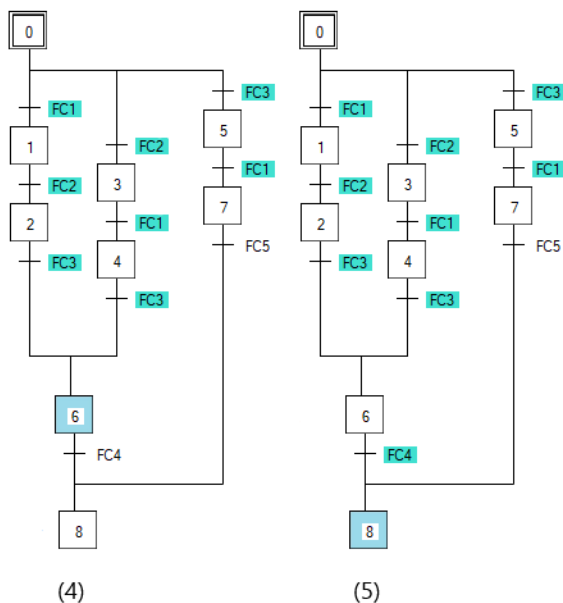


Figura 2.2.2.3. Exemple convergència i divergència en “O” (camí 1 i camí 2). (Font: Pablo Mañas)

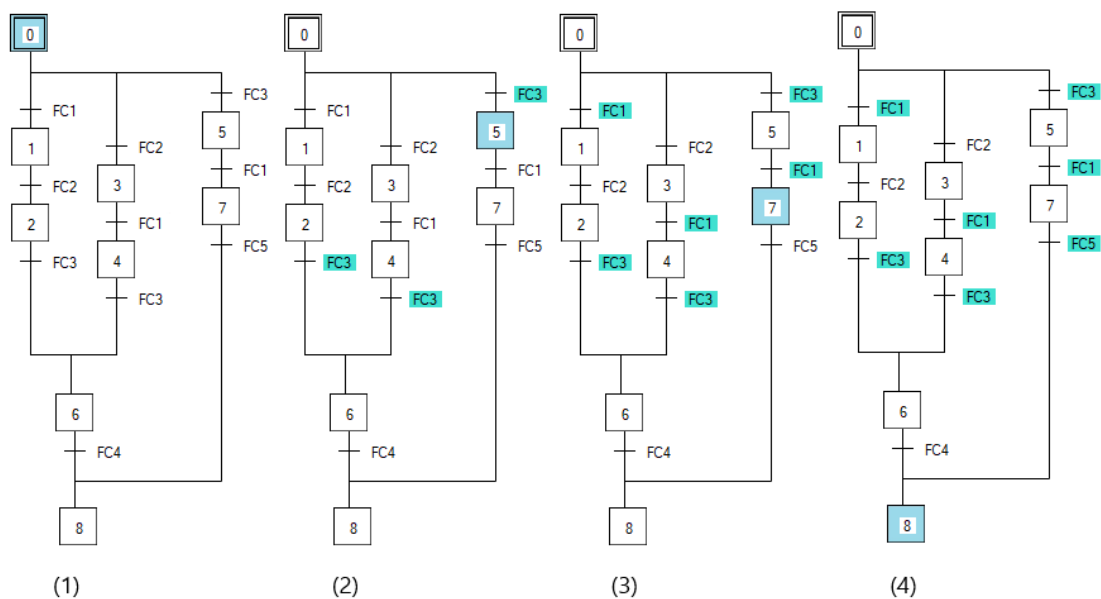


Figura 2.2.2.4. Exemple convergència i divergència en “O” (camí 3). (Font: Pablo Mañas)

Tal i com s’observa a les figures anteriors, en funció de les transicions que s’accionin a la divergència n’hi ha 3 camins possibles. Si després d’iniciar el procés s’activa FC1, l’etapa 1 es posa en marxa i, per tant, la seqüència que segueix es farà seguint les figures 2.2.2.1 i 2.2.2.3. En el cas que s’activi FC2 a l’inici, es segueix el segon camí observable a les figures 2.2.2.2 i 2.2.2.3. Per últim, si inicialment s’activa FC3, la successió adequada es veu a la figura 2.2.2.4. Així doncs, es pot deduir que la divergència en “O” s’inicia a l’etapa 0 amb les transicions FC1, FC2 i FC3. Les seqüències lineals es formen amb les

parelles d'etapes 1 i 2, etapes 3 i 4 i etapes 5 i 7. Pel que fa a la convergència en "O" es poden veure 2. Una d'elles es forma després de les etapes 2 i 4 amb les dues transicions FC3 i l'altra rere les etapes 6 i 7 amb les transicions FC4 i FC5 respectivament.

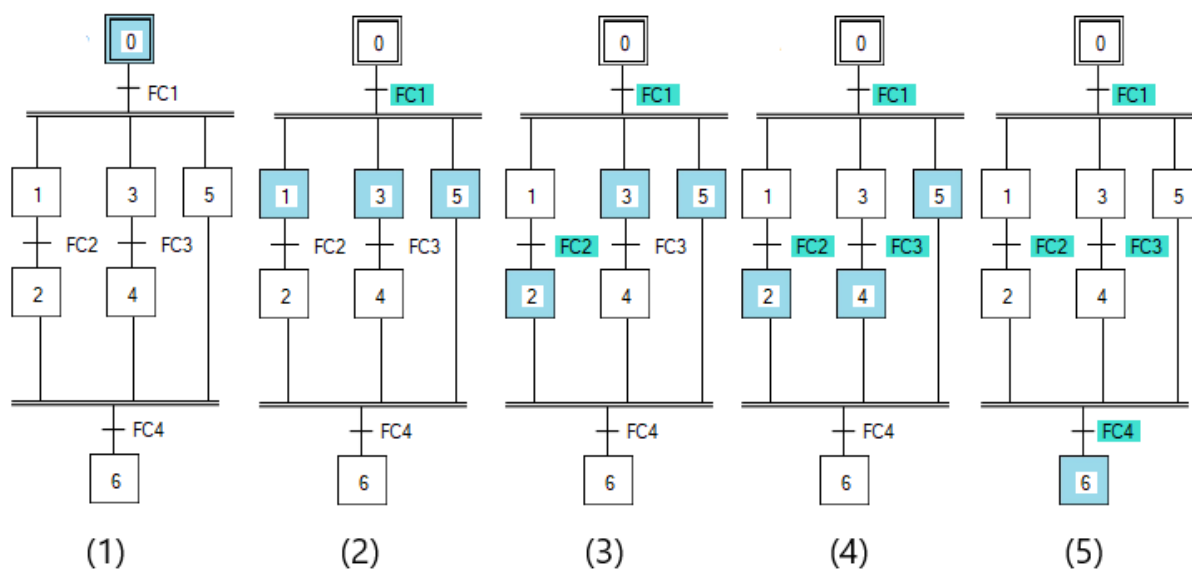
### 2.2.3. Convergència i divergència en "Y"

A més de les estructures de seqüències lineals i convergències i divergències en "O", existeixen les convergències i divergències en "Y" les quals permeten l'activació d'etapes al mateix temps, o sigui que es poden realitzar tasques de forma simultània.

Així com la tipologia anterior, aquest tipus també està format per tres parts. La primera construcció és una divergència en "Y" on a partir d'una etapa inicial es divideixen diversos camins que s'activaran al mateix temps quan es compleixi la transició associada a aquesta divergència. Després es solen trobar seqüències lineals per cada un dels camins possibles i, per últim, una convergència en "Y" on es reuneixen els anteriors recorreguts.

Per diferenciar-se de les transicions normals, les convergències i divergències d'aquest tipus es representen amb dues línies paral·leles sota la transició d'entrada (a la divergència) i sobre la transició de sortida (a la convergència).

Amb la fi d'ajudar a comprendre com funciona aquesta estructura i com és la seva representació, a continuació es mostra una imatge (figura 2.2.3.1) a mode d'exemple.



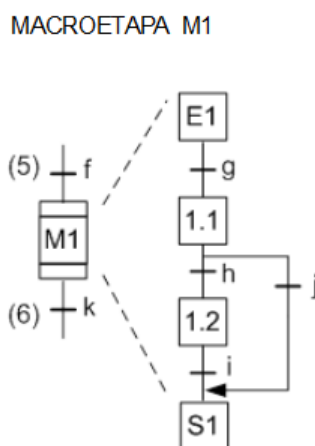
**Figura 2.2.3.1.** Exemple convergència i divergència en "Y". (Font: Pablo Mañas)

A la imatge anterior, es pot veure l'evolució d'un GRAFCET que conté una estructura de convergència i divergència en "Y". A l'inici (figura 2.2.3.1 (1)), es troba l'etapa inicial activada i quan es valida la transició FC1, la divergència en "Y" permet l'activació de les etapes 1,3 i 5 (figura 2.2.3.1 (2)). En aquest punt, si s'activés FC4, no es podria activar l'etapa 6 ja que la convergència en "Y" no ho permet pel fet de que les etapes anteriors actives haurien de ser 2,4 i 5 i no, únicament la 5. Per tant, es continuen les seqüències lineals dels dos camins encara oberts. A l'instant en que es valida FC2 (figura 2.2.3.1 (3)) s'activa l'etapa 2 i quan es verifica la transició FC3 (figura 2.2.3.1 (4)) es posa en marxa l'etapa 4. Ara sí que es podria activar l'etapa 6 quan la variable FC4 sigui vertadera i això és el que succeeix finalment a través de la convergència en "Y" (figura 2.2.3.1 (5)).

## 2.3. Macroetapes

Quan és necessari fer un GRAFCET d'un procés d'automatització a l'inici s'acostuma a fer ús de les macroetapes. Aquests elements serveixen per a fixar les primeres bases del procés sense donar molts detalls d'aquest. Això permet dividir el problema a grans trets sense haver d'especificar tot des del principi.

Posteriorment a la creació de les macroetapes aquestes s'hauran de detallar per tal d'obtenir les accions simples que només depenguin de les entrades i sortides del sistema. En resum, una macroetapa és un sol element que simbolitza diverses tasques i que més tard s'ha d'especificar. A continuació es pot observar la seva representació en un GRAFCET (figura 2.3.1):



**Figura 2.3.1.** Exemple macroetapa. (Font: Pablo San Segundo [13])

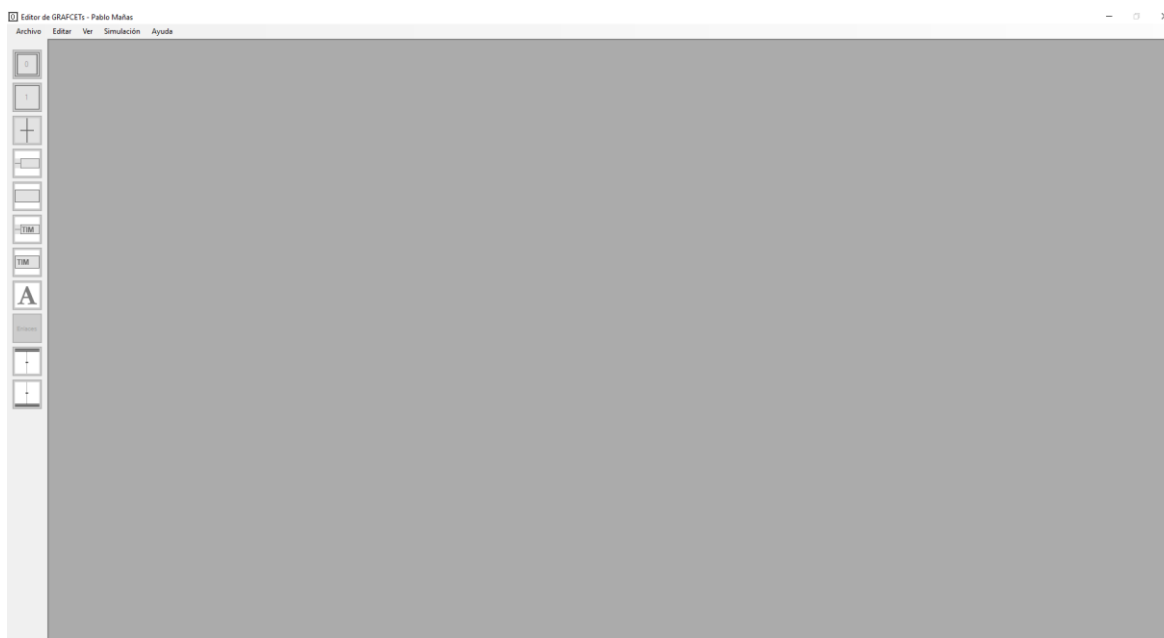
Com es pot veure a la imatge anterior, M1 representa el conjunt definit posteriorment per les etapes E1, 1.1, 1.2, S1 i les seves transicions.

### 3. Editor de GRAFCETs – Pablo Mañas

Ara que ja s’han presentat els diferents elements d’un GRAFCET juntament amb una breu descripció, en aquest apartat es mostrarà el funcionament de l’editor realitzat per l’autor amb Visual Studio, més concretament amb Microsoft Visual Studio 2017 i el llenguatge de programació de Visual Basic. Així doncs, a continuació s’exposaran totes les funcions possibles que permet el programa amb les imatges corresponents, les quals podran ser d’utilitat a l’usuari que usi l’editor. Més endavant es trobarà l’explicació del funcionament a nivell de programació.

#### 3.1. Inici del programa

Quan s’inicia el programa, la primera finestra que es mostra és la següent:

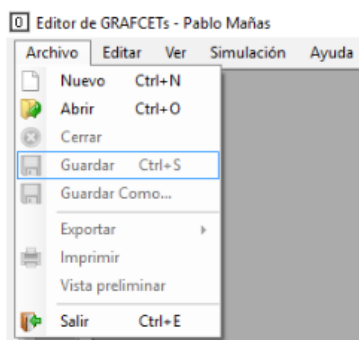


**Figura 3.1.1.** Inici editor de GRAFCETs (Font: Pablo Mañas)

Un primer punt observable del programa (vèrtex dret superior de la figura 3.1.1) és que es pot minimitzar i tancar, però no és possible maximitzar/restaurar. Això està dissenyat d’aquesta manera perquè l’usuari aprofiti en tot moment tota la pantalla del seu ordinador a l’hora d’editar els GRAFCETs. Altre fet apreciable és la barra esquerra de la figura anterior on es troben tots els botons desactivats ja que com no hi ha cap arxiu creat no cal que aquests es trobin habilitats. A la barra superior es poden veure les següents opcions: “Archivo”, “Editar”, “Enlaces”, “Simulación” i “Ayuda”, les quals es mostren amb més detall a continuació.

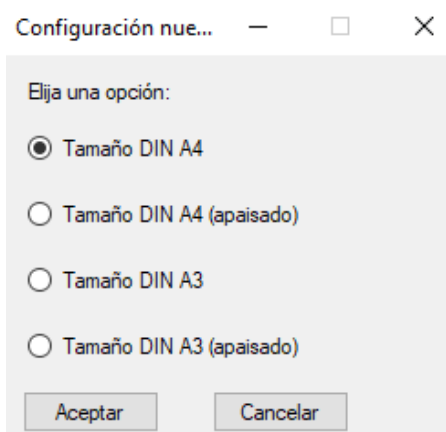


Quan l'usuari prem la casella "Archivo", es despleguen les següents opcions: "Nuevo", "Abrir", "Cerrar", "Guardar", "Guardar Como...", "Exportar", "Imprimir", "Vista preliminar" i "Salir"; tal i com es pot veure a la propera imatge (figura 3.1.2).



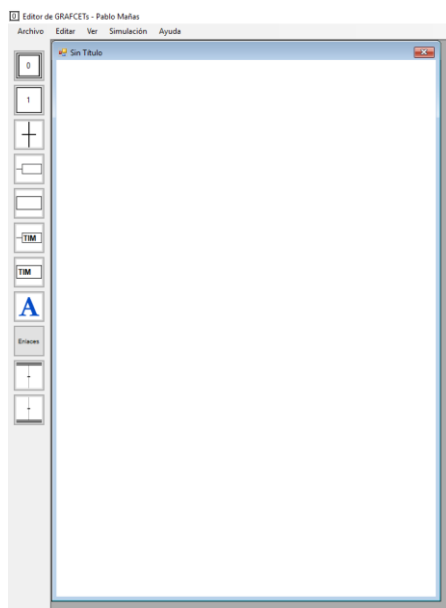
**Figura 3.1.2.** Clic en "Archivo" editor de GRAFCETS (Font: Pablo Mañas)

Arribat a aquest punt l'usuari podrà fer clic únicament a les opcions de "Nuevo", "Abrir" i "Salir", ja que encara no s'ha creat cap GRAFCET i per tant, no es podria ni guardar ni exportar ni imprimir. Si es considera que és el primer cop que s'inicia el programa, l'usuari únicament deuria clicar en "Nuevo" o "Salir" ja que no podria obrir cap arxiu. En el cas que es premi el botó de "Salir" o les tecles Ctrl+E, com és evident, el programa es tancarà i caldrà tornar a obrir-lo si es vol, en el cas que es faci clic en "Nuevo" o les tecles Ctrl+N, s'obrirà una finestra com la següent:



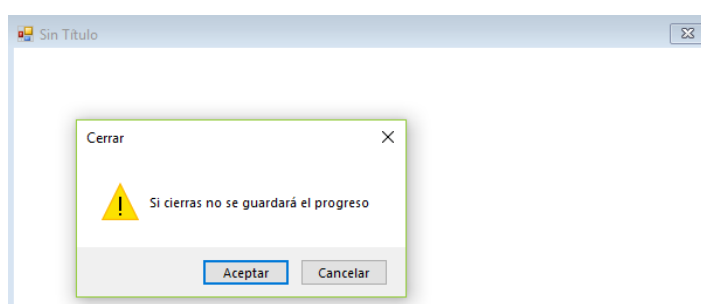
**Figura 3.1.3.** Clic en "Archivo"->"Nuevo" editor de GRAFCETS (Font: Pablo Mañas)

Ara es podrà escollir una opció segons la mida que es vulgui de paper (DIN A4 o DIN A3) amb la fi de crear un nou paper que contengui el GRAFCET que es desitja amb la mida adequada. En cas de fer clic a "Cancelar" l'operació es cancel·la i es torna a l'inici del programa (figura 3.1.1). Si, en canvi, es prem el botó "Aceptar" es crea un full com el que es mostra a continuació:



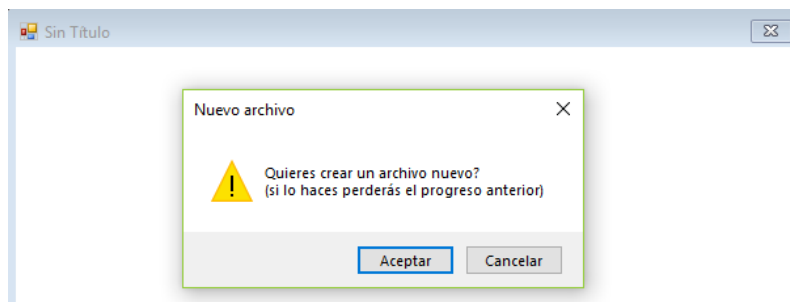
**Figura 3.1.4.** Nou arxIU (DIN A4) editor de GRAFCETs (Font: Pablo Mañas)

En el moment que apareix el nou document, s'habiliten tots els botons de la barra esquerra ja mencionats anteriorment i observables a la figura anterior i a partir d'ara podrem crear el GRAFCET fent clic a aquestes caselles. També s'habiliten les opcions anteriors de "Cerrar", "Guardar", "Guardar Como...", "Exportar", "Imprimir" i "Vista preliminar" encara que, com es pot veure (figura 3.1.4) el full està buit. En qualsevol punt del procés de creació i edició del GRAFCET, si es fa clic a "Archivo" i a "Cerrar" o a la creu del nou full creat, el programa adverteix que el progrés no es guardarà (figura 3.1.5) encara que s'hagi guardat prèviament.



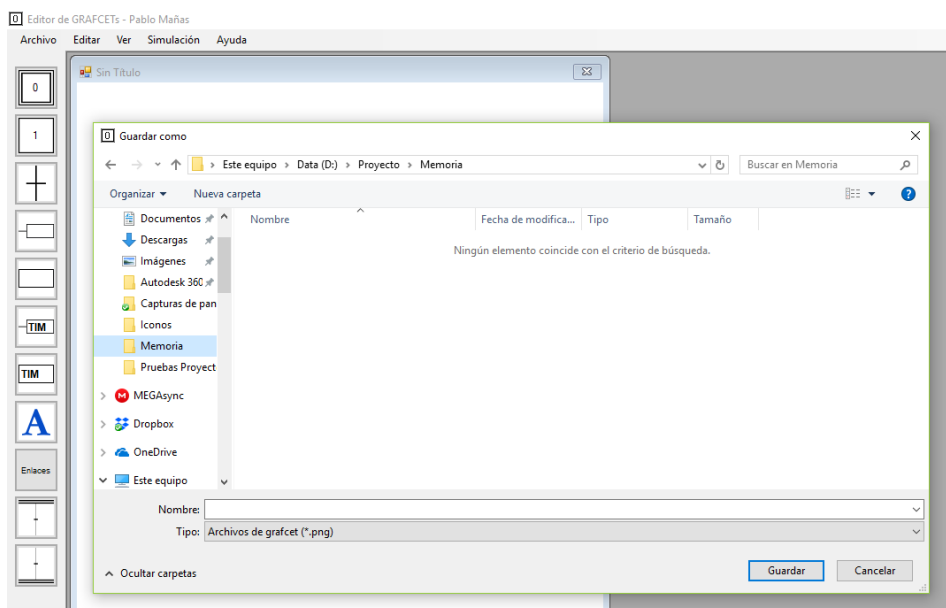
**Figura 3.1.5.** Advertència de tancament (Font: Pablo Mañas)

És necessari mencionar que el programa no admet que es creïn dos GRAFCETs al mateix temps i, per tant, en el cas que es premi la seqüència de "Archivo" i "Nuevo" o "Archivo" i "Abrir" després que s'hagi creat ja un arxIU es mostrarà la següent advertència:



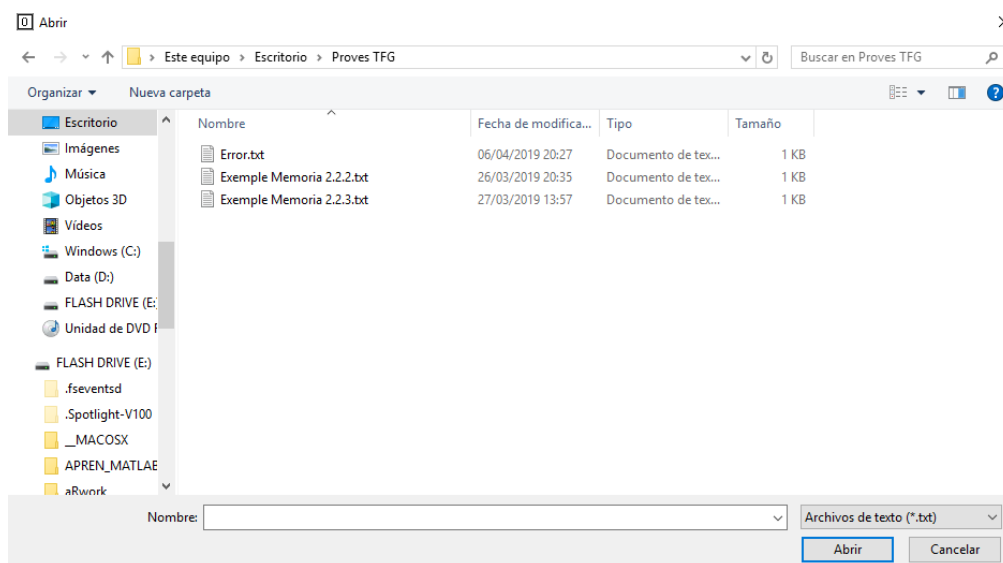
**Figura 3.1.6.** Advertència de crear nou arxiu sobre un altre (Font: Pablo Mañas)

Un cop es vulgui guardar el GRAFCET creat, caldrà fer clic a “Archivo” i “Guardar” o a “Archivo” i “Guardar Como...”. Si es el primer cop que es guarda, ambdós possibilitats actuaran de la mateixa forma. El programa obre un quadre de diàleg on l’usuari haurà de decidir on guardar l’arxiu tal i com es mostra a la figura 3.1.7. Si no es el primer cop que es guarda, al fer clic a guardar el programa automàticament guardarà on havia guardat el primer cop i amb el mateix nom que s’ha donat per primera vegada. En el moment que es fa el guardat es creen dos arxius, un amb la imatge en format .png del GRAFCET creat i altre en format .txt que conté la informació del GRAFCET i que és necessari per a quan es vulgui obrir de nou la creació en qüestió.



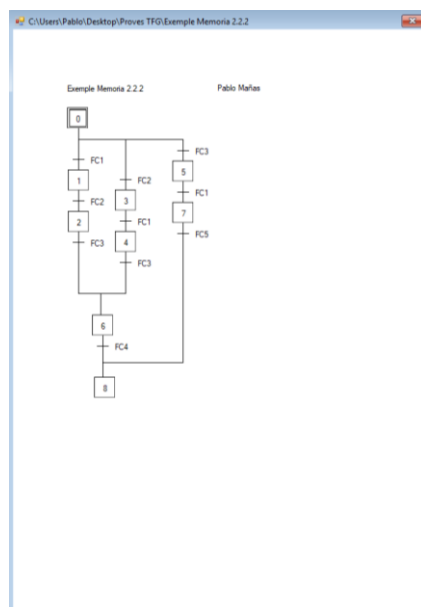
**Figura 3.1.7.** Quadre de diàleg guardar (Font: Pablo Mañas)

En el cas que ja s’hagi guardat un arxiu i es vulgui obrir, a l’iniciar el programa caldrà prémer sobre la seqüència “Archivo” i “Abrir”. El quadre de diàleg que es mostrarà serà similar al de guardat però es mostren tots els arxius .txt que es poden obrir així com es pot veure a la figura 3.1.8:



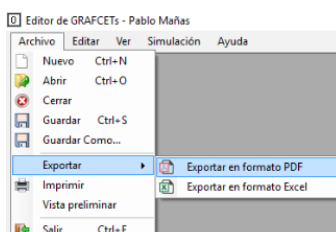
**Figura 3.1.8.** Quadre de diàleg obrir (Font: Pablo Mañas)

En el cas que s'obri un arxiu que no s'hagi creat a partir del programa, com pot ser el primer element que trobem a la imatge anterior que s'anomena Error.txt, a l'intentar obrir-se el programa no farà res i tornarà a la pantalla inicial. En canvi, si s'obre un arxiu correcte, per exemple el segon que trobem a la figura 3.1.8 que s'anomena "Exemple Memoria 2.2.2.txt" i que correspon a les figures 2.2.2.1, 2.2.2.2, 2.2.2.3 i 2.2.2.4 el programa obriria la següent finestra:



**Figura 3.1.9.** Obrir arxiu "Exemple Memoria 2.2.2" (Font: Pablo Mañas)

En el cas que es vulgui exportar el GRAFCET creat a un tipus d'arxiu pdf, l'usuari haurà de fer clic sobre "Archivo", "Exportar" i "Exportar en formato PDF" així com es veu a la figura que es mostra a continuació:

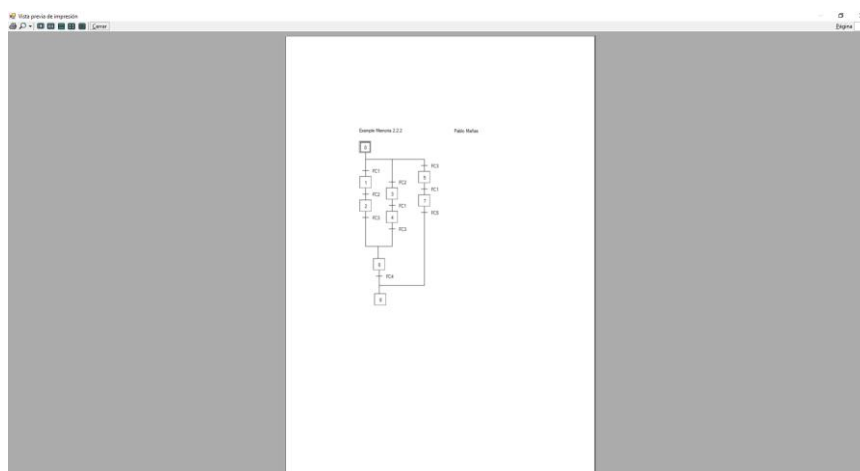


**Figura 3.1.10.** Seqüència "Exportar en formato PDF" (Font: Pablo Mañas)

En acceptar l'acció s'obre un quadre de diàleg similar al de la figura 3.1.7 on cal escollir el lloc on es vol guardar l'arxiu ara en format pdf. Cal dir que si el GRAFCET creat té la mida d'un DIN A3, el programa avisarà que no es pot crear un pdf d'aquesta mida ja que lamentablement no funciona correctament l'opció i, per tant, no ha estat inclòs.

Seguint l'ordre de la pestanya s'hauria d'explicar el funcionament de l'element "Exportar en formato Excel" però, aquest botó es profunditza més endavant a l'apartat d'aquest document 3.5 Simulació ja que cal conèixer altres característiques del programa prèviament.

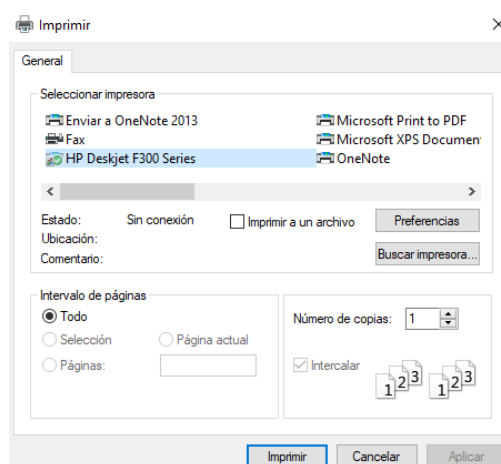
Si l'usuari decideix imprimir el GRAFCET és recomanable que primer faci clic sobre l'ítem "Vista preliminar", ja que proporciona la imatge de com quedarà la representació en paper primerament. Fent ús del GRAFCET obert a la figura 3.1.9, en el moment que es fa l'acció de clicar sobre l'element mencionat anteriorment, s'obre una finestra com la següent:



**Figura 3.1.11.** Exemple "Vista preliminar" (Font: Pablo Mañas)

Tal i com es mostra a la figura anterior, la 3.1.11, al vèrtex superior esquerre es troba un botó amb la imatge d'una impressora. Si es fa clic a sobre, segons si l'usuari ha fet clic anteriorment a imprimir o a exportar en format pdf, s'imprimirà el document o per al contrari, s'obrirà el quadre de diàleg ja explicat anteriorment sobre l'exportació en format pdf. En cas que el full sigui de mida DIN A4 (apaïsat) es mostrarà el full en posició horitzontal. Així com quan s'exporta, en cas que el GRAFCET sigui de mida DIN A3 no es podrà veure la vista preliminar i el programa ho avisa amb un missatge: "No hay vista preliminar DinA3".

En cas que l'usuari premi el botó de "Imprimir", el quadre de diàleg que es genera es el següent:

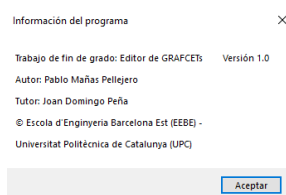


**Figura 3.1.12.** Quadre de diàleg imprimir (Font: Pablo Mañas)

En la imatge anterior es veu com es pot seleccionar la impressora adient i algunes característiques de la impressió. Quan estigui tot configurat l'usuari pot clicar sobre "Imprimir" i si no es produeix cap error s'imprimirà el GRAFCET actual.

És necessari remarcar que per defecte, la vista prèvia i la impressió creen uns marges per sobre i per sota del full en el que es fa el GRAFCET i, per tant, un cop s'hagi imprès el document no es veurà igual que a la finestra que es mostra al programa mentre s'edita.

Un últim element trobat a l'iniciar el programa el troba en "Ayuda" i "Acerca de..." on es troba la informació del creador del programa en un requadre petit que es pot observar a continuació:



**Figura 3.1.13.** Informació del programa (Font: Pablo Mañas)

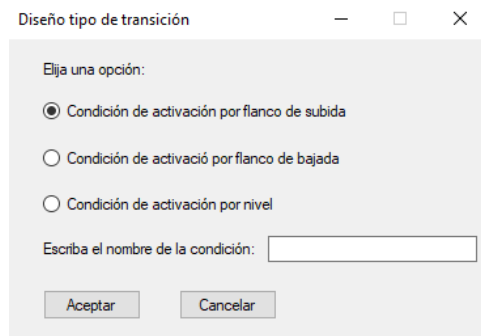
### 3.2. Barra d'elements

Ara que ja es coneix la manera de crear un nou arxiu, guardar, obrir, exportar i imprimir; a continuació s'explica el funcionament de la barra d'elements que es troba a la part esquerra del programa així com es veu a la imatge següent:



**Figura 3.2.1.** Barra d'elements (Font: Pablo Mañas)

El primer element que es troba a la figura anterior (començant per dalt) permet crear l'etapa inicial. Un cop l'usuari fa clic, es crea al GRAFCET l'etapa inicial i acte seguit es desactiva aquest botó ja que no es permet crear diverses etapes inicials. El segon ítem genera una etapa amb un número a dintre, aquest número, que defineix l'etapa, s'incrementa en 1 cada cop que es fa clic sobre aquest botó i això permet crear les diferents etapes que conformen el diagrama. El lloc d'aparició de les etapes en el GRAFCET és el mateix i per tant, es solapen; per tant, es recomana moure els objectes (s'explica a l'apartat 3.3) de forma adequada per a que al crear els nous elements, no es produeixi cap confusió. Si es segueix l'ordre de dalt a baix, es troba la transició. En prémer sobre aquest element es genera un diàleg amb l'usuari com el següent:

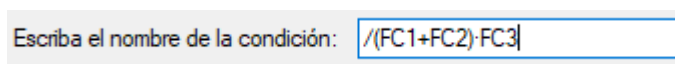


**Figura 3.2.2.** Disseny tipus de transició (Font: Pablo Mañas)

En aquest punt, l'usuari podrà escollir entre 3 opcions, si la condició d'activació es per flanc de pujada, de baixada o si és per nivell. Els dos primers casos només es diferencien en que es crearà a més de la etiqueta un dibuix que permet identificar el flanc, tot i que en simulació no canvien res i es consideren totes les condicions com activació per nivell. Així doncs només serveixen per fer la representació correcta. A l'hora d'escriure la condició cal tenir en compte una sèrie de consideracions si es vol fer la simulació del GRAFCET posteriorment:

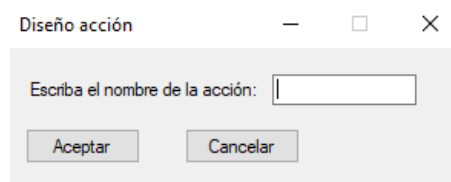
1. És necessari escriure la condició com si fos un GRAFCET de nivell 2, és a dir, no s'admeten espais i només es descriu l'acció de forma operativa.
2. Es poden utilitzar les operacions lògiques OR, AND i NOT i els parèntesis i s'hauria d'escriure utilitzant el +, · i /, respectivament.

Una possible expressió que respecta les consideracions anteriors pot ésser com la que es mostra en la figura 3.2.3:



**Figura 3.2.3.** Exemple expressió de condició (Font: Pablo Mañas)

El quart element de la barra d'elements (figura 3.2.1) permet crear una acció associada a una etapa. En aquest cas el quadre de diàleg que es genera és més simple i el seu aspecte és el següent:

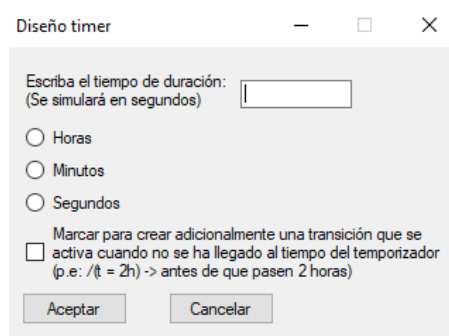


**Figura 3.2.4.** Quadre de diàleg acció (Font: Pablo Mañas)



En aquest cas només cal tenir en compte que l'acció com a màxim pot ocupar 8 caràcters, per això és importar descriure el GRAFCET com a nivell 2.

Si l'usuari fa clic sobre el següent element de la barra ja mencionada anteriorment es genera com en el cas anterior un quadre de diàleg molt similar al que s'observa a la figura 3.2.4. En escriure el nom de l'acció, es crea una acció associada a una acció ja creada anteriorment. Si es vol simular, només s'admet una acció i una acció associada en total, per tant, tan sols es poden associar dos accions a una etapa. Tot i així, es podrien representar més accions associades a una etapa.



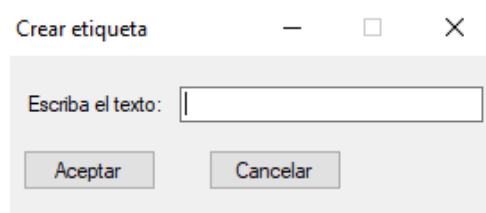
**Figura 3.2.5.** Quadre de diàleg disseny temporitzador (Font: Pablo Mañas)

Continuant amb la barra d'elements es troben els temporitzadors. En fer clic sobre l'ítem similar al de l'acció, però identificat amb les lletres "TIM" les quals representen "timer", que en anglès es tradueix com a temporitzador, es mostra un quadre com el que s'observa a la figura 3.2.5. En aquest tipus de diàleg s'ha d'escriure el temps amb nombres enters, ja que no s'admeten comes ni punts al requadre. Després es selecciona l'escala de temps en hores, minuts i segons i per últim, un requadre que s'ha de marcar en cas que es vulgui crear una transició activa mentre el temps del temporitzador no hagi transcorregut. És necessari mencionar que a la simulació el temps que s'escriu sobre el primer requadre serà el que es simularà en segons, és a dir, no és té en compte a la simulació si són hores, minuts o segons ja que sempre es farà l'avaluació com si fossin segons. Un cop es configuren els elements del temporitzador i es fa clic en acceptar, es crea un objecte similar a l'acció però que dins conté l'etiqueta TIM i un número que l'identifica de forma similar al que succeeix amb les etapes. A més es crea la transició corresponent a quan el temps succeeixi i si s'ha seleccionat el requadre mencionat anteriorment, la transició del temporitzador negada, o sigui, quan encara no ha transcorregut el temps.

Si es fa clic sobre el següent element, el qual també té l'etiqueta de "TIM", es mostra el mateix quadre de diàleg que s'acaba de mencionar. Les característiques són les mateixes, tot i que comparteixen comptador, és a dir, si ja s'ha creat un temporitzador amb el botó anterior, en configurar un de nou

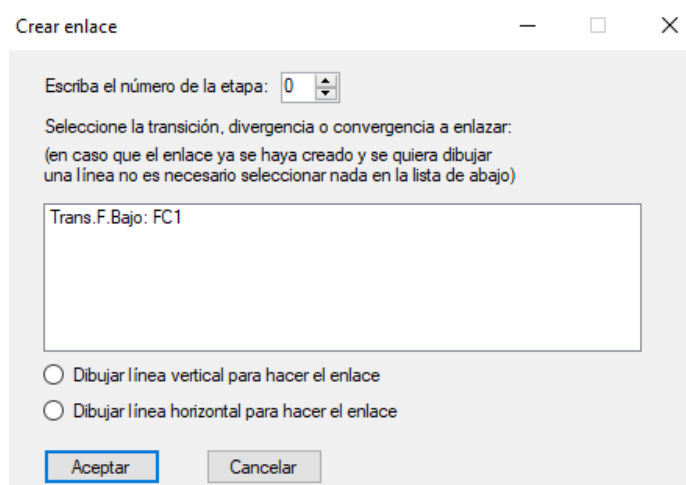
amb el botó ara en qüestió l'etiquetatge del nou temporitzador serà "TIM 2" ja que el primer ha sigut l'anterior. La única diferència observable amb l'element anterior és la forma de l'objecte, ja que en aquest cas, la morfologia és igual a la de les accions associades. Així com les accions, a una etapa, en simulació només es poden associar dos temporitzadors.

Per crear etiquetes al GRAFCET l'usuari ha de prémer sobre el botó amb una "A" blava observable a la figura 3.2.1. En el moment que es fa clic a sobre, es produeix un quadre de diàleg on es pot escriure l'etiqueta que es vol generar tal i com es mostra a la figura següent:



**Figura 3.2.6.** Quadre de diàleg etiqueta (Font: Pablo Mañas)

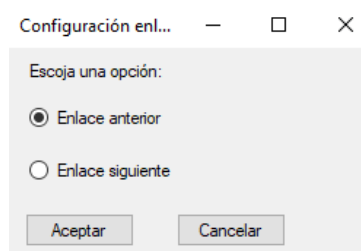
En un GRAFCET és necessari unir els elements amb enllaços, per tant, cal dibuixar-los d'alguna manera per fer la seva representació correcta. Si l'usuari clica sobre l'element "Enlaces" de la barra d'elements (figura 3.2.1) es genera un quadre com el que es veu a continuació:



**Figura 3.2.7.** Quadre de diàleg crear enllaç (Font: Pablo Mañas)

La primera part del quadre de diàleg consisteix en crear els enllaços que s'associen a una etapa. Com es pot veure a la imatge anterior, inicialment es pot escollir el número de l'etapa a la que s'associarà l'element seleccionat al quadre inferior. Això està dissenyat perquè poden produir-se errors quan es construeix el GRAFCET i l'única manera d'arreglar-lo és que el usuari d'aquesta forma fixi la transició,

divergència o convergència que s'enllaça. Quan es selecciona un ítem de la llista i s'accepta, es genera una nova finestra on es decideix si l'enllaç és anterior o següent tal i com es mostra la figura 3.2.8:

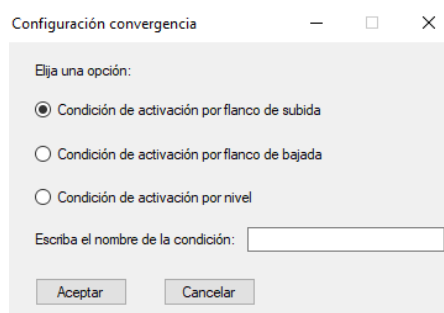


**Figura 3.2.8.** Quadre de diàleg configuració enllaç (Font: Pablo Mañas)

Aquesta seqüència explicada permet actualitzar la taula d'enllaços de les etapes, característica que es mostra i es detalla en l'apartat d'aquest document 3.4.

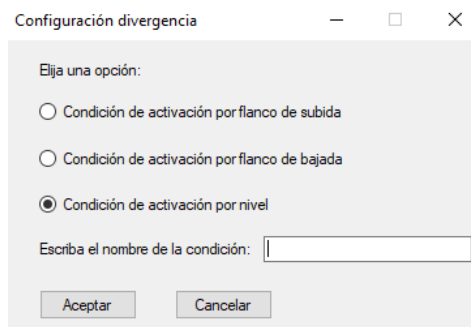
Sense canviar de tema, a més de generar els enllaços a la taula mencionada amb anterioritat, la segona part del quadre de diàleg observat a la figura 3.2.7. consisteix en generar una línia vertical o horitzontal. Si no cal arreglar els enllaços és possible només crear les línies verticals i horitzontals del GRAFCET. És recomanable primer construir el GRAFCET amb les línies corresponents i, després, fixar els enllaços que no siguin correctes, ja que un cop es fixin no es poden desfer. En acceptar sense fer clic en cap element de la llista es crea una línia vertical o horitzontal al document que a l'apartat següent es mostra com es deuen moure, ampliar i connectar.

Els últims elements que s'observen a la barra d'element de la figura 3.2.1. són les convergències en "Y" i les divergències en "Y". En el primer cas, en fer clic al botó corresponent apareix un quadre de diàleg similar al de la figura 3.2.2. però, amb el títol diferent, tal i com es mostra a la imatge següent:



**Figura 3.2.9.** Disseny tipus de convergència (Font: Pablo Mañas)

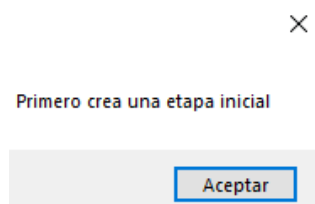
En el segon cas, és a dir, quan es fa clic a la divergència en "Y" el quadre que es genera és el següent:



**Figura 3.2.10.** Disseny tipus de divergència (Font: Pablo Mañas)

Les regles que s’han mencionat amb anterioritat sobre la simulació a les transicions en aquests dos casos també s’apliquen i, per tant, un bon exemple de com caldria escriure els noms de les condicions s’observa a la figura 3.2.3.

Per últim, és necessari dir que en el cas que no hi hagi una etapa inicial creada no es permet crear cap dels següents elements, ja que es considera imprescindible que el GRAFCET contingui una etapa d’aquest tipus. Si s’intenta crear un element diferent sense haver creat amb anterioritat l’etapa inicial el programa adverteix amb el següent missatge:

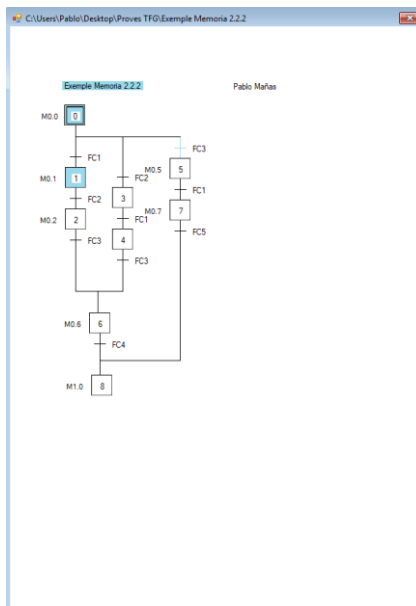


**Figura 3.2.11.** Missatge d’alerta quan no hi ha etapa inicial (Font: Pablo Mañas)

Com ja s’ha dit anteriorment amb les etapes, tots els elements es solapen entre sí, per tant, si es creen primer tots els elements possiblement sigui més difícil trobar-los després. Es recomana en aquest cas moure els elements inicialment cosa que s’explica en el pròxim apartat.

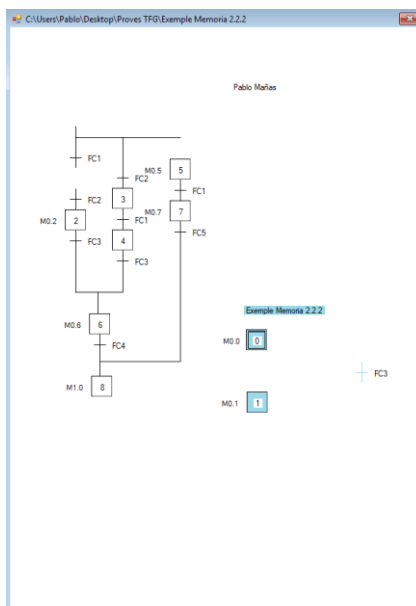
### 3.3. Esdeveniments de teclat, ratolí i edició d’elements

En aquest apartat de la memòria s’explica com es mouen els objectes creats a partir de la barra d’elements i com es modifiquen amb l’ajuda del teclat, el ratolí i la pestanya superior “Editar”. Per detallar com és el funcionament, a continuació es mostrarà un GRAFCET que ja s’ha mostrat amb anterioritat per veure com es poden moure i modificar els diferents elements que el componen.



**Figura 3.3.1.** Elements seleccionats del GRAFCET en color blau (Font: Pablo Mañas)


Com es pot observar a la figura anterior, en fer clic sobre qualsevol objecte del GRAFCET, aquest es selecciona. Per distingir un element seleccionat d'un altre, el primer es representa amb el fons de color blau o amb l'element en sí de color blau tal i com es mostra a la imatge en qüestió (figura 3.3.1). Un cop seleccionats es poden moure en cas que es mantingui polsat el botó del ratolí i aquest es mogui o, també, utilitzant les fletxes del teclat. Si desplaçem els elements anteriors una nova imatge reflexa el que succeeix:

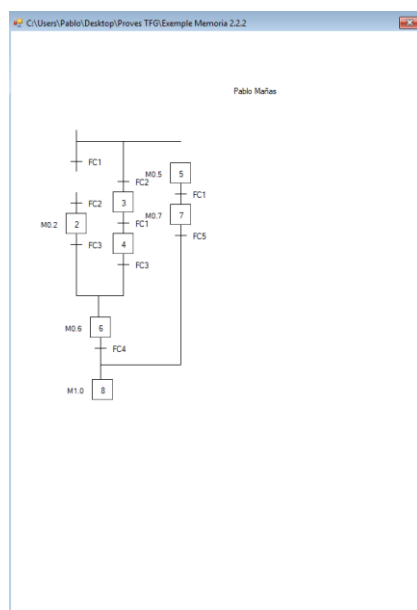


**Figura 3.3.2.** Desplaçament d'elements seleccionats (Font: Pablo Mañas)

En cas que es vulgui tornar a com es trobava abans de fer l'acció es poden prémer les tecles del teclat Ctrl+Z o fer clic a "Editar" i "Deshacer". Cal dir que si l'acció de moure s'ha realitzat amb el teclat l'opció de desfer només tornarà al desplaçament anterior i, per tant, no s'aconseguirà tornar mai a la posició inicial, només a l'últim moviment que haurà sigut d'un píxel. Es recomana, doncs, moure els elements del GRAFCET amb el ratolí. Si amb el ratolí es mouen els elements com es mostra a la figura 3.3.2 es prem Ctrl+Z o "Editar" i "Deshacer" i acte seguit es prem Ctrl+Y o "Editar" i "Rehacer" el GRAFCET tornaria a mostrar-se com la imatge anterior ja que el botó "Rehacer" recupera la posició quan es fa l'acció de "Deshacer".

Tal i com s'observa a la figura 3.3.2 com hi havia diversos elements seleccionats tots es mouen en grup, tot i que es podrien moure de forma individual seleccionant únicament un a cada moviment. En el cas que s'intenti moure algun dels elements fora del full, el programa no ho permet i es quedaran els elements a l'extrem de qualsevol dels marges on s'hagin desplaçat.

Continuant amb la imatge mostrada anteriorment, la figura 3.3.2, en aquest instant, si es prem la tecla de "supr" o la fletxa per esborrar "  " anomenada "Backspace", els elements seleccionats s'eliminen i desapareixen del GRAFCET. A la figura següent es mostraria el resultat:



**Figura 3.3.3.** Elements esborrats (Font: Pablo Mañas)

En aquest cas si després es fa clic al botó de "Deshacer" ja explicat anteriorment, es recuperarien els objectes eliminats i apareixerien al lloc on han sigut esborrats. Cal afegir que si ara s'intentés fer l'acció de "Rehacer", no es podria ja que no tindria molt sentit tornar a esborrar, els elements. En cas que es vulgui tornar a esborrar només caldria tornar a seleccionar els objectes i prémer la tecla adient.

A la pestanya de “Editar”, es troba l’ítem de “Cortar” que també s’activa polsant Ctrl+X. Aquesta eina permet esborrar els elements seleccionats, però guardant-los a portafolis. Per tant, si es selecciona un element i després es talla, es podrà utilitzar el botó de “Pegar” o Ctrl+V, perquè torni a aparèixer aquest objecte o grup d’objectes en cas que s’hagi seleccionat varis. Una característica de “Pegar” és que els elements que s’hagin copiat prèviament o tallat en el seu defecte, un cop s’enganxin ho faran al lloc inicial on es crea el propi objecte. Per exemple, si l’usuari ha tallat una transició i una etapa i posteriorment l’enganxa, aquesta transició apareixerà al lloc on apareixen totes les transicions quan es creen i l’etapa al seu lloc de naixement corresponent. Es podria modificar per a què apareguin en altres llocs però s’ha considerat que així ja funciona correctament.

Tot i que ja s’ha explicat per sobre una mica amb l’exemple anterior, el botó de “Copiar” o Ctrl+C permet copiar els elements al portafolis per després ser enganxats. Cal dir que l’etapa inicial no es pot copiar, ja que no té sentit que n’hi hagin dues o més etapes inicials i el programa adverteix que l’etapa inicial si està seleccionada en el moment de copiar no es copiarà a través d’un missatge. Una característica destacable a comentar és que no es recomanable copiar etapes, ja que aquestes, un cop s’enganxin conserven el mateix número d’etapa i marca i, per tant, a l’hora de simular o veure els enllaços provoca errors al programa o males interpretacions.

Si es segueix parlant de la pestanya de “Editar”, per últim, es troba el botó de “Seleccionar todo”. Com el seu nom indica, aquest ítem permet seleccionar tots els objectes creats al GRAFCET, cosa bastant útil quan es vol moure tot el conjunt.

Al subapartat anterior (3.2 Barra d’elements), s’han mencionat els enllaços i la seva forma de crear-los. Ara, cal parlar sobre com s’editen o es poden variar. En primer lloc, un enllaç no deixa de ser gràficament una línia recta i aquesta per comoditat pot ser vertical o horitzontal. Bé doncs, el programa permet dibuixar línies verticals i línies horitzontals, però un cop es creen l’usuari pot creure que la línia no es pot allargar ni escurçar ja que no s’ha programat cap esdeveniment de ratolí que ho permeti fer. També, es pot pensar que unint diferents línies verticals, per exemple, es pot fer una de més llarga. Tot i que aquest raonament és lògic, el programa està pensat per a que qui l’utilitzi faci ús de les tecles “+” i “-” del teclat per allargar i escurçar les línies en qüestió. D’aquesta manera, no caldrà unir dues línies del mateix tipus per formar una de més llarga. La taula que es mostra a continuació, resumeix tot els esdeveniments de teclat aplicables als objectes creats i seleccionats:

Tecla	Acció	Objecte en que s’aplica (seleccionat)
Ctrl+Z	Desfer	Tots
Ctrl+Y	Refer	Tots

<b>Ctrl+X</b>	Tallar	Tots
<b>Ctrl+C</b>	Copiar	Tots
<b>Ctrl+V</b>	Enganxar	Tots
↑	Moure un píxel cap amunt	Tots
↓	Moure un píxel cap avall	Tots
→	Moure un píxel cap a la dreta	Tots
←	Moure un píxel cap a l'esquerra	Tots
<b>Supr i Backspace</b>	Esborrar	Tots
+	Allargar	Línia vertical i línia horitzontal
-	Escurçar	Línia vertical i línia horitzontal

**Figura 3.3.4.** Taula esdeveniments de teclat aplicats als objectes (Font: Pablo Mañas)

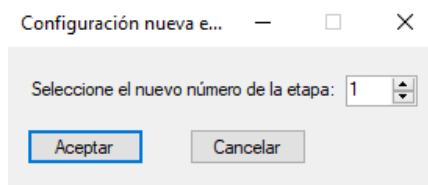
Per ajudar a l'usuari a enllaçar els diferents elements entre sí de forma correcta quan es mou un element seleccionat a prop d'un altre de forma que es solapen entre sí en deixar anar el clic del ratolí, la majoria de cops el programa interpreta que aquests dos objectes s'han d'unir i per tant, els situa al lloc corresponent per a que es creï l'enllaç de forma automàtica. Aquests enllaços dels quals es parla es veuen més endavant en profunditat a l'apartat 3.4 d'aquest document.

Tot i que aquesta característica del programa sol ser de gran ajuda, no funciona de forma correcta quan s'està movent un grup d'elements ja units a un objecte on es vol fer l'annexió, ja que, probablement només un dels elements s'unirà (el que fa contacte amb l'altre objecte) i la resta perdran la unió que tenien amb l'element que s'acaba d'unir. També és important mencionar, que si un element es vol unir a altres diferents com podria ser una línia vertical que es vol enllaçar a una etapa, una transició i dues línies horitzontals, probablement l'element s'unirà a un dels 4 i per tant, caldrà moure la resta per acomodar-se. En aquest cas, es recomana unir-lo a 1 inicialment i afegir la resta a ell i en cas que es vulgui moure, fer-ho amb les fletxes del teclat, per precaució.

En el cas que l'usuari vulgui desactivar aquesta opció, caldrà que polsi sobre "Ayuda" i "Desactivar ayuda posicionamiento". Gràcies a aquest botó tota persona que utilitzi el programa podrà activar i desactivar aquesta opció quan vulgui o quan cregui convenient.

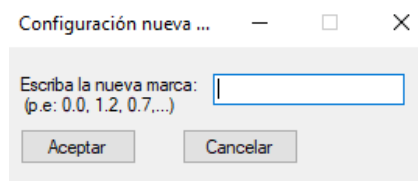


Per finalitzar aquest subapartat de la memòria cal explicar alguns esdeveniments del ratolí que encara no s'havien profunditzat. Es probable que l'usuari s'equivoqui a l'hora d'escriure una etiqueta, una acció, una condició de la transició, etc. Per evitar haver de crear de nou aquests elements i esborrar els que contenen les errades, l'usuari pot fer doble clic a esquerre a l'etiqueta en qüestió i s'obrirà un quadre de diàleg on es pot establir el nou nom que se li dona a l'element que es vol canviar. En el cas que es faci doble clic sobre el número d'una etapa (exceptuant l'inicial), el quadre de diàleg que es mostra és el següent:



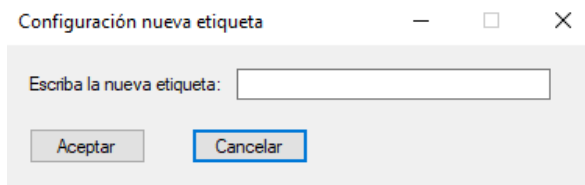
**Figura 3.3.5.** Quadre de diàleg nou número de etapa (Font: Pablo Mañas)

Així doncs, l'usuari podrà establir clicant sobre les fletxes el número d'etapa que vol. Si es canvia el número d'etapa és lògic també canviar la marca, per tant, si es fa doble clic sobre l'etiqueta de la marca el quadre de diàleg mostrat és el que es representa a continuació:



**Figura 3.3.6.** Quadre de diàleg nova marca de etapa (Font: Pablo Mañas)

Al text entre parèntesis es mostra com hauria d'escriure's la nova marca i per tant, no cal recalcar cap característica més. Per canviar la resta d'etiquetes possibles el quadre de diàleg que es mostra és el següent:



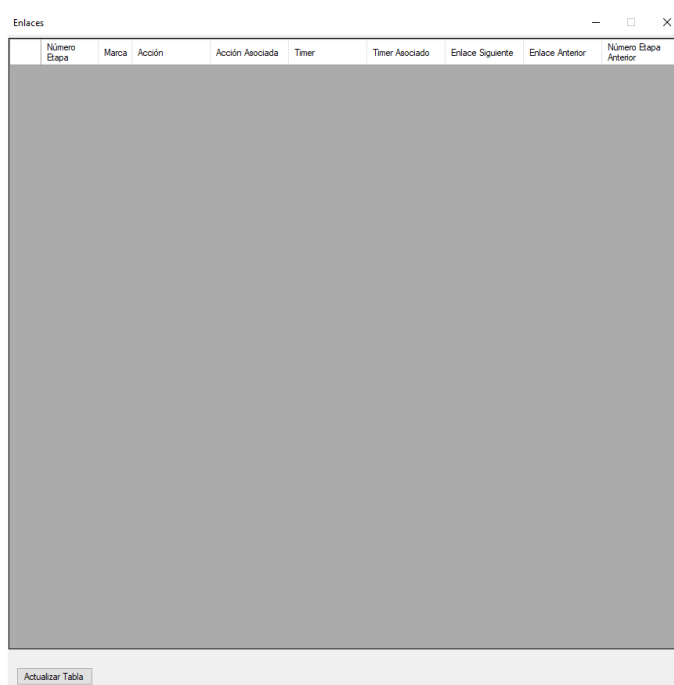
**Figura 3.3.7.** Quadre de diàleg nova etiqueta (general) (Font: Pablo Mañas)

El quadre mostrat a la figura anterior, apareix en fer doble clic sobre les etiquetes dels elements que es poden canviar. Si es vol canviar una etiqueta d'un element i no apareix aquest diàleg en qüestió,

possiblement no es pugui fer aquesta operació com es dona en el cas dels temporitzadors, els quals no admeten canvi d'etiqueta i exigeixen que s'hagin de crear de nou si s'ha produït una errada.

### 3.4. Enllaços

Un cop explicat el moviment dels objectes, aquest apartat versa sobre la correcta creació dels enllaços entre els elements que componen el GRAFCET que es vol generar. Amb la fi d'informar a l'usuari si està fent els enllaços de forma correcta per a realitzar la simulació posteriorment, a la pestanya "Ver" es troba la casella de "Ver enlaces" . Quan l'usuari fa clic a sobre i no hi ha cap element al GRAFCET, el programa mostra una finestra com la següent:



**Figura 3.4.1.** Finestra enllaços (Font: Pablo Mañas)

Com en aquest cas el GRAFCET està buit, la taula que es mostra a la imatge anterior també està buida. En el cas que es tinguin enllaços creats, com quan es realitza un diagrama com el que es mostra a la figura 3.3.1, la taula que s'observa és la següent:

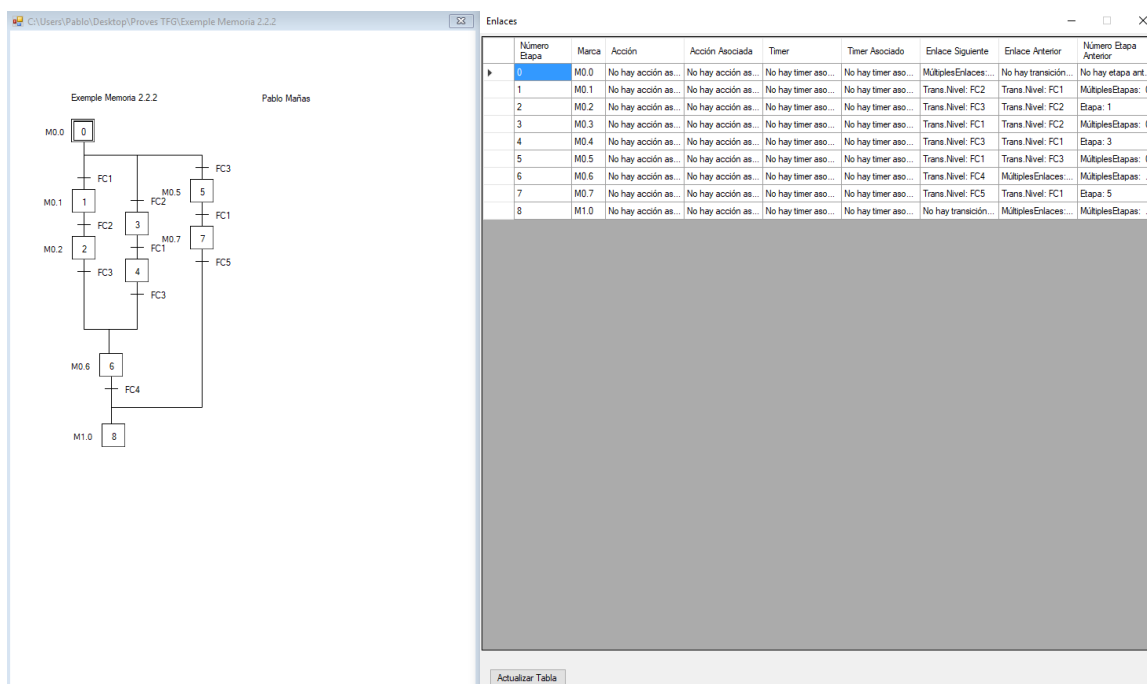


Figura 3.4.2. GRAFCET (esquerra) i finestra d'enllaços (dreta) (Font: Pablo Mañas)

Tal i com es pot observar a la imatge anterior (figura 3.4.2 (dreta)), a la finestra d'enllaços es mostra primer de tot les etapes ordenades per número. En el cas que s'hagin copiat etapes o n'hi hagin de repetides (amb el mateix número) si es vol fer la simulació posteriorment, caldrà posar-los un número diferent a cada etapa per evitar errors, tot i que als enllaços es veuria correctament. A cada etapa s'associa també una marca, la qual es pot observar de forma gràfica a la imatge de l'esquerra de la figura 3.4.2. Si no es volen mostrar al GRAFCET, l'usuari pot prémer el botó de "Ver" i "Ocultar marcas" i deixaran de ser visibles, en el cas que es vulguin tornar a fer perceptibles caldrà fer clic de nou al mateix botó i tornaran a ser observables. Com no hi ha cap acció ni temporitzador associat a cap de les etapes, les columnes corresponents mostren un missatge en el que es diu que no hi ha cap acció o temporitzador associat. En el cas que existissin accions i temporitzadors associats, les columnes corresponents estarien complertes amb el missatge "Acción: " i el nom de l'acció o "Timer: " i el nom del temporitzador. Pel que fa als enllaços, en el cas que una etapa en tingui múltiples associats, mostrarà que té múltiples i els separarà amb comes (",") i encara que a la imatge anterior no es pot veure sí que es pot observar a la figura 3.4.3. Pot succeir que per la morfologia del GRAFCET l'enllaç anterior sigui el següent i l'enllaç següent, l'anterior, per arreglar-lo caldrà fer aquests enllaços de forma "manual" com ja s'ha explicat al subapartat de 3.2 Barra d'elements a la secció que parla sobre els enllaços. Altres errors menors observables a la imatge de la dreta anterior és quan una etapa està associada únicament a una altra i el missatge que es mostra es que n'hi ha de múltiples associades. Això es degut a com està programada la forma de cercar els enllaços de forma automàtica però no és

un error greu ja que només és una errada en el nom de l'etiqueta ja que després només es mostra un element associat, en aquest cas un únic número d'etapa.

▶	0	M0.0	No hay acción as...	No hay acción as...	No hay timer aso...	No hay timer aso...	Múltiples	MúltiplesEnlaces: FC1, FC3, FC2	ay etapa ant...
---	---	------	---------------------	---------------------	---------------------	---------------------	-----------	---------------------------------	-----------------

**Figura 3.4.3.** Múltiples enllaços (Font: Pablo Mañas)

En el cas que es vulgui fer la simulació del GRAFCET creat de forma correcta o simplement, exportar la taula a un document Excel i que aquesta no contengui errors, és recomanable que l'usuari un cop creï el GRAFCET de nou, obri aquesta finestra per veure si està realitzant de forma correcta els enllaços o per al contrari no està enllaçant bé els elements. També, cal dir que el botó "Actualizar tabla" serveix per actualitzar la taula d'enllaços, tot i que el programa ho fa automàticament gairebé a cada canvi que es produeix al GRAFCET.

### 3.5. Simulació

Es considera un apartat important del programa el fet que el GRAFCET es pugui simular, així doncs, en aquesta secció es parla sobre el funcionament de la simulació. Per poder simular un diagrama primer de tot és necessari que prèviament s'hagi creat el GRAFCET i els seus enllaços estiguin ben construïts. Per tal d'explicar com funciona aquesta característica del programa, a continuació es mostra una sèrie d'exemples realitzats amb el programa.

El primer GRAFCET que es sotmet a estudi és el següent:

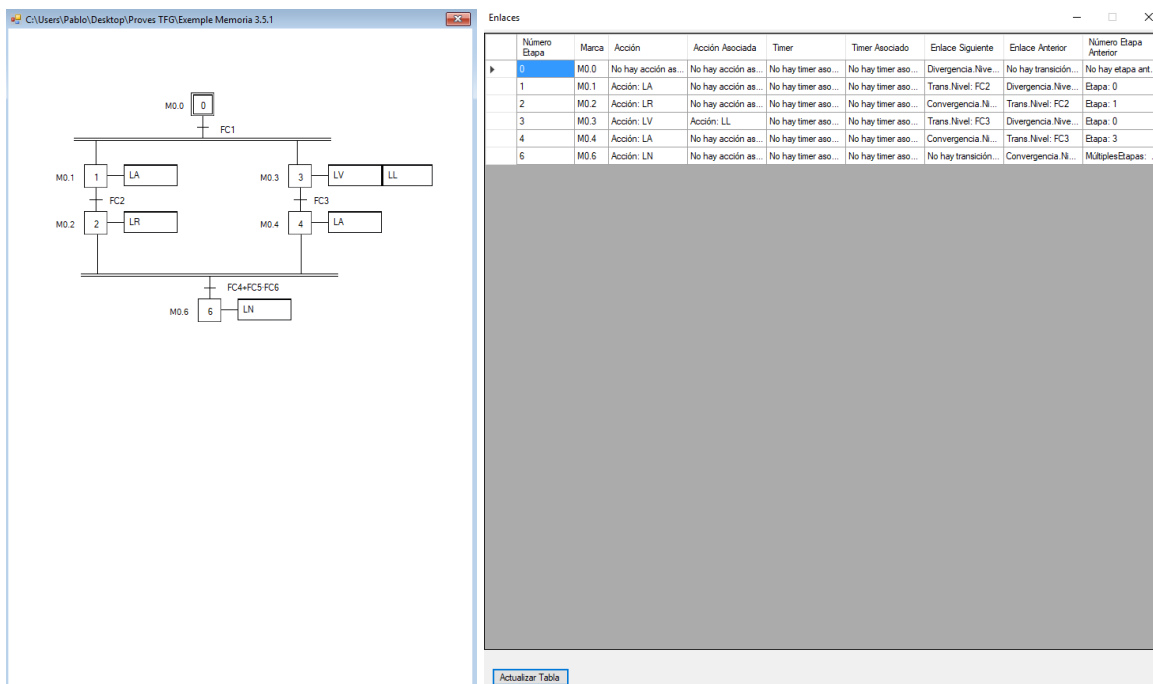


Figura 3.5.1. Exemple GRAFCET i la seva taula d'enllaços (Font: Pablo Mañas)

Tal i com s'observa a la figura anterior, s'ha creat un GRAFCET amb diferents etapes, transicions, convergències, divergències, accions i accions associades. Si l'usuari vol simular-lo només caldrà que faci clic a "Simulació" i a "Simular". Fent aquesta seqüència el programa obrirà una nova finestra on es fa la simulació que és una còpia del GRAFCET fet i on es veuran les etapes actives i accions, a més, apareixerà una nova taula amb les entrades i sortides del sistema i els temporitzadors actius. Com en aquest cas no hi ha cap temporitzador actiu el programa ho avisa amb un missatge. La nova taula creada és la següent:

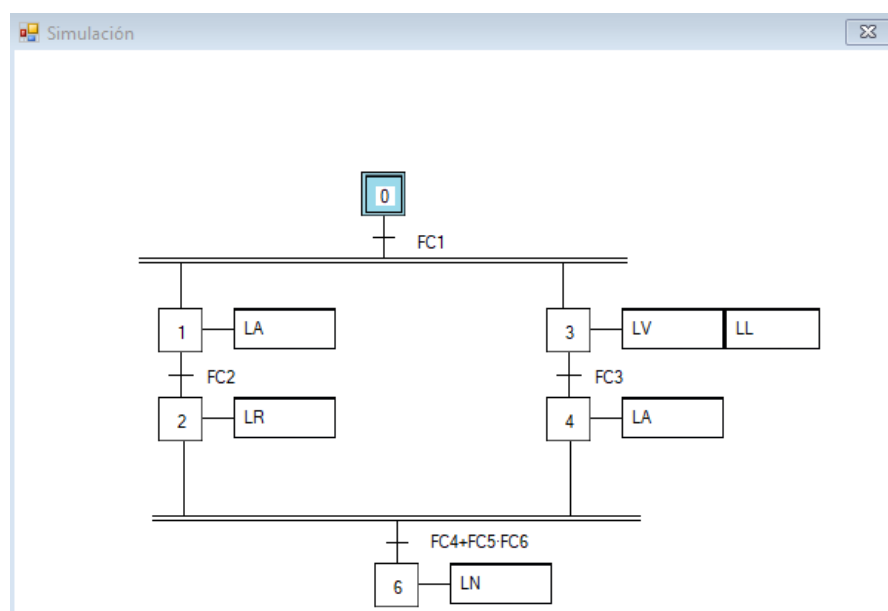
Entradas/Salidas

Entradas	Salidas	Transiciones Timer	Timers
FC1 - I0.0	LA - Q0.0		
FC2 - I0.1	LR - Q0.1		
FC4 - I0.2	LV - Q0.2		
FC3 - I0.3	LL - Q0.3		
FC5 - I0.4	LN - Q0.4		
FC6 - I0.5			

Activar entrada    Desactivar entrada

**Figura 3.5.2.** Taula d'entrades i sortides del sistema del GRAFCET 3.5.1 (Font: Pablo Mañas)

I la nova finestra que apareix es mostra a continuació:



**Figura 3.5.3.** Finestra de simulació del GRAFCET 3.5.1 (Font: Pablo Mañas)

Fent un petit incís, cal dir que en cas que es tanqui una de les dues taules anteriors (figura 3.5.1 (dreta) i figura 3.5.2) la simulació deixarà de funcionar i caldrà tancar la finestra de simulació (figura 3.5.3) per a que es torni a habilitar el GRAFCET inicial (figura 3.5.1 (esquerra)). Així doncs, per fer la simulació de forma correcta, s'han de deixar obertes les dues taules ja mencionades.

Així com s'observa a la figura 3.5.2. es troben totes les entrades del sistema és a dir, totes les condicions que fan que el sistema variï o sigui, que les sortides, en aquest cas les accions, s'activin i desactivin. Aquestes entrades es poden activar i desactivar mitjançant els botons inferiors de la taula. En el cas que s'activin les entrades FC2, FC3 i FC4 a la simulació es mostrarà aquest canvi:

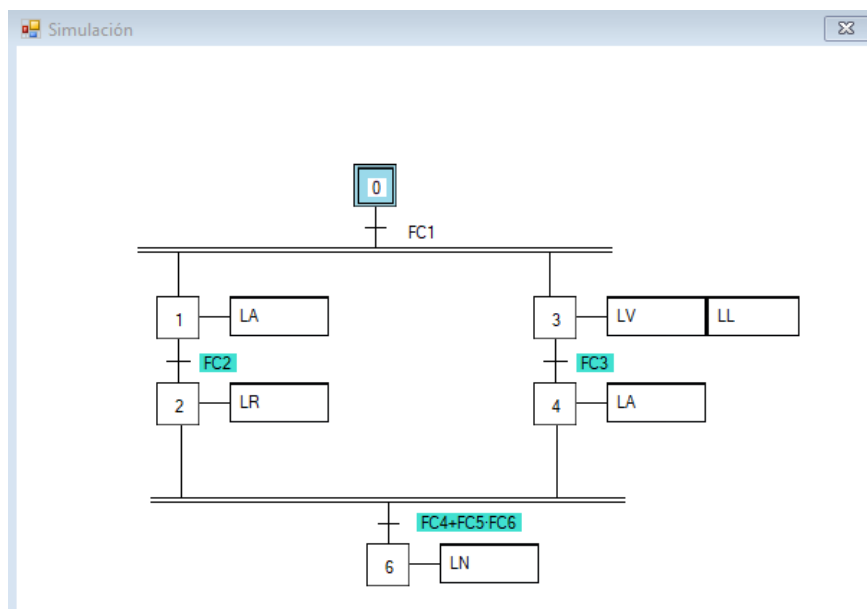


Figura 3.5.4. Finestra de simulació del GRAFCET 3.5.1 (Font: Pablo Mañas)

I a la pròpia taula es veurà el següent:

Entradas/Salidas

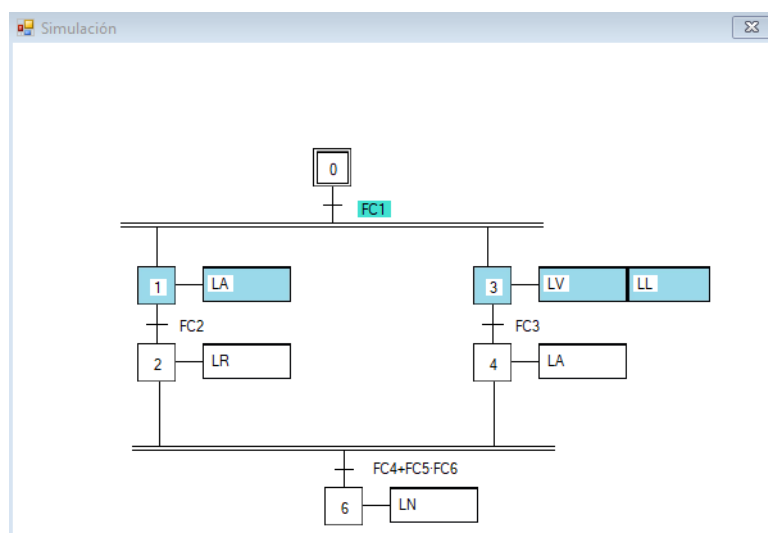
Entradas	Salidas
FC1 - I0.0	LA - Q0.0
FC2 - I0.1	LR - Q0.1
FC4 - I0.2	LV - Q0.2
FC3 - I0.3	LL - Q0.3
FC5 - I0.4	LN - Q0.4
FC6 - I0.5	

Activar entrada    Desactivar entrada

Figura 3.5.5. Fragment taula d'entrades i sortides de simulació del GRAFCET 3.5.1 (Font: Pablo Mañas)

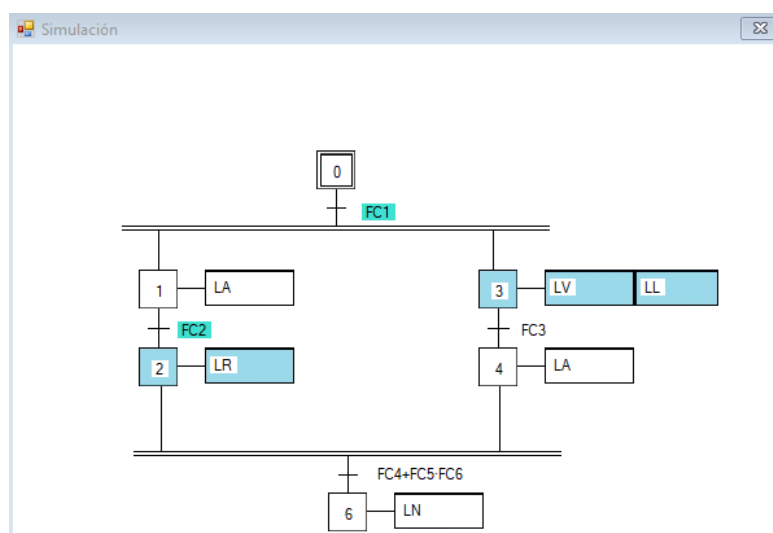
Com s'observa a la figura 3.5.4 i a la 3.5.5 les entrades ara actives es marquen amb un fons blau turquesa, en el cas de la primera figura en qüestió el fet que s'activi FC4 permet que la condició FC4+FC5-FC6 es validi i, per tant, també es marca del color corresponent. Si es desactiven les entrades amb l'altre botó es tornarà a obtenir el mateix que a la figura 3.5.3 i a la 3.5.2.

En el cas que s'activi inicialment FC1, s'haurien d'activar les etapes 1 i 3 juntament amb les seves accions associades i així succeeix tal i com es mostra a continuació:



**Figura 3.5.6.** Finestra de simulació del GRAFCET 3.5.1 (Font: Pablo Mañas)

Si ara s'activa FC2 s'obté el següent:



**Figura 3.5.7.** Finestra de simulació del GRAFCET 3.5.1 (Font: Pablo Mañas)



A la figura anterior es mostra com s'activa l'etapa 2 degut a la validació de la condició FC2. També es desactiva l'etapa anterior, és a dir l'etapa 1 juntament amb les accions associades que en aquest cas només és "LA" ja que es considera que la acció ja s'ha acabada de manera instantània i, per tant, es valida la transició. En activar FC3 la simulació avançaria de la següent forma:

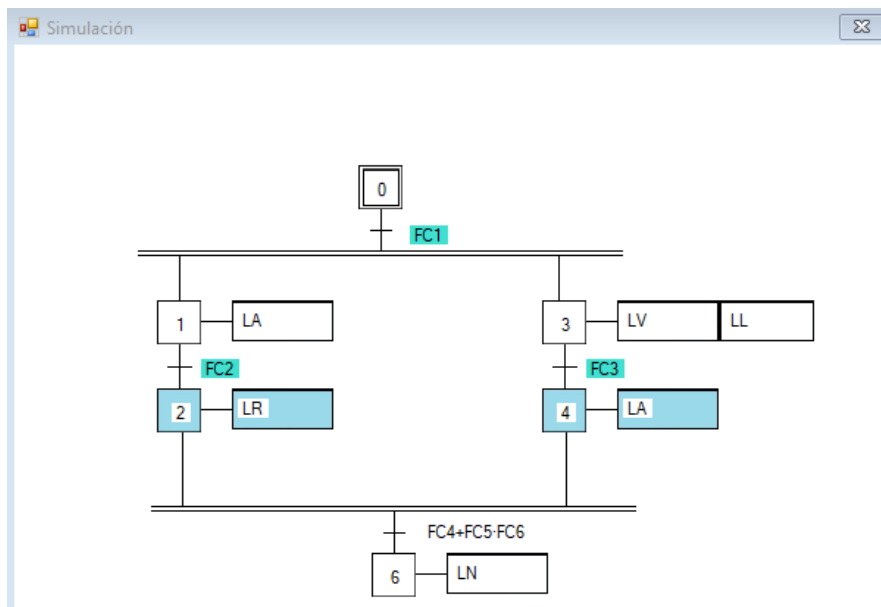


Figura 3.5.8. Finestra de simulació del GRAFCET 3.5.1 (Font: Pablo Mañas)

I, finalment, si s'activés FC4 o FC5 i FC6 el resultat final seria el següent:

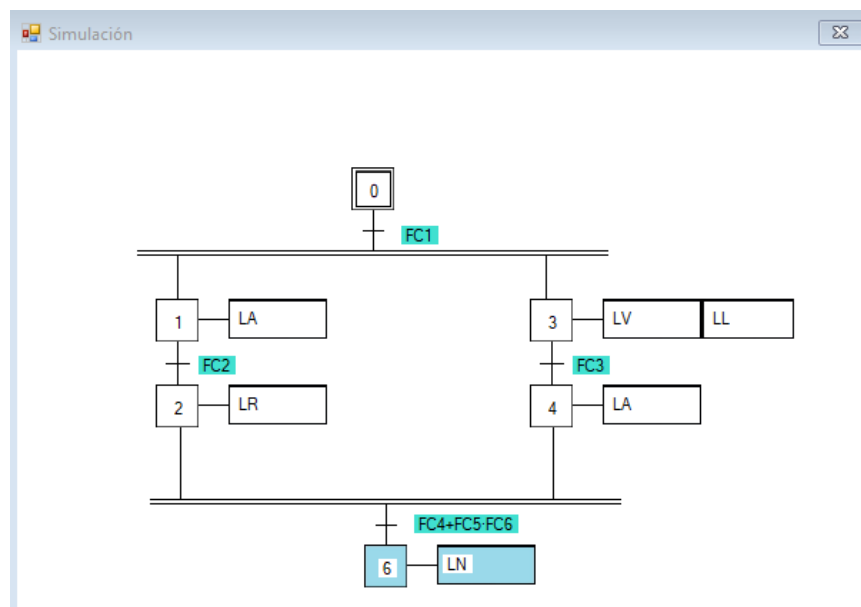
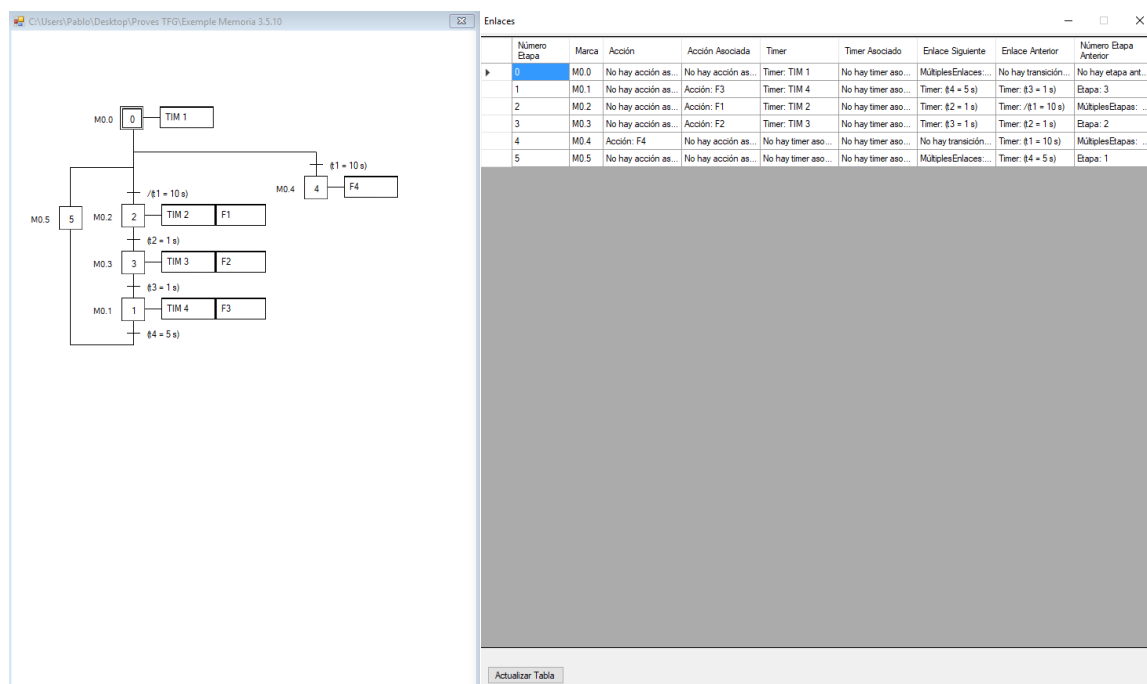


Figura 3.5.9. Finestra de simulació del GRAFCET 3.5.1 (Font: Pablo Mañas)

És necessari afegir que aquest exemple es considera acadèmic i únicament serveix per mostrar com funciona el programa i per tant, no guarda cap relació amb un exemple de control real.

El segon exemple consisteix en la utilització de temporitzadors per demostrar el funcionament d'aquests a la simulació. Així doncs, a continuació es mostra un GRAFCET adequat a aquest propòsit:



**Figura 3.5.10.** Exemple GRAFCET amb temporitzadors i la seva taula d'enllaços (Font: Pablo Mañas)

En aquest cas, únicament s'observen sobre el GRAFCET temporitzadors i accions associades a les etapes. Així doncs, quan l'usuari polsa el botó de simular, la taula d'entrades i sortides generada com no existeixen transicions de nivell, convergències ni divergències, tindrà buida la primera columna perquè el programa no considera les transicions dels temporitzadors com entrades en sí al sistema. Per mostrar aquest fet, a continuació es presenta la captura corresponent de la taula exposada al fer clic sobre simular:

Entradas	Salidas	Transiciones Timer	Timers
	F3 - Q0.0	t1=10s	TIM1
	F1 - Q0.1	t4=5s	TIM4
	F2 - Q0.2	t3=1s	TIM2
	F4 - Q0.3	t2=1s	TIM3

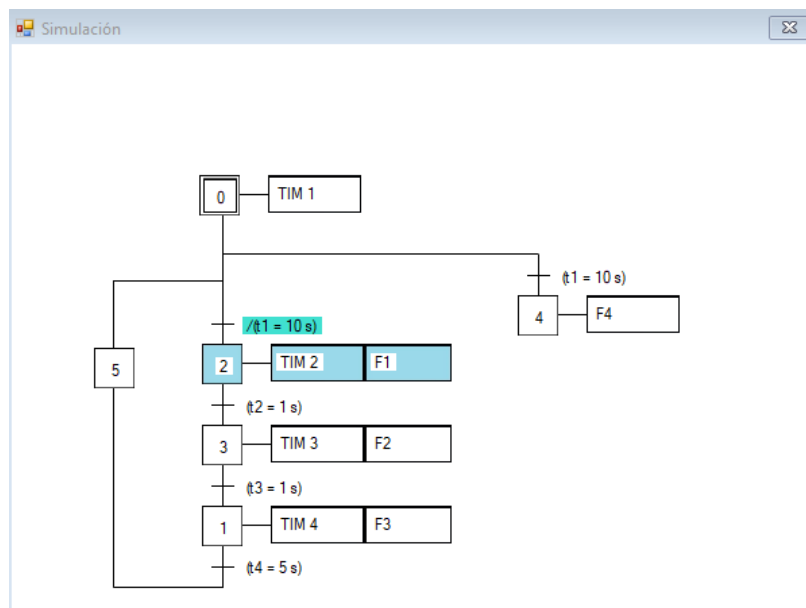
Activar entrada    Desactivar entrada

**Figura 3.5.11.** Taula d’entrades i sortides del sistema del GRAFCET 3.5.1 (Font: Pablo Mañas)

Com es pot observar a la figura anterior, les sortides, transicions i temporitzadors queden desordenats, és a dir, no hi ha una relació entre els elements creats; per exemple la transició del temporitzador 3 hauria de situar-se al costat del seu temporitzador el TIM 3 però no ho fa perquè el programa ordena les transicions, temporitzadors i sortides de forma independent sense tenir en compte els altres. En cas que aquest fet fos un problema, quan s’exporti la taula a Excel, en aquest programa podria canviar l’ordre i ordenar-lo com cregui.

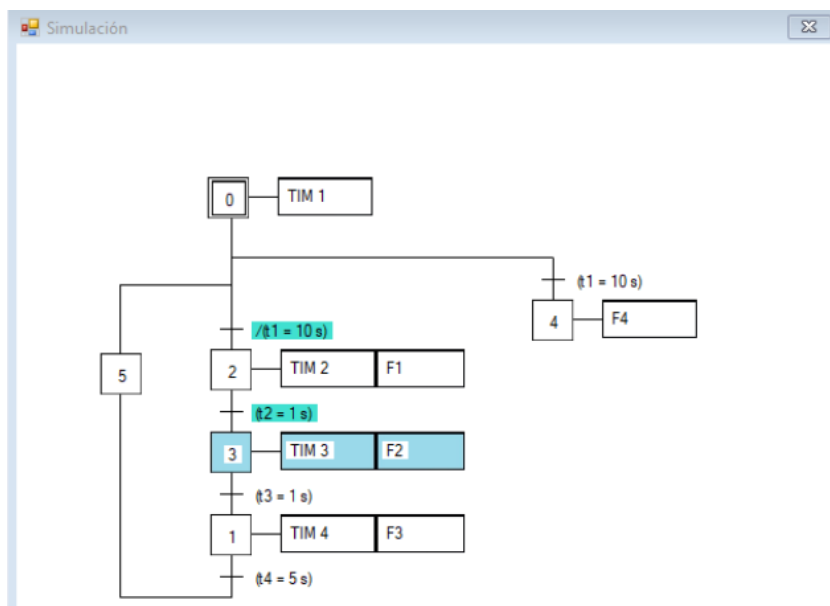
Continuant amb la imatge anterior (figura 3.5.11), al no aparèixer entrades, els botons “Activar entrada” i “Desactivar entrada” no farien res, ja que, com es pot veure no hi ha cap entrada per activar ni desactivar com és lògic.

Analitzant el GRAFCET, un cop es faci clic en simular, inicialment l’etapa activa passarà a ser la 2 ja que el temporitzador 1 no ha transcorregut els 10 segons de transició, el diagrama poc després que s’iniciï s’observa a la imatge següent:



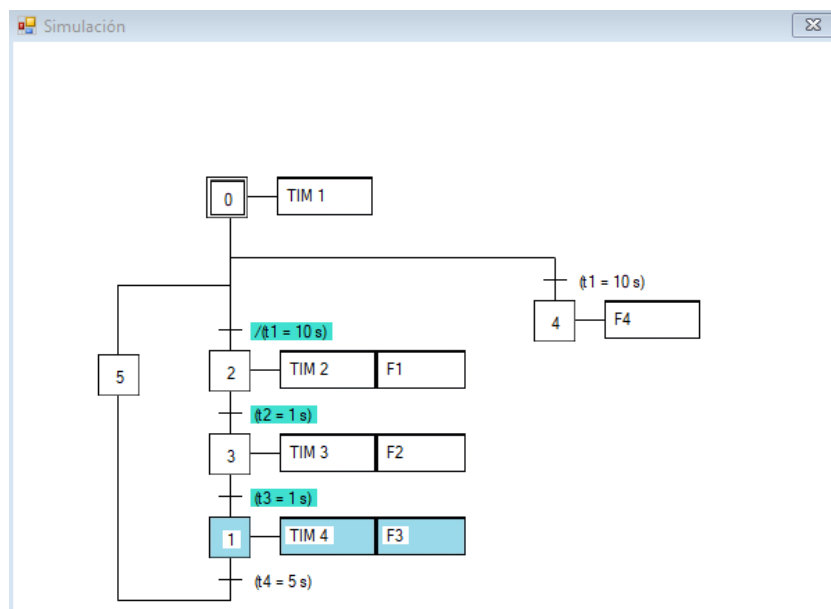
**Figura 3.5.12.** Finestra de simulació del GRAFCET 3.5.10 poc després de l'inici (Font: Pablo Mañas)

En activar-se l'etapa 2, també s'inicia el temporitzador 2 i la acció associada F1, al transcórrer 1 segon (temps establert per TIM 2), aquesta etapa juntament amb la seva acció hauria de desactivar-se i donar pas a la següent etapa. Per tant, a continuació es mostra que succeiria 1 segon després d'haver iniciat la simulació:



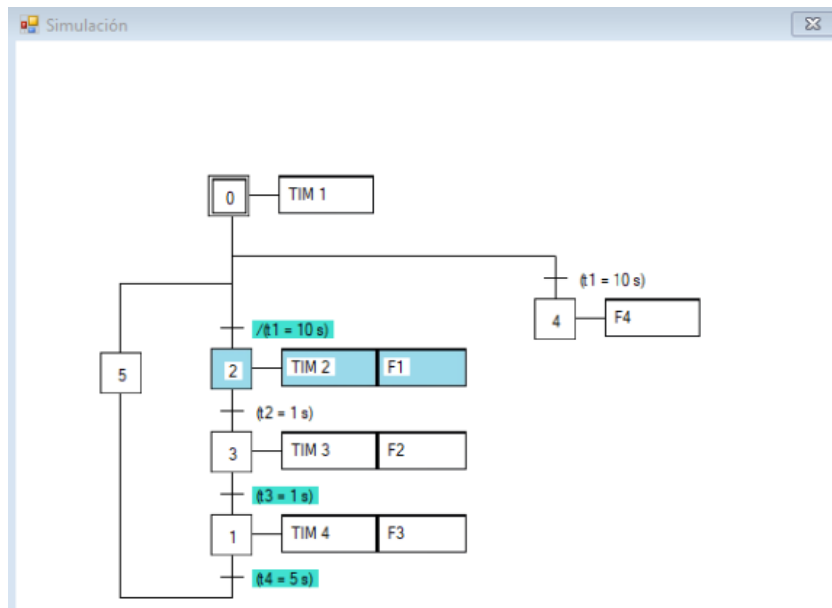
**Figura 3.5.13.** Finestra de simulació del GRAFCET 3.5.10 després d'un segon de l'inici (Font: Pablo Mañas)

Com es pot veure a la imatge anterior, al transcórrer un segon, l'etapa 2 i les seves accions es desactiven i passen lloc a l'etapa 3 ja que el temps establert pel temporitzador 2 ja ha passat. En aquest instant s'activa l'etapa nova i la seva acció associada, també s'estableix un temps d'1 segon així que la figura següent mostra la successió en 1 segon més de simulació:



**Figura 3.5.14.** Finestra de simulació del GRAFCET 3.5.10 després de dos segons de l'inici  
(Font: Pablo Mañas)

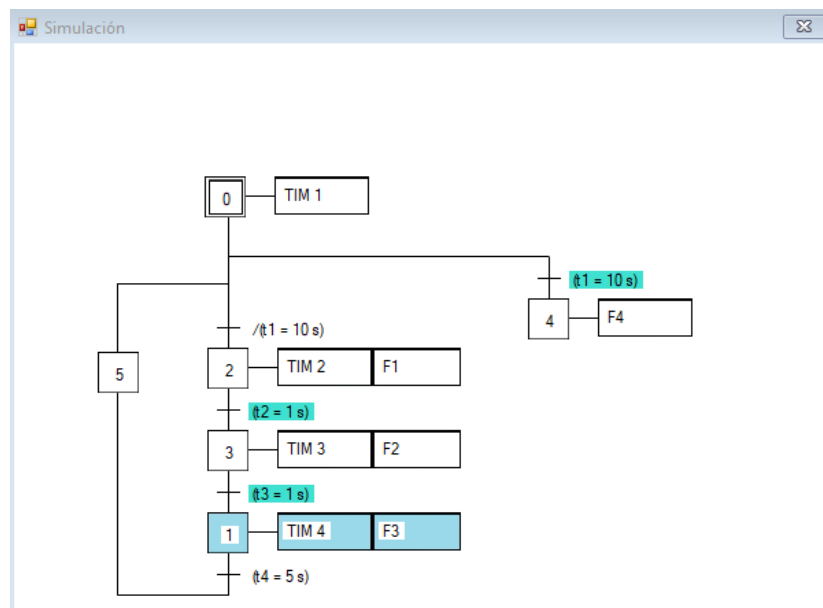
En aquest moment, s'ha desactivat l'etapa 3 i s'ha activat l'etapa 1 que era la següent. Ara, el temporitzador fixat es de 5 segons i la simulació esperarà 5 segons fins que s'activi la transició de temporitzador 4. Un cop passin els 5 segons, s'activarà l'etapa 5 i instantàniament, l'etapa 2 ja que encara no han succeït els 10 segons del temporitzador 1 i, per tant, la transició del temporitzador negat de l'etapa inicial seguirà activa. A continuació es mostra la imatge a l'haver transcorregut 7 segons des de l'inici:



**Figura 3.5.15.** Finestra de simulació del GRAFCET 3.5.10 després de set segons de l'inici  
(Font: Pablo Mañas)

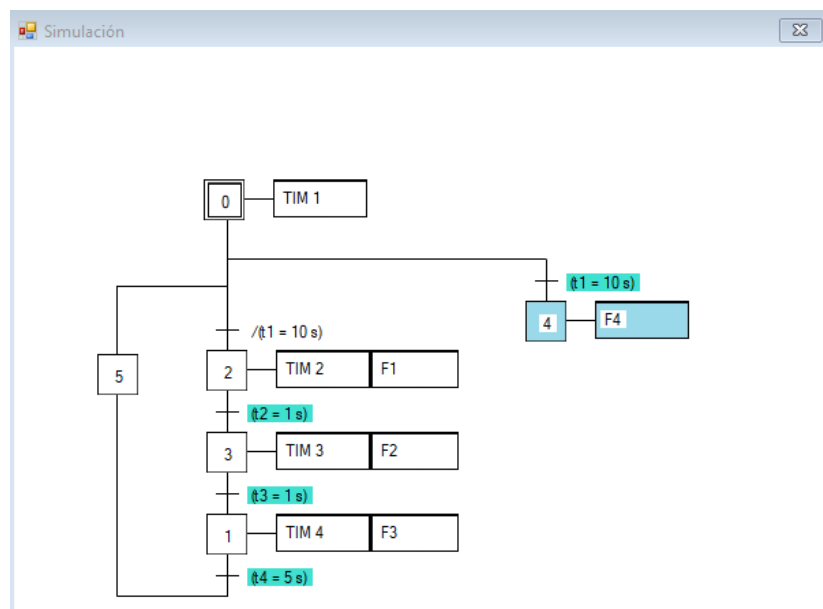
Com ja s'ha dit amb anterioritat, a la figura 3.5.15 es veu com es torna a activar l'etapa 2, el seu temporitzador i la seva acció associada. A partir d'ara, tornarà a produir-se el bucle que s'ha vist inicialment, o sigui, quan es validin els temporitzadors corresponents anirà avançant el diagrama de la mateixa forma ja explicada.

Quan hagin passat 10 segons des de l'inici de la simulació, la transició del temporitzador 1 s'activarà, desactivant així la transició negativa corresponent. En aquest punt (10 segons), calculant els temps, el GRAFCET anirà per l'etapa 1 amb les seves accions activades. La imatge de l'exemple en aquest instant és la següent:



**Figura 3.5.16.** Finestra de simulació del GRAFCET 3.5.10 després de 10 segons de l'inici  
(Font: Pablo Mañas)

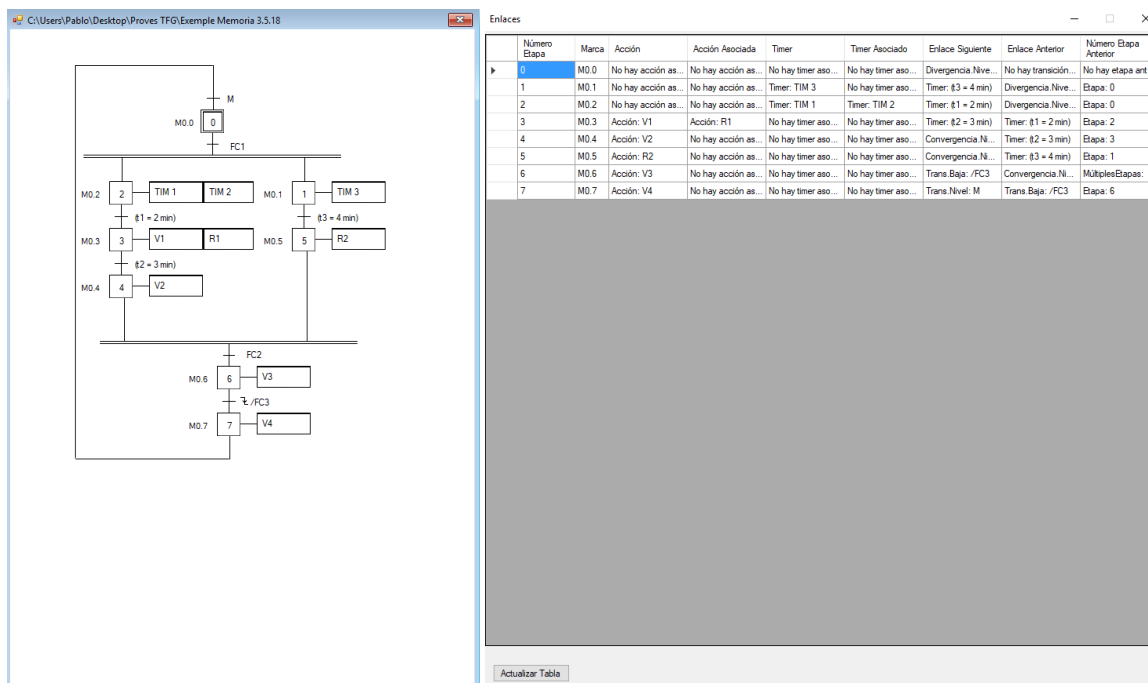
Finalment, quan s'hagi validat per segona vegada el temporitzador 4, com ja s'ha activat la transició del temporitzador 1, s'activarà l'etapa 4 i la seva acció associada. Per tant, l'estat del GRAFCET al termini de la simulació és el que es mostra a la següent captura:



**Figura 3.5.17.** Finestra de simulació del GRAFCET 3.5.10 després de 14 segons de l'inici  
(Font: Pablo Mañas)

Així com l'exemple anterior, aquest tampoc és un exemple real i només pretén mostrar el funcionament dels temporitzadors en les simulacions.

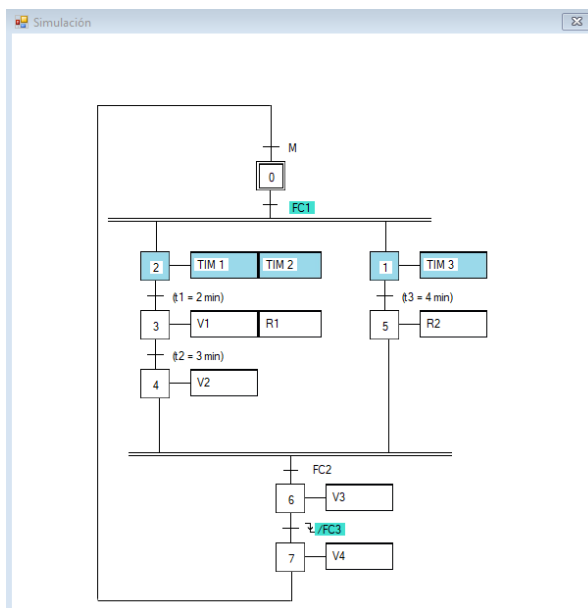
Per acabar de mostrar com funcionen les simulacions, l'últim exemple disposa de tots els elements possibles que poden aparèixer en un GRAFCET utilitzant l'editor. No es tracta d'un exemple real, però ajuda a veure i consolidar el maneig de la simulació. Així doncs, l'últim exemple d'aquest subapartat és el següent:



**Figura 3.5.18.** Exemple GRAFCET amb temporitzadors, convergències i transicions juntament amb la taula d'enllaços (Font: Pablo Mañas)

En aquest exemple es pot observar l'ús de temporitzadors, una convergència en "Y" juntament amb una divergència en "Y", transicions i accions associades a les diferents etapes. Un cop s'iniciï la simulació del GRAFCET, com sempre, es partirà de l'etapa inicial. Com en els exemples anteriors a continuació es mostra com evoluciona el diagrama en activar les diferents entrades del sistema:



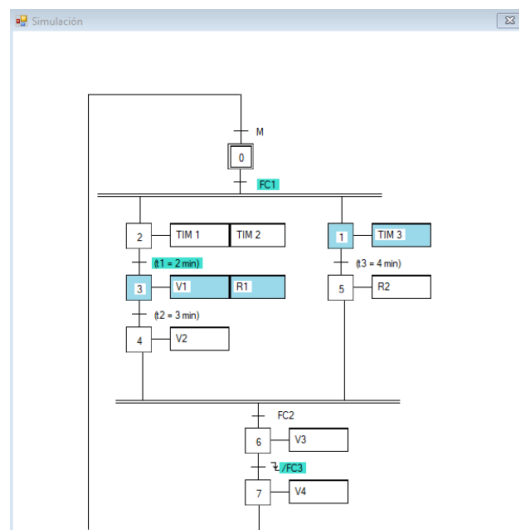


**Figura 3.5.19.** Evolució del GRAFCET 3.5.18. quan s’activa l’entrada FC1 (Font: Pablo Mañas)

En validar-se l’entrada FC1 a través de la taula d’entrades i sortides degut a la divergència en “Y”, s’activen les etapes 1 i 2. Aquestes tenen associades diferents temporitzadors que en complir el seu temps activaran les transicions corresponents. Tal i com es mostra a la figura 3.5.19, la condició /FC3 per flanc de baixada està activa, això es degut a que l’entrada FC3 inicialment està desactivada. Com ja s’ha dit anteriorment, a la simulació els flancs de baixada i pujada no es tenen en compte i es tracten totes les condicions com si fossin de nivell, és a dir, si estan actives s’activen i sinó no, sense tenir en compte l’estat anterior, o sigui, es mantenen en el seu estat sempre i quan no es tornin a activar o desactivar.

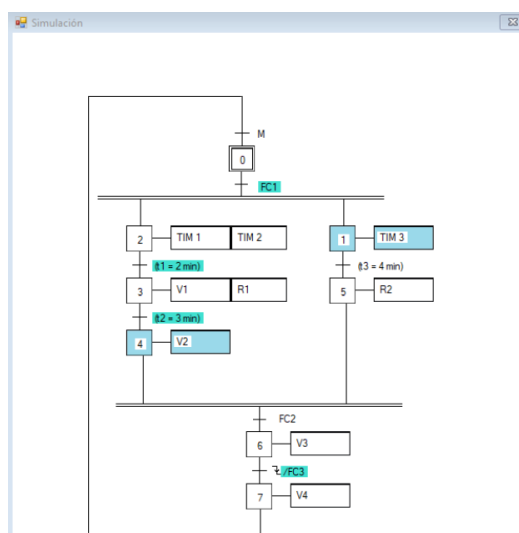
També cal remarcar de nou que la simulació es fa amb segons i no té en consideració l’escala de temps seleccionada per l’usuari quan crea el temporitzador. Això es fa per comoditat, ja que no es creu convenient que la persona que està mirant com funciona hagi d’esperar durant tant de temps per veure si el temporitzador avança o no.

Així doncs, dos segons després, d’activar-se l’entrada FC1, el temporitzador 1 associat a l’etapa 2 ja haurà complert el seu temps indicat i l’avanç del GRAFCET és el següent:



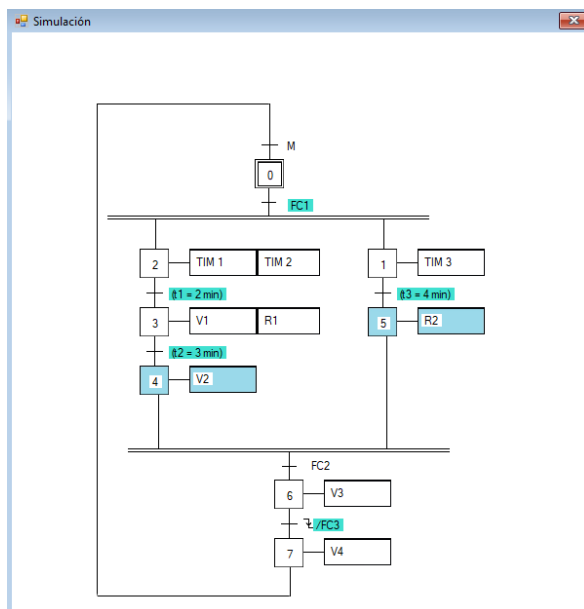
**Figura 3.5.20.** Evolució del GRAFCET 3.5.18. 2 segons (simulació) després de l'activació de FC1  
(Font: Pablo Mañas)

Si això fos portat a la realitat, el diagrama es mantindria en l'estat anterior (figura 3.5.20) durant 1 minut, ja que és el temps en el que el temporitzador 2 de l'etapa 2 s'activaria, ja que haurien transcorregut 2 minuts pel temporitzador 1 i només restaria 1 minut per complir els 3 minuts del segon temporitzador associat. Seguint amb la simulació, en transcórrer un segon més el GRAFCET pateix el següent canvi:



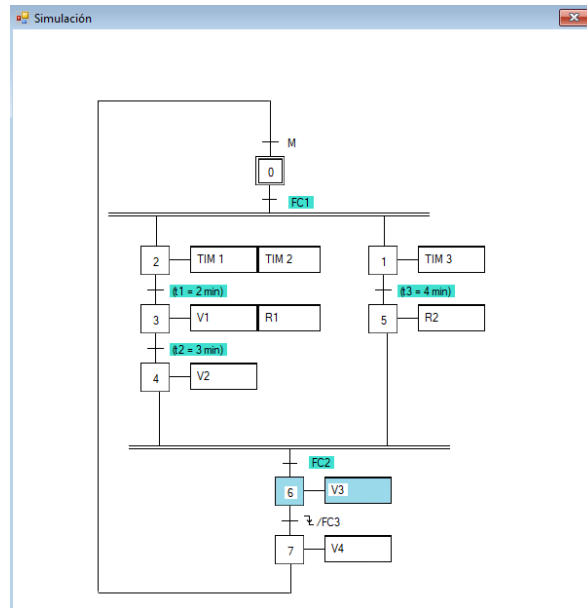
**Figura 3.5.21.** Evolució del GRAFCET 3.5.18. 3 segons (simulació) després de l'activació de FC1  
(Font: Pablo Mañas)

En aquest punt, si es validés la condició FC2, el GRAFCET es mantindria igual que a la figura anterior 3.5.21. ja que la convergència en 1 no ho permetria fins que l'etapa 5 no estigués activa. Per tant, primerament hauria d'esperar a que el temporitzador associat a l'etapa 1, activés la seva transició a l'haver transcorregut els 4 minuts. A continuació es mostra la imatge de la simulació quan ha succeït aquest interval de temps en segons després de l'activació de l'entrada FC1:



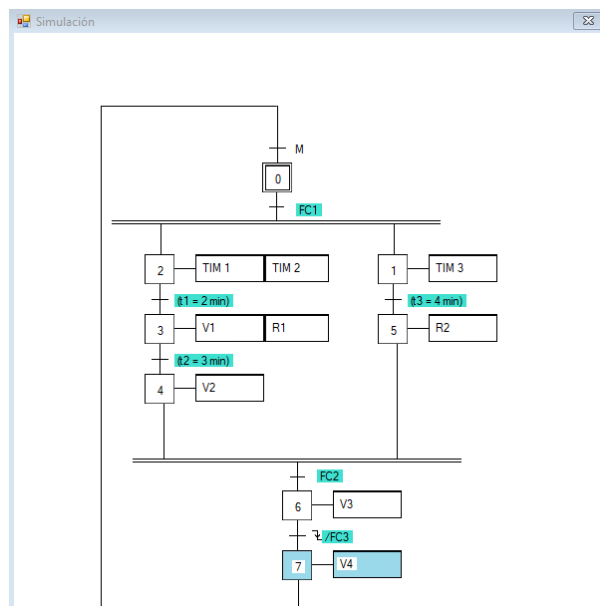
**Figura 3.5.22.** Evolució del GRAFCET 3.5.18. 4 segons (simulació) després de l'activació de FC1  
(Font: Pablo Mañas)

Com l'entrada FC3 està desactivada, es decideix activar-la, ja que sinó, després d'activar FC2 en aquest instant (figura 3.5.22) sobtadament s'activaria l'etapa 7 (en simulació trigaria 0,1 segons aproximadament). Així doncs, quan es desactiva FC3 i s'activa FC2 en aquest ordre, el GRAFCET evoluciona tal i com es mostra a la captura següent:



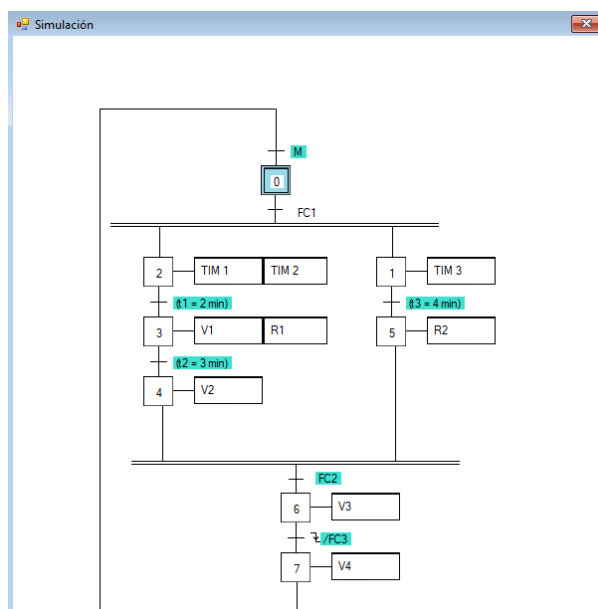
**Figura 3.5.23.** Evolució del GRAFCET 3.5.18. després d'activar FC3 i FC2 i haver transcorregut el temps anterior (Font: Pablo Mañas)

En aquest instant per tal que avanci el diagrama en desactivar de nou FC3 l'etapa 6 juntament amb la seva acció associada passaria a inhabilitar-se i l'etapa 7 amb la seva acció associada s'activarien. A continuació s'observa aquest canvi produït a l'exemple a mode de una imatge de la finestra de la simulació:



**Figura 3.5.24.** Evolució del GRAFCET 3.5.18. en desactivar FC3 (Font: Pablo Mañas)

Finalment, si s'activés l'entrada M del sistema, es tornaria a veure el mostrat anteriorment a la figura 3.5.19 on es tornarien a activar els temporitzadors corresponents i seguiria un bucle. En canvi, si es desactiva primerament l'entrada FC1 i partint del que s'observa a la figura 3.5.24, en el moment en que s'activés l'entrada M el GRAFCET tornaria al seu estat inicial, tal i com es veu reflectit a la imatge següent:



**Figura 3.5.25.** Evolució del GRAFCET 3.5.18. en activar M i desactivar FC1, tornada a l'inici  
(Font: Pablo Mañas)

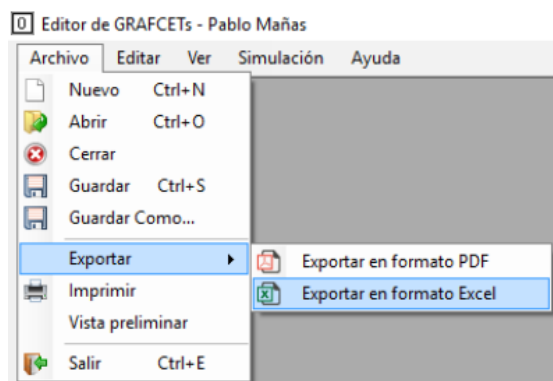
Per concloure aquest subapartat és necessari afegir que prèviament a la simulació és molt important tenir ben construït el GRAFCET i sobre tot la taula d'enllaços, ja que sense aquesta, la simulació no funcionarà correctament i es produiran errors. En cas que es vulgui tancar la simulació, es pot tancar la finestra de simulació fent clic a la creu o amb la seqüència "Simulación" i "Detener".

### 3.6. Exportació en Excel

Microsoft Excel és un programa que permet fer fulls de càlculs i és una eina molt aconsellable en la visualització i anàlisi de dades. Ja que l'editor conté dades importants en sobre els diagrames elaborats en forma de taules, es va decidir en un inici la possibilitat d'exportar aquestes en un full de càlcul del programa Excel.

Ara que ja s'han explicat les dues taules important del programa: la taula d'enllaços del GRAFCET i la taula d'entrades, sortides i temporitzadors del mateix; en aquest subapartat es mostra com fer l'exportació en Excel a través del programa d'aquestes taules en concret.

Partint d'un GRAFCET ja fet com pot ser el mostrat a la figura del subapartat anterior 3.5.18, per exportar-lo caldrà fer clic a “Archivo”, “Exportar” i “Exportar en formato Excel” de la mateixa manera que s’observa a la captura de pantalla següent:



**Figura 3.6.1.** Seqüència “Exportar en formato Excel” (Font: Pablo Mañas)

Un cop es polsi el botó corresponent el programa obre un quadre de diàleg igual al que es pot observar a la figura mostrada en el subapartat 1, la 3.1.7. En aquest instant, l’usuari decideix on guardarà el nou document d’Excel.

Per la manera en la que està programat, en el moment en que s’exporti, si la finestra de simulació no està oberta i els enllaços tampoc, els obrirà. Això ho fa ja que necessita treure les dades d’aquestes taules tant importants i exportar-les a l’Excel en qüestió. Un cop s’exporti aquestes finestres es tancaran.

Si el GRAFCET conté temporitzadors, entrades i sortides a l’obrir la finestra de simulació no mostrarà cap missatge, però si en canvi, un dels 3 o varis no existeixen, a l’hora d’exportar sortiran els missatges d’alerta i caldrà tancar-los en el moment de l’exportació.

A la imatge següent es mostra una captura de l’Excel obtingut quan s’exporta a Excel l’exemple 3.5.18:

Tabla de enlaces:										Tabla de entradas, salidas y temporizadores:			
Número Etapa	Marca	Acción	Acción Asociada	Timer	Timer Asociado	Enlace Siguiente	Enlace Anterior	Número Etapa Anterior	Entradas	Salidas	Transiciones	Timer	Timers
0	MO.0	No hay acción asociada	No hay acción asociada	No hay timer asociado	No hay timer asociado	No hay transición asociada	Trans.Nivel: M	Etapa: 7	M - Q0.0	V1 - I0.0	t3=4min		TIM3
1	MO.1	No hay acción asociada	No hay acción asociada	Timer: TIM 3	No hay timer asociado	Timer: (t3 = 4 min)	Divergencia.Nivel: FC1	Etapa: 0	FC1 - Q0.1	R1 - I0.1	t1=2min		TIM1
2	MO.2	No hay acción asociada	No hay acción asociada	Timer: TIM 1	Timer: TIM 2	Timer: (t1 = 2 min)	Divergencia.Nivel: FC1	Etapa: 0	FC2 - Q0.2	V2 - I0.2	t2=3min		TIM2
3	MO.3	Acción: V1	Acción: R1	No hay timer asociado	No hay timer asociado	Timer: (t2 = 3 min)	Timer: (t2 = 3 min)	Etapa: 2	FC3 - Q0.3	R2 - I0.3			
4	MO.4	Acción: V2	No hay acción asociada	No hay timer asociado	No hay timer asociado	Convergencia.Nivel: FC2	Timer: (t2 = 3 min)	Etapa: 3		V3 - I0.4			
5	MO.5	Acción: R2	No hay acción asociada	No hay timer asociado	No hay timer asociado	Convergencia.Nivel: FC2	Timer: (t3 = 4 min)	Etapa: 1		V4 - I0.5			
6	MO.6	Acción: V3	No hay acción asociada	No hay timer asociado	No hay timer asociado	Trans.Baja: /FC3	Convergencia.Nivel: FC2	MúltiplesEtapas: 5, 4					
7	MO.7	Acción: V4	No hay acción asociada	No hay timer asociado	No hay timer asociado	Trans.Nivel: M	Trans.Baja: /FC3	Etapa: 6					

**Figura 3.6.2.** Captura d’imatge del document Excel exportat de l’exemple 3.5.18 (Font: Pablo Mañas)

Pel que fa al tema de l’exportació en Excel no cal remarcar res més, únicament que en cas que es produeixi algun error o l’Excel on es vol guardar ja estigui obert, es mostrarà un missatge d’alerta de que no es possible accedir a aquest arxiu.

## 3.7. Funcionament de l'aplicació a nivell de programació

Tot i que inicialment no es va plantejar la realització d'aquest apartat, més endavant es va decidir afegir la informació necessària que explica com funciona el programa a nivell de programació, és a dir, com s'han programat les funcions de l'aplicació i com en alguns casos es podria variar i, possiblement, millorar. Per tant, aquest subapartat tracta de desglossar el programa en les diferents classes (veure annex A0) i explicar l'algorítmica, els mètodes i els tipus de dades utilitzats.

### 3.7.1. Menú Principal

Aquesta classe conté totes les funcions referents als diferents botons de la finestra principal la qual s'obre a l'inici del programa. També, controla tots els esdeveniments de ratolí i teclat, això inclou el moviment dels objectes, la cerca d'enllaços descrita anteriorment i la creació de les taules que s'exporten.

#### 3.7.1.1. Variables

Per començar a explicar aquesta part de l'aplicació es poden nomenar tots els atributs que intervenen. Aquests són els que s'observen a l'annex A1 entre les línies 6 i 111. El primer de tots es tracta d'una d'un atribut de tipus panel que consisteix en un panell que contindrà els objectes al seu interior i públic ja que es comparteix entre les diferents classes. El segon és una declaració d'una variable nova que crea una finestra filla de la principal, on es trobarà el panell descrit anteriorment. Les dues variables públiques següents permeten guardar la ruta d'arxiu que l'usuari decideix quan s'obre la finestra de guardat i es guarda el diagrama (figura 3.1.7). Les tres següents permeten emmagatzemar en variables del tipus string (cadena), totes les dades referents a les posicions dels objectes i els objectes al panell. Cal dir que una variable del tipus string com a molt pot contenir  $2^{31}$  caràcters, o sigui, aproximadament 2000 milions i, per tant, en cas que n'hi hagin molts objectes no es creu que hi hagin problemes de memòria ja que per cada objecte com a molt s'ocuparia menys de 100 caràcters. Es tracta de 3 variables diferents ja que una permet guardar les dades actuals, altre les dades prèvies a pulsar l'acció de desfer i l'última amb les dades al fer clic a copiar.

El següent bloc de variables conté dades del tipus booleà que controlen si s'ha fet un guardat, si existeix un arxiu creat, si el timer que es vol crear és del tipus associat o no i si es vol activar l'ajuda de posicionament.

El tercer bloc conté les coordenades dels objectes creats al panell en variables del tipus punt o llistes de tipus punt. Les dues primeres són utilitzades per guardar les posicions del ratolí (x i y) actuals i finals quan l'usuari fa clic sobre el panell i ho mou. Com només n'hi ha una etapa inicial i un panell, únicament les tres primeres es guarden en una variable, la resta de coordenades es guarda en llistes, cadascuna

amb el seu tipus d'objecte. En cadascuna de les llistes es guarden les posicions dels objectes apareguts al panel, les quals coincideixen amb els atributs Top i Left dels objectes en sí. Cal dir que es podrien guardar totes les llistes en una matriu que contingués tota la informació i així reduir línies de codi en les funcions en que s'ha de fer un recorregut per totes. Tot i així, fer-ho resulta una mica incòmode perquè s'ha de redimensionar la matriu cada cop que s'introdueix un nou valor a una de les llistes de l'interior i finalment, es va descartar la idea.

Seguint amb l'ordre d'adalt abaix es troben 4 comptadors inicialitzats amb els seus valors corresponents que serveixen per comptar les etapes que han sigut creades, els números mostrats a les marques de les etapes, els temporitzadors i una variable que restringeix el número de rectangles que s'observen al panell quan es vol fer una selecció, ja que únicament interessaria que fos 1 únic rectangle que s'anès fent gran en estirar el ratolí.

El següent bloc conté les variables que guarden els objectes de l'etapa inicial i les etapes, juntament amb l'etiqueta que es situa a l'interior de l'etapa inicial i la seva marca. Les dos primeres variables són de tipus PictureBox i permeten guardar la imatge del tipus d'objecte, en aquest cas, la imatge de l'etapa inicial i l'etapa normal i controlar-la. Les altres dues són de tipus Label i únicament mostren un text en una posició concreta que normalment es relaciona amb la posició de l'objecte al qual pertanyen.

A mode similar al bloc anterior, les següents variables guarden tot el relacionat amb les etapes. Com pot haver més d'1 etapa al panell totes les variables són llistes dels tipus anteriors i contenen la informació de totes les etapes (marca, etiqueta central i imatge).

Els següents tres bloc pertanyen a les imatges dels flancs de pujada i baixada que l'usuari pot escollir en els transicions, convergències i divergències. Tot i que tots aquests grups compartien la mateixa imatge de flanc de pujada i baixada es va decidir separar en llistes amb la fi que cadascú dels flancs estigués associat al seu tipus d'objecte.

Continuant en aquesta línia, el proper bloc consisteix en la llista d'etiquetes de cada tipus d'element creat. Això engloba la llista d'etiquetes de cada tipus de transició, convergència, divergència, temporitzador, acció i etiquetes que crea l'usuari.

Per últim es troben les definicions de les llistes que contindran informació amb les etiquetes i els objectes de la simulació, per no haver d'utilitzar les llistes dels objectes ja creats al panell i separar-les de les utilitzades a la simulació ja que aquestes es manipulin únicament en la simulació i no afectin al diagrama que l'usuari estigui creant en aquell moment.



### 3.7.1.2. Procediments i funcions

Ara que ja s'han explicat totes les diferents variables que apareixen en un primer lloc a la classe del menú principal, a continuació es parla de les funcions o mètodes d'aquesta classe. En cas que la funció sigui molt similar o idèntica, es deixarà constància de les variacions, però sense tornar a repetir les coses ja explicades.

#### 3.7.1.2.1 Procediment GuardarDatos

El primer procediment que apareix seguint l'ordre anterior és el que té com a títol GuardarDatos, ja que a cada moviment d'un objecte o a qualsevol acció es necessita actualitzar les dades sobre la variable DatosGuardados es va decidir crear una funció que únicament truqués al procediment que permet sobreescrivre les dades sobre la variable i habilités l'opció de desfer i deshabilités l'opció de refer. Així doncs aquest mètode únicament fa això i es va crear per reduir línies de codi.

#### 3.7.1.2.2 Esdeveniments de teclat

La següent funció controla els events de teclat que permeten moure els objectes, esborrar i modificar en alguns casos. La seva capçalera és: `Protected Overrides Function ProcessCmdKey(ByRef msg As System.Windows.Forms.Message, ByVal keyData As System.Windows.Forms.Keys) As Boolean`. En primer lloc és de tipus protegit ja que únicament permet actuar dintre de la classe i no fora d'ella. També disposa del modificador Overrides amb la fi que l'usuari quan premi una tecla es faci aquella funció invalidant totes les altres possibles accions produïdes al fer l'acció. El primer paràmetre consisteix en un missatge que es retorna i que és modificable dintre de la funció ja que és comparteix per referència (ByRef), el segon paràmetre té relació amb la tecla que l'usuari prem i la qual rep la funció. Per últim la funció retorna un valor False o True en funció de si la tecla que s'ha premut cal o no avaluar-la, és a dir, si quan es polsa una de les tecles necessàries per moure i hi ha un objecte seleccionat, caldrà fer un return True i en els altres casos retornar False, és a dir, si es prem una altra tecla o alguna tecla sense que hagi res seleccionat. Ara que ja s'ha explicat més o menys com funciona en sí al funció, es pot explicar com s'ha programat perquè permeti els esdeveniments de teclat. Perquè el lector no hagi de mirar a l'annex, a continuació es mostra una imatge amb el retall de les estructures importants dintre de la funció:

```

120     'Función que contiene los eventos de teclado:
121     Protected Overrides Function ProcessCmdKey(ByRef msg As System.Windows.Forms.Message, ByVal keyData As
System.Windows.Forms.Keys) As Boolean
122     'Si se pulsa la flecha derecha del teclado:
123     If keyData = Keys.Right Then
124         If Me.Contains(Panel1) Then
125             If Me.Panel1.Contains(etapa0) Then
126                 'En caso de que la etapa0 esté seleccionada:
127                 If etapa0.Tag = True Then
128                     GuardarDatos() 'Guarda la posición anterior del objeto
129                     'Desplaza el objeto junto con sus etiquetas:
130                     etapa0.Left += 1
131                     Label0.Left += 1
132                     LabelMarca0.Left += 1
133                     'Limita el objeto para que no salga del panel que lo contiene:
134                     If (etapa0.Left + etapa0.Width) >= Panel1.Width Then
135                         etapa0.Left = Panel1.Width - etapa0.Width
136                         Label0.Left = etapa0.Left + 9
137                         LabelMarca0.Left = etapa0.Left - 35
138                     End If
139                 End If
140             End If
141         'Se repite el proceso anterior para todos los objetos existentes:
142         For ind = 0 To listaEtapa.LongCount - 1
143             If listaEtapa(ind).Tag = True Then
144                 GuardarDatos()
145                 listaEtapa(ind).Left += 1
146                 listaLabsEtapa(ind).Left += 1
147                 listaMarcaLabsEtapa(ind).Left += 1
148                 If (listaEtapa(ind).Left + listaEtapa(ind).Width) >= Panel1.Width Then
149                     listaEtapa(ind).Left = Panel1.Width - listaEtapa(ind).Width
150                     listaLabsEtapa(ind).Left = listaEtapa(ind).Left + 9
151                     listaMarcaLabsEtapa(ind).Left = listaEtapa(ind).Left - 35
152                 End If
153             End If
154         Next ind

```

**Figura 3.7.1.2.2.1.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

A la línia 123 de la figura anterior es veu una estructura del tipus condicional, en cas que l'usuari hagi pulsat la tecla de la fletxa dreta es farà tot el que està adintre d'aquest If. Per evitar problemes s'aniden una sèrie de condicions, en primer lloc cal que existeixi el panell (línia 124) perquè hi pugui haver algun moviment d'objecte. En el cas de l'etapa inicial cal que a dintre del panell hi aparegui una etapa inicial (línia 125). Per últim, l'objecte ha d'estar seleccionat per l'usuari (atribut Tag de l'objecte igual a True), més endavant es parlarà exactament com es fan les seleccions. Abans de fer res, és a dir, moure l'objecte o objectes seleccionats, es guarda tot el diagrama cridant al procediment ja explicat anteriorment que guarda les dades, això es fa amb la fi de guardar abans del moviment per si després es vol tornar enrere. Després de guardar es mou l'objecte i les etiquetes associades amb ell que sempre mantindran la mateixa distància. En el cas que l'etapa inicial (línia 134) es pugui sortir del panell, es delimita i es manté dintre del panell perquè en cas que l'usuari mogui algun objecte fora no pugui fer-ho ja que sinó no el podria tornar a seleccionar amb el ratolí mai més perquè el panell i la finestra sempre mantenen les seves dimensions. En el cas que les dades dels objectes es trobin en llistes, caldrà fer un recorregut com en el cas de les etapes (línia 142). En aquests casos, les llistes d'etiquetes i marques tenen el mateix tamany ja que s'omplen i s'eliminen al mateix temps, per això no es recorren totes les llistes sinó només les dels objectes en sí. En el cas que s'haguessin guardat totes les variables de les llistes en una sola matriu, s'hauria de recórrer la matriu però fixant-se cada cop en l'element que estigui seleccionat ja que podria ser problemàtic a l'hora de moure les etiquetes relacionades amb els objectes. De la manera en que està programat es recorren totes les llistes d'objectes i es mouen les etiquetes o objectes corresponents en cada moment. Al final d'aquest apartat de la funció es troba una trucada al procediment ActualizarObjetos que s'explica més endavant i un Return True perquè es pugui fer l'acció.

Tot el paràgraf anterior que explica únicament el moviment de la fletxa dreta es repeteix per la fletxa esquerra, amunt i avall. L'única característica que difereix és el moviment, per exemple la fletxa esquerra resta 1 píxel a la posició Left de l'objecte i la fletxa avall sumaria 1 a l'atribut Top. També canvia la restricció dels límits del panell que mantenen els objectes dintre del full.

La resta de tecles a considerar s'observen a les següents línies de codi:

```

1181 | 'Lo mismo que anteriormente pero si se pulsa la tecla - las lineas se hacen más cortas.
1182 | If keyData = Keys.Subtract Or keyData = Keys.OemMinus Then
1183 |     If Me.Contains(Panel1) Then
1184 |         For ind = 0 To LineaForm.listaLineaVertical.LongCount - 1
1185 |             If LineaForm.listaLineaVertical(ind).Tag = True Then
1186 |                 GuardarDatos()
1187 |                 LineaForm.listaLineaVertical(ind).Height() -= 1
1188 |             End If
1189 |         Next ind
1190 |         For ind = 0 To LineaForm.listaLineaHorizontal.LongCount - 1
1191 |             If LineaForm.listaLineaHorizontal(ind).Tag = True Then
1192 |                 GuardarDatos()
1193 |                 LineaForm.listaLineaHorizontal(ind).Width() -= 1
1194 |             End If
1195 |         Next ind
1196 |         ActualizarObjetos()
1197 |         Return True
1198 |     End If
1199 |
1200 | 'Si el usuario pulsa supr o la flecha para borrar:
1201 | If keyData = Keys.Delete Or keyData = Keys.Back Then
1202 |     If Me.Contains(Panel1) Then
1203 |         GuardarDatos()
1204 |         EliminarSeleccion() 'Se eliminan todos los objetos seleccionados
1205 |         ActualizarObjetos()
1206 |         Return True
1207 |     End If
1208 | End If
1209 | Return MyBase.ProcessCmdKey(msg, keyData)
1210 | End Function

```

**Figura 3.7.1.2.2.2.** Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

A la primera imatge de la figura anterior s'observa el que succeeix quan es polsa sobre la tecla – del teclat que seria el mateix que la tecla + però modificant l'actualització d'algun atribut. La tecla – únicament serveix per reduir les línies verticals i horitzontals que serveixen per fer els enllaços com ja s'ha explicat al subapartat 3.3. Així doncs únicament modifiquen els objectes de les llistes corresponents, en concret, en el cas de les línies verticals l'alçada (atribut Height) i en el cas de les línies horitzontals l'amplada (atribut Width). Com en el cas anterior, l'objecte ha d'estar seleccionat i prèviament es guarden les dades anteriors a l'acció. La tecla + seria l'anàloga però incrementant els atributs anteriors en comptes de reduir-los. A la segona imatge es mostra el que succeeix quan es prem el botó de suprimir o la tecla "BackSpace". En resum es sobreescrueixen les dades, es truca al procediment que elimina els objectes seleccionats EliminarSeleccion del qual es parla més endavant i al procediment ActualizarObjetos també important. Per últim, a la línia 1209 s'observa un procés que retorna fals en cas que la tecla que s'hagi premut no faci res o no hagi de fer res, per evitar que quan es premin altres tecles falli el programa. Com totes les funcions, finalitza amb un End Function.

### 3.7.1.2.3 Procediment Form1\_Load

Aquest procediment únicament s'executa quan s'inicia el programa y tan sols serveix per deshabilitar tota la sèrie de botons que no tindria sentit que estiguessin activats a l'iniciar el programa. Per això

dintre del procés es truca a altre procediment que s'anomena Deshabilitar i el qual es presenta a continuació.

#### 3.7.1.2.4 Procediment Deshabilitar

Aquest procediment consisteix en reiniciar tots els comptadors necessaris i deshabilitar tots els botons i caselles convenients. Així doncs, es torna a donar el valor inicial a les variables dels comptadors explicades a l'anterior subapartat i es crea una llista amb tota la sèrie de botons i caselles la qual es recorre i es modifica de cada element l'atribut Enabled a qual permet que es pugui clicar a False perquè ja no es permeti aquesta acció. La manera en que s'ha programat es mostra a la següent imatge:

```

1219     Public Sub Deshabilitar()
1220         'Se reinician los contadores, por si acaso
1221         contador = 1
1222         contador2 = 0
1223         contador3 = 0
1224         contadorTimer = 1
1225         SimulacionForm.SimulacionActivada = False
1226         'Definición de lista que contiene todos aquellos elementos a deshabilitar:
1227         Dim ListaDeshabilitarBool = (DesactivarAyudaPosicionamientoToolStripMenuItem, CerrarToolStripMenuItem,
GuardarToolStripMenuItem, GuardarComoToolStripMenuItem, ExportarToolStripMenuItem, ImprimirToolStripMenuItem,
VistaPreliminarToolStripMenuItem, DeshacerToolStripMenuItem, RehacerToolStripMenuItem, CortarToolStripMenuItem,
CopiarToolStripMenuItem, PegarToolStripMenuItem, SeleccionarTodoToolStripMenuItem, VerEnlacesToolStripMenuItem,
MarcasToolStripMenuItem, SimularToolStripMenuItem, DetenerToolStripMenuItem, BotonEtapa0, BotonEtapa, BotonTransicion,
BotonAccion, BotonAccionAsociada, BotonEtiqueta, BotonEnlace, BotonConvergencia, BotonDivergencia, BotonTimer,
BotonTimerAsociado)
1228         For ind = 0 To ListaDeshabilitarBool.LongCount - 1 'Se recorre la lista anterior
1229             ListaDeshabilitarBool(ind).Enabled = False 'Se deshabilita el elemento de la lista
1230         Next ind
1231     End Sub

```

Figura 3.7.1.2.4.1. Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

#### 3.7.1.2.5 Procediment ConstruirEtapa0

Dintre d'aquest mètode es crea l'objecte de l'etapa inicial del tipus PictureBox amb les seves característiques inicials, s'associa els events de ratolí que s'executaràn quan es faci clic, es mogui o es solti el ratolí i per últim s'afegeix al panell aquesta etapa inicial. El codi corresponent és el següent:

```

1233     'Función que crea la etapa0
1234     Private Sub ConstruirEtapa0()
1235         'Se crea la etapa0 y sus características iniciales:
1236         etapa0 = New PictureBox With {
1237             .Image = Image.FromFile("Imagenes\Etapa0.jpg"), 'Imagen de la etapa0
1238             .SizeMode = PictureBoxSizeMode.StretchImage, 'Adaptación de la imagen
1239             .Width = 30, 'Tamaño ancho etapa0
1240             .Height = 30, 'Tamaño alto etapa0
1241             .Left = 100, 'Posición x inicial
1242             .Top = 100, 'Posición y inicial
1243             .Cursor = Cursors.SizeAll, 'Cursor del tipo SizeAll en cuanto se pasa el ratón sobre la etapa0
1244             .Tag = False 'Se utiliza el tag para la selección del objeto, inicialmente False
1245         } 'Creación etapa0
1246         'En las demás construcciones se emplea el With {.Atributos = } en una sola línea para definir todas las
características y reducir líneas de código
1247         'Declaración de funciones con los eventos de ratón:
1248         AddHandler etapa0.MouseDown, AddressOf Etapa0MouseDown
1249         AddHandler etapa0.MouseMove, AddressOf Etapa0MouseMove
1250         AddHandler etapa0.MouseClick, AddressOf Etapa0MouseClick
1251         Me.Panel1.Controls.Add(etapa0) 'Se añade la etapa0 al panel
1252     End Sub

```

Figura 3.7.1.2.5.1. Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

Els esdeveniments de ratolí es programen utilitzant l'instrucció AddHandler la qual associa l'event del ratolí amb un procediment quan es fa clic inicialment sobre l'etapa inicial (etapa0.MouseDown) el programa directament fa la funció que s'anomena Etapa0MouseDown la qual s'explica més endavant.

#### 3.7.1.2.6 Procediment ConstruirTextoEtapa0 i procediment ConstruirTextoMarcaEtapa0

Aquí es construeix l'etiqueta on es mostra el número 0 a l'interior de l'etapa inicial i la marca M0.0. Així com en el procediment anterior es crea un nou element en aquest cas de tipus Label que es guarda a la variable corresponent ja declarada anteriorment. En aquest cas algunes de les característiques depenen de la posició de l'etapa inicial ja que es construeixen al mateix temps i s'han de relacionar sempre unides. També es modifica la mida de la font i després d'afegir-se al panell es porten al front d'aquest perquè es puguin mostrar, d'altre manera no es veurien perquè la pròpia etapa inicial el taparia ja que es situaria la imatge per sobre del número. Cal dir que en aquest cas no tenen cap esdeveniment de ratolí associat ja que es consideren etiquetes que no es poden modificar. A la imatge anterior es pot mostrar el codi d'aquests dos procediments.

```

1254     'Función que construye el texto interior de la Etapa0:
1255     Private Sub ConstruirTextoEtapa0()
1256         Label0 = New Label With { .AutoSize = True, .Text = "0", .Left = etapa0.Left + (etapa0.Width() / 3), .Top = etapa0.Top
+ (etapa0.Height() / 3), .BackColor = Color.Transparent}
1257         Label0.Font = New Font(Label0.Font.Name, 8)
1258         Me.Panel1.Controls.Add(Label0) 'Se añade la etiqueta al panel
1259         Label0.BringToFront() 'Se expone en el frente
1260     End Sub
1261
1262     'Función que construye el texto de la marca, igual que la función anterior pero la posición y el texto son diferentes:
1263     Private Sub ConstruirTextoMarcaEtapa0()
1264         LabelMarca0 = New Label With { .AutoSize = True, .Text = "M0.0", .Left = etapa0.Left - 1.2 * etapa0.Width(), .Top =
etapa0.Top + (etapa0.Height() / 3), .BackColor = Color.Transparent}
1265         LabelMarca0.Font = New Font(LabelMarca0.Font.Name, 7)
1266         Me.Panel1.Controls.Add(LabelMarca0)
1267     End Sub
    
```

Figura 3.7.1.2.6.1. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

### 3.7.1.2.7 Esdeveniments de ratolí (general)

En el subapartat anterior (3.7.1.2.5) s'ha explicat la forma en que es direccionen els esdeveniments de ratolí cap als procediments referents de l'etapa inicial. En aquesta part de la memòria s'expliquen els procediments que s'efectuen dintre d'aquests procediments. Com són esdeveniments que es repeteixen en cadascun dels tipus d'objectes s'ha decidit explicar un parell per veure el funcionament. En el cas que es vulgui veure més informació, a l'annex A1 es troba tot el codi de l'aplicació. Per començar, les funcions bàsiques que permet el programa amb el ratolí són moure els objectes amb el clic esquerre i arronsant i canviar noms a etiquetes fent doble clic. A continuació es mostren les imatges corresponents al codi de les funcions de fer clic esquerre sobre un objecte, més concret sobre l'etapa inicial.

```

1269     'Función evento de ratón al clicar sobre un objeto:
1270     Private SubMouseDownObjeto(ByVal sender As Object, ByVal ruta As String)
1271         GuardarDatos()
1272         sender.Tag = True 'Al hacer clic se selecciona, por tanto, el tag es True
1273         Dim p As PictureBox = DirectCast(sender, PictureBox) 'Definición del objeto que se está clicando
1274         p.Image = Image.FromFile(ruta) 'Se cambia la imagen del objeto que se clicca
1275         ActualizarObjetos() 'Se actualizan los objetos
1276         GuardarCoords() 'Se guardan las coordenadas
1277     End Sub
1278
1279     'Función en el momento que el usuario hace clic sobre la etapa0:
1280     Private Sub Etapa0MouseDown(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs)
1281         MouseDownObjeto(sender, "Imágenes\Etapa0se1.jpg")
    
```

```

1282     End Sub
1283
1284     'Funció durant el usuario tiene mantenido el click y lo mueve:
1285     Private Sub Etapa0MouseMove(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs)
1286     If e.Button = Windows.Forms.MouseButtons.Left Then 'si el usuario está clicando:
1287     sender.Tag = True 'De nuevo se define que la etapa0 se está seleccionando
1288     Dim p As PictureBox = DirectCast(sender, PictureBox)
1289     p.Image = Image.FromFile("Imagenes\Etapa0Sel.jpg") 'Se cambia la imagen
1290     MoverObjetos() 'Se mueven los objetos
1291     End If
1292 End Sub
1293
1294 'Función que se utiliza para fijar las etiquetas del objeto a su posición:
1295 Private Sub AjusteLabsYMarcaEtapa0()
1296 Label0.Left = etapa0.Left + (etapa0.Width() / 3)
1297 Label0.Top = etapa0.Top + (etapa0.Height() / 3)
1298 LabelMarca0.Left = etapa0.Left - 1.2 * etapa0.Width()
1299 LabelMarca0.Top = etapa0.Top + (etapa0.Height() / 3)
1300 End Sub
1301
1302 'Función cuando el usuario suelta el clic del ratón:
1303 Private Sub Etapa0MouseClicked(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs)
1304 etapa0.Tag = True 'De nuevo se define que la etapa0 se está seleccionando
1305 Dim p As PictureBox = DirectCast(sender, PictureBox)
1306 p.Image = Image.FromFile("Imagenes\Etapa0Sel.jpg") 'Se cambia la imagen
1307 'Se recorren todos los demás objetos para situar la etapa0 de forma correcta respecto a ellos, es una herramienta para
ayudar al usuario a colocar los objetos de forma correcta.
1308 If AyudaPosicionamiento = True Then
1309     For transicion = 0 To TransicionesForm.listaTrans.LongCount - 1
1310     If etapa0.Top >= TransicionesForm.listaTrans(transicion).Top And etapa0.Top <=
TransicionesForm.listaTrans(transicion).Left And etapa0.Left <= TransicionesForm.listaTrans(transicion).Left +
TransicionesForm.listaTrans(transicion).Width() Then
1311     etapa0.Top = TransicionesForm.listaTrans(transicion).Location.Y +
TransicionesForm.listaTrans(transicion).Height()
1312     etapa0.Left = TransicionesForm.listaTrans(transicion).Location.X
AjusteLabsYMarcaEtapa0()
1313     End If
1314     If etapa0.Top <= TransicionesForm.listaTrans(transicion).Top And etapa0.Top + etapa0.Height() >=
TransicionesForm.listaTrans(transicion).Top And etapa0.Left >= TransicionesForm.listaTrans(transicion).Left And etapa0.Left <=
TransicionesForm.listaTrans(transicion).Left + TransicionesForm.listaTrans(transicion).Width() Then
1315     etapa0.Top = TransicionesForm.listaTrans(transicion).Location.Y - etapa0.Height()
1316     etapa0.Left = TransicionesForm.listaTrans(transicion).Location.X
AjusteLabsYMarcaEtapa0()
1317     End If
1318     Next transicion
1319 End If
1320

```

**Figura 3.7.1.2.7.1.** Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

A la primera imatge de la figura anterior es veu el procediment al qual es dirigeix el programa quan l'usuari fa clic sobre l'etapa 0. Es declara el mètode de la mateixa forma que s'havia fet l'adreça amb l'AddHandler i s'utilitzen dos paràmetres els quals no es modifiquen perquè els arguments es passen per valor. El primer etiquetat com sender farà referència a l'objecte que s'hagi fet clic i el segon als esdeveniments de ratolí. A l'etapa inicial en quant es fa clic (MouseDown) dintre del procediment es truca a altre procediment anomenat MouseDownObjeto el qual es troba immediatament a munt pel que fa al codi. Dintre d'aquesta nova funció la qual demana dos arguments que són un objecte (sempre és el sender anterior) i una cadena que contengui la ruta de la imatge de l'objecte seleccionat. Abans de fer res, sempre es guarden les dades, després es modifica l'atribut tag de l'objecte a True perquè s'ha seleccionat i per tant s'utilitza aquest atribut per determinar-ho. En cas que no existís un atribut similar, s'hauria de crear una nova classe que heretés de PictureBox tot i afegir el nou atribut amb la informació necessària. Un cop s'ha modificat l'atribut tag es declara una nova variable amb el procés DirectCast el qual permet canviar la imatge de l'element que s'ha seleccionat modificant l'atribut corresponent Image. A continuació es truca als procediments de ActualizarObjetos i GuardarCoords.

Pel fet que tots els objectes del tipus PictureBox quan els cliquen actuen de la mateixa manera, es va crear la funció anteriorment explicada ja que només caldrà canviar la ruta que s'envia per paràmetre per cada tipus d'objecte a l'esdeveniment de ratolí concret.

A la segona imatge de la mateixa figura anterior es troba l'esdeveniment quan l'usuari mou el ratolí sobre l'objecte, en aquest cas sobre l'etapa inicial. Així doncs, cal assegurar-se que s'ha fet un clic o s'estigui clicant per poder moure l'objecte i, per això s'utilitza la condició de la línia 1286 la qual detecta

això mateix. Després es fan les mateixes accions que en el procediment explicat anterior però en comptes d'actualitzar objectes i guardar coordenades es mouen els objectes (al subapartat següent s'expliquen aquestes funcions). En el cas que es tracti de moure la resta d'objectes, és a dir, tots els que no són etapes inicials, s'utilitza aquest procediment:

```

1529 'Función al mover cualquier objeto:
1530 Private Sub MouseMoveObjeto(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs, ByVal lista As List(Of
PictureBox), ByVal ruta As String)
1531 For ind = 0 To lista.LongCount - 1
1532 If e.Button = Windows.Forms.MouseButtons.Left Then
1533 sender.tag = True
1534 Dim p As PictureBox = DirectCast(sender, PictureBox)
1535 p.Image = Image.FromFile(ruta)
1536 MoverObjetos()
1537 End If
1538 Next ind
1539 End Sub
1540
1541 'Función al mover el mouse sobre la etapa:
1542 Private Sub EtapaMouseMove(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs)
1543 MouseMoveObjeto(sender, e, listaEtapa, "Imagenes\EtapaSel.jpg")
1544 End Sub

```

**Figura 3.7.1.2.7.2.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

El procediment és similar a l'utilitzat al MouseDown però amb la diferència de que ara s'utilitza una funció amb dos paràmetres més, la llista a la qual pertany l'objecte i la e que correspon a l'esdeveniment del ratolí. D'aquesta manera s'estalvien línies de codi ja que no s'ha d'escriure tota l'estona la funció per a cada tipus d'objecte.

Per últim, queda per explicar què succeeix quan l'usuari deixa de clicar. En aquest cas, cada tipus d'objecte actua de manera diferent ja que pel fet de tenir el booleà actiu que ajuda a posicionar els objectes amb la fi de crear els enllaços aquest procediment varia segons el tipus ja que no es el mateix posicionar una etapa que una transició. En principi es repeteixen les mateixes accions que canvien la imatge i l'atribut tag de l'objecte seleccionat i ara si l'ajuda està activada es fa un recorregut de les llistes de cadascun de la resta d'objectes i en cas que l'objecte seleccionat en aquest cas l'etapa inicial s'apropi molt de manera que arribi a travessar un dels extrems de la resta d'objectes posiciona l'etapa respecte l'objecte amb que ha tingut proximitat segons com hagi sigut aquesta aproximació. Es podria haver programat tot amb la funció `objecte1.bounds.intersectWith(objecte2.bounds)` però ja que tot i així calia mirar com era aquesta intersecció per veure si es posicionava l'objecte d'una manera o altra, es va decidir fer-ho com la tercera imatge de la figura anterior. En resum es miren les coordenades Top i Left de l'etapa inicial i si es troba dintre de la transició per sota situa l'etapa inicial abaix de la transició a la posició correcta on es crea l'enllaç; també si es per dalt situarà l'etapa inicial amunt de la transició i això amb cada un de la resta d'objectes. La funció que també es mostra a la imatge a la qual s'ha fet referència en primera instància és el procediment `AjusteLabsYMarcaEtapa0` el qual consisteix en situar de forma correcta les etiquetes de la etapa inicial un cop es produeix el reposicionament. Cal dir que el codi de la funció a la imatge està tallat i falten tots els recorreguts dels diversos objectes amb els quals pot interactuar l'etapa inicial però no cal explicar-los perquè funcionen de la mateixa forma que les transicions. Tot i així, al final del procediment es troba una trucada a la funció `ActualizarObjetos`.

Com ja s'ha mencionat abans, per cada tipus d'objecte estan programats els tres procediments anteriors. En el cas dels objectes que es troben en llistes, primerament es recorre la llista per assignar a cada objecte els 3 procediments corresponents tal i com s'observa a la figura següent:

```

1457     'Función con todos los eventos de ratón de las etapas.
1458     Private Sub EventosRatonEtapa()
1459         For ind = 0 To listaEtapa.LongCount - 1
1460             AddHandler listaEtapa(ind).MouseDown, AddressOf EtapaMouseDown
1461             AddHandler listaEtapa(ind).MouseMove, AddressOf EtapaMouseMove
1462             AddHandler listaEtapa(ind).MouseClicked, AddressOf EtapaMouseClicked
1463         Next ind
1464     End Sub

```

**Figura 3.7.1.2.7.3.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

L'altre esdeveniment de ratolí que és necessari detallar és el dobleclic sobre algunes de les etiquetes que poden aparèixer en els GRAFCETs. A continuació es mostra com és el codi amb l'exemple de les etapes:

```

1466     'Función con todos los eventos de ratón de las etiquetas de las etapas
1467     Private Sub EventosRatonLabEtapa()
1468         For ind = 0 To listaLabsEtapa.LongCount - 1
1469             AddHandler listaLabsEtapa(ind).MouseDoubleClick, AddressOf EtapaLabMouseDoubleClick
1470             AddHandler listaMarcaLabsEtapa(ind).MouseDoubleClick, AddressOf EtapaLabMarcaMouseDoubleClick
1471         Next ind
1472     End Sub
1473
1474     'Función que se activa al hacer doble clic sobre el número de la etapa
1475     Private Sub EtapaLabMouseDoubleClick(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs)
1476         GuardarDatos()
1477         If (MousePosition.X - ChildForm.Location.X - Splitter1.Width() - 10) >= sender.Left And (MousePosition.X -
1478 ChildForm.Location.X - Splitter1.Width() - 10) <= sender.Left + sender.Width() Then
1479             CambiarEtiquetaEtapaForm.ShowDialog() 'Se llama a la clase CambiarEtiquetaEtapaForm
1480         End If
1481         Try
1482             If CambiarEtiquetaEtapaForm.nuevaEtiquetaEtapa.ToString() <> "" Then
1483                 sender.Text = CambiarEtiquetaEtapaForm.nuevaEtiquetaEtapa.ToString() 'Se intenta cambiar el texto en el caso que
1484 el usuario lo haya hecho
1485                 CambiarEtiquetaEtapaForm.nuevaEtiquetaEtapa = ""
1486             End If
1487         Catch
1488             sender.Text = sender.text 'En el caso que no se haya podido, se mantiene el texto anterior
1489         End Try
1490     End Sub
1491
1492     'Función que se activa al hacer doble clic sobre la marca de la etapa
1493     Private Sub EtapaLabMarcaMouseDoubleClick(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs)
1494         GuardarDatos()
1495         If (MousePosition.X - ChildForm.Location.X - Splitter1.Width() - 10) >= sender.Left And (MousePosition.X -
1496 ChildForm.Location.X - Splitter1.Width() - 10) <= sender.Left + sender.Width() Then
1497             CambiarEtiquetaMarcaForm.ShowDialog() 'Se llama a la clase CambiarEtiquetaMarcaForm
1498         End If
1499         Try
1500             If CambiarEtiquetaMarcaForm.nuevaEtiquetaMarcaEtapa.ToString() <> "" Then
1501                 sender.Text = "M" + CambiarEtiquetaMarcaForm.nuevaEtiquetaMarcaEtapa.ToString() 'Se intenta cambiar el texto en
1502 el caso que el usuario lo haya hecho
1503                 CambiarEtiquetaMarcaForm.nuevaEtiquetaMarcaEtapa = ""
1504             End If
1505         Catch
1506             sender.Text = sender.text 'En el caso que no se haya podido, se mantiene el texto anterior
1507         End Try
1508     End Sub
1509     CambiarEtiquetaMarcaForm.TextBox1.Text = ""
1510 End Sub

```

**Figura 3.7.1.2.7.4.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

En primera instància s'observa com s'afegeix a cada una de les etiquetes l'esdeveniment del doble clic amb el recorregut de la llista de les etiquetes de les etapes. Quan es fa doble clic a sobre de l'etiqueta d'una etapa es fa la funció que comença a la línia 1475. Primer de tot es guarden les dades i es comprova que s'està fent a clic únicament a l'etapa en la que es troba el ratolí i es truca a la classe corresponent que permet fer el canvi d'etiqueta obrint una nova finestra. Això es va fer així ja que si no es limita quan hi ha diverses etapes en aquest cas, en el moment en que es cliqui sobre alguna, pel fet d'haver realitzat el recorregut de la llista, s'obrien múltiples finestres quan es truca a la classe corresponent. Una solució alternativa podria ser que cada cop que es creï una nova etapa es faci un



recorregut de la llista eliminant els handlers amb la funció RemoveHandler i tornant a crear-los. La nova finestra que apareix al programa (figura 3.3.5) permet escollir un número, acceptar i cancel·lar l'operació (s'explica el funcionament més endavant). Quan s'acaba el quadre de diàleg amb l'usuari poden haver passat dos coses, l'usuari ha cancel·lat l'operació o, en canvi, la ha acceptada. Així doncs, es presenta una estructura de tipus Try-Catch que provi de canviar l'etiqueta per la nova que ha definit l'usuari a través de la nova finestra fent clic a acceptar o que si no pot mantengui el text de l'etiqueta com era abans, o sigui que no faci res en cas que s'hagi cancel·lat. A més en cas que s'accepti es torna restaurar la variable que guarda el resultat de la finestra.

En fer doble clic sobre la marca de la etapa, el funcionament és igual que l'anterior encara que ara s'obre una nova finestra (figura 3.3.6) la qual té unes noves característiques que s'expliquen amb més profunditat més endavant. En aquest cas també s'utilitza l'estructura mencionada anteriorment i també es restaura la variable adient.

Per a les etiquetes que no són com les anteriors i no tenen limitació de caràcters, és a dir, les etiquetes associades a les transicions, convergències, divergències i etiquetes sense cap vincle; el codi que s'utilitza per canviar el seu text fent doble clic es pot observar a la següent figura:

```

1728 'Función de doble clic sobre cualquier etiqueta (transicion, convergencia, divergencia):
1729 Private Sub DobleClickLabel(ByVal sender As Object)
1730     GuardarDatos()
1731     If (MousePosition.X - ChildForm.Location.X - Splitter1.Width() - 10) >= sender.Left And (MousePosition.X -
ChildForm.Location.X - Splitter1.Width() - 10) <= sender.Left + sender.Width() Then
1732         CambiarEtiquetaForm.ShowDialog()
1733     End If
1734     Try
1735         If CambiarEtiquetaForm.nuevaEtiqueta.ToString <> "" Then
1736             sender.Text = CambiarEtiquetaForm.nuevaEtiqueta.ToString
1737             CambiarEtiquetaForm.nuevaEtiqueta = ""
1738         End If
1739     Catch
1740         sender.Text = sender.text
1741     End Try
1742     CambiarEtiquetaForm.TextBox1.Text = ""
1743 End Sub

```

**Figura 3.7.1.2.7.5.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Funciona de la mateixa forma que els procediments anteriors encara que ara es truca a una nova finestra (figura 3.3.7) i per tant les variables que intervenen són diferents. En cas que hagi una limitació de caràcters com en el cas de les etiquetes de les accions i les accions associades el codi novament ha de canviar i és el següent:

```

2024 Private Sub DobleClickLabAccion(ByVal sender)
2025     GuardarDatos()
2026     If (MousePosition.X - ChildForm.Location.X - Splitter1.Width() - 10) >= sender.Left And (MousePosition.X -
ChildForm.Location.X - Splitter1.Width() - 10) <= sender.Left + sender.Width() Then
2027         CambiarEtiquetaForm.TextBox1.MaxLength = 8 'Se limita el texto a 8 caracteres ya que las acciones se limitan a
este tamaño debido al volumen del texto
2028         CambiarEtiquetaForm.ShowDialog() 'Se llama a la clase CambiarEtiquetaForm
2029     End If
2030     Try
2031         If CambiarEtiquetaForm.nuevaEtiqueta.ToString <> "" Then
2032             sender.Text = CambiarEtiquetaForm.nuevaEtiqueta.ToString 'Se intenta cambiar el nombre de la etiqueta
2033             CambiarEtiquetaForm.nuevaEtiqueta = ""
2034         End If
2035         CambiarEtiquetaForm.TextBox1.MaxLength = 50 'Se amplia en caso de que se quiera volver a hacer doble clic sobre
otra cosa que no sea una etiqueta de la acción
2036     Catch
2037         sender.Text = sender.text 'Si no se puede cambiar el nombre se queda igual
2038         CambiarEtiquetaForm.TextBox1.MaxLength = 50 'La limitación vuelve a la original
2039     End Try
2040     CambiarEtiquetaForm.TextBox1.Text = ""
2041 End Sub
2042
2043
2044 'Función al hacer doble clic sobre la etiqueta de la acción
2045 Private Sub AccionLabMouseDoubleClick(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventHandler)
2046     DobleClickLabAccion(sender)
2047 End Sub

```

**Figura 3.7.1.2.7.6.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

De nou, l'única diferència observable és la limitació del TextBox que s'observa a la figura 3.7 perquè l'usuari no pugui crear una acció de més de 8 caràcters ja que no cabria dins de l'objecte. En cas que es volgués programar alguna manera per ampliar aquest número s'hauria de fer que la font de l'etiqueta fos més petita a partir d'aquest número o habilitar la multilínia i reestructurar l'etiqueta dins de l'acció. Tot i així degut a que es tracta de crear diagrames de GRAFCET en nivell 2 no es va considerar l'opció. Cal dir que les etiquetes dels temporitzadors no es permeten canviar a l'usuari ja que no interessa que ho facin, en cas que aquest s'equivoqui sempre pot esborrar i crear de nou.

Aquestes serien tots els procediments utilitzats en fer els esdeveniments de ratolí pel que fa als objectes. També es podria afegir i activar el clic dret per obrir una pestanya que permetés, per exemple, fer les accions d'edició.

Ara resta explicar les funcions de ratolí que actuen sobre el panell, és a dir, aquells esdeveniments que s'activen quan l'usuari interactua únicament amb el panell ja sigui amb la fi de seleccionar objectes o deseleccionar-los. Així doncs a la figura següent es mostra la part del codi corresponent:

```

3734 Private Sub EventosPanel()
3735     AddHandler Me.Panel1.MouseDown, AddressOf PanelMouseDown
3736     AddHandler Me.Panel1.MouseMove, AddressOf PanelMouseMove
3737     AddHandler Me.Panel1.MouseUp, AddressOf PanelMouseUp
3738     AddHandler Me.Panel1.MouseClick, AddressOf PanelMouseClick
3739 End Sub
3740
3741 Private Sub PanelMouseDown(sender As Object, e As MouseEventArgs)
3742     coords.X = MousePosition.X - sender.left
3743     coords.Y = MousePosition.Y - sender.top
3744     Panel1.CreateGraphics.Clear(Color.White)
3745     Deseleccionar()
3746 End Sub

```

```

3749 Private Sub PanelMouseMove(sender As Object, e As MouseEventArgs)
3750 coordsAct.X = MousePosition.X - sender.Left
3751 coordsAct.Y = MousePosition.Y - sender.Top
3752 Dim Izquierda As Int32
3753 Dim IzquierdaAct As Int32
3754 Dim Arriba As Int32
3755 Dim ArribaAct As Int32
3756 Izquierda = coords.X - ChildForm.Location.X - Splitter1.Width() - 11
3757 IzquierdaAct = coordsAct.X - ChildForm.Location.X - Splitter1.Width() - 11
3758 Arriba = coords.Y - ChildForm.Location.Y - MenuStrip1.Height() - 56
3759 ArribaAct = coordsAct.Y - ChildForm.Location.Y - MenuStrip1.Height() - 56
3760 If e.Button = Windows.Forms.MouseButtons.Left Then
3761     If coords.X <= coordsAct.X And coords.Y < coordsAct.Y Then
3762         Panel1.CreateGraphics.DrawRectangle(Pens.Aquamarine, Izquierda, Arriba, IzquierdaAct - Izquierda, ArribaAct -
Arriba)
3763         contador3 += 1
3764     ElseIf coordsAct.X <= coords.X And coordsAct.Y < coords.Y Then
3765         Panel1.CreateGraphics.DrawRectangle(Pens.Aquamarine, IzquierdaAct, ArribaAct, (IzquierdaAct - Izquierda) * -1,
(IzquierdaAct - Arriba) * -1)
3766         contador3 += 1
3767     ElseIf coordsAct.X <= coords.X And coords.Y < coordsAct.Y Then
3768         Panel1.CreateGraphics.DrawRectangle(Pens.Aquamarine, IzquierdaAct, Arriba, (IzquierdaAct - Izquierda) * -1,
ArribaAct - Arriba)
3769         contador3 += 1
3770     ElseIf coords.X <= coordsAct.X And coordsAct.Y < coords.Y Then
3771         Panel1.CreateGraphics.DrawRectangle(Pens.Aquamarine, Izquierda, ArribaAct, IzquierdaAct - Izquierda, Arriba -
ArribaAct)
3772         contador3 += 1
3773     End If
3774     If contador3 > 1 Then
3775         contador3 = 0
3776         Panel1.CreateGraphics.Clear(Color.White)
3777         Dim Izquierda As Int32
3778         Dim IzquierdaAct As Int32
3779         Dim Arriba As Int32
3780         Dim ArribaAct As Int32
3781         Izquierda = coords.X - ChildForm.Location.X - Splitter1.Width() - 11
3782         IzquierdaAct = coordsAct.X - ChildForm.Location.X - Splitter1.Width() - 11
3783         Arriba = coords.Y - ChildForm.Location.Y - MenuStrip1.Height() - 56
3784         ArribaAct = coordsAct.Y - ChildForm.Location.Y - MenuStrip1.Height() - 56
3785         If Panel1.Contains(etapa0) Then
3786             If (etapa0.Left >= Izquierda And etapa0.Left < IzquierdaAct And etapa0.Top >= Arriba And etapa0.Top < ArribaAct)
Or
(etapa0.Left <= Izquierda And etapa0.Left > IzquierdaAct And etapa0.Top <= Arriba And etapa0.Top > ArribaAct) Or
(etapa0.Left >= Izquierda And etapa0.Left < IzquierdaAct And etapa0.Top <= Arriba And etapa0.Top > ArribaAct) Or
(etapa0.Left <= Izquierda And etapa0.Left > IzquierdaAct And etapa0.Top >= Arriba And etapa0.Top < ArribaAct) Then
3787                 etapa0.Tag = True
3788                 etapa0.Image = Image.FromFile("Imagines\Etapa0Sel.jpg")
3789             End If
3790             End If
3791             For ind = 0 To listaEtapa.LongCount - 1
3792                 If (listaEtapa(ind).Left >= Izquierda And listaEtapa(ind).Left < IzquierdaAct And listaEtapa(ind).Top >= Arriba
And listaEtapa(ind).Top < ArribaAct) Or
(listaEtapa(ind).Left <= Izquierda And listaEtapa(ind).Left > IzquierdaAct And listaEtapa(ind).Left > IzquierdaAct And
listaEtapa(ind).Top <= Arriba And listaEtapa(ind).Top > ArribaAct) Or
(listaEtapa(ind).Left >= Izquierda And listaEtapa(ind).Left < IzquierdaAct And listaEtapa(ind).Left > IzquierdaAct And
listaEtapa(ind).Top >= Arriba And listaEtapa(ind).Top < ArribaAct) Or
(listaEtapa(ind).Left <= Izquierda And listaEtapa(ind).Left > IzquierdaAct And listaEtapa(ind).Left > IzquierdaAct And
listaEtapa(ind).Top >= Arriba And listaEtapa(ind).Top < ArribaAct) Then
3793                     listaEtapa(ind).Tag = True
3794                     listaEtapa(ind).Image = Image.FromFile("Imagines\EtapaSel.jpg")
3795                 End If
3796             Next ind
3797         End Sub
3798     Private Sub PanelMouseClicked(sender As Object, e As EventArgs)
3799         Deseleccionar()
3800     End Sub

```

Figura 3.7.1.2.7.7. Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

La primera captura de la figura anterior mostra la declaració dels esdeveniments de ratolí que afecten al panell on es troben els objectes i la primera funció que pertany a quan es fa clic. Inicialment es guarden les coordenades del ratolí respecte la posició del panell a la finestra principal i es truca a la funció de deseleccionar la qual s'explicarà més endavant. També es fa un clear del panell per assegurar-se que no hi ha cap imperfecció i mantenir el color blanc de fons.

La segona captura mostra l'acció que succeeix quan es mou el ratolí després d'haver clicat. En aquesta funció es van guardant les coordenades actuals segons el moviment del ratolí a les variables

corresponents i es declaren 4 de noves que defineixen els punts d'esquerra inicial, esquerra actual i el mateix a la coordenada Y. Gràcies a aquestes variables es pot construir un rectangle buit mentre es mou el ratolí. A la línia 3760 es limita a actuar mentre es mantingui el clic o sigui mentre s'està arrossegant el ratolí al lloc que es vol. En funció a on es faci el clic inicial i cap la direcció que es mogui el ratolí mentre s'arrossega apareixen 4 possibilitats que es cliqui i es mogui cap amunt a la dreta, cap a amunt a l'esquerra, cap avall a la dreta i cap avall a l'esquerra. En funció a com hagi sigut el moviment s'haurà de crear el rectangle amb la funció `CreateGraphics.DrawRectangle()` la qual dibuixa un rectangle al panell a partir d'un tipus de llapis i 4 posicions, les 2 posicions del rectangle corresponents al vèrtex superior esquerre, la llargada i l'amplada. Aquestes posicions s'obtenen mitjançant les variables anteriors de forma directa o amb petites manipulacions matemàtiques. També, s'incrementa en 1 el comptador de rectangles. Això es fa per limitar a que només es vegi un rectangle al panell, ja que si no es fes tacaria tota la pantalla de rectangles un rere l'altre mentre es fa l'acció. Així doncs cada cop que es supera el comptador de rectangles (línia 3774) es reinicia el comptador, es neteja el panell i es torna a crear un rectangle final amb la posició en que es troba en aquell moment ja que sinó mentre es moguéss el ratolí no es veuria cap rectangle, cosa que tampoc interessa.

La següent funció la qual s'activa quan l'usuari deixa de clicar consta de les mateixes variables definides anteriorment però declarades de nou ja que no es tracten de variables globals. En aquest instant si la part extrema superior esquerra d'un objecte es troba dins del rectangle dibuixat anteriorment l'objecte estarà seleccionat. Per fer-ho, es recorren totes les llistes d'objectes i objectes per separat i es mira si es troben dintre del rectangle. En el cas que sí l'atribut `tag` de l'objecte es torna `true` i la seva imatge canvia a la imatge d'objecte seleccionat. Cal dir que la tercera imatge està tallada i faltaria tota la part de codi de cada una de les diferents llistes. Aquest algoritme es podria millorar considerant no només la part superior esquerra dels objectes sinó el rectangle de l'objecte en sí, però caldria elaborar una mica més les limitacions.

L'última funció pertany a l'acció quan l'usuari fa un clic sobre el panell sense arrossegar. Únicament caldrà deixar de seleccionar els objectes que estiguin seleccionats.

### **3.7.1.2.8 Procediments MoverObjetos, GuardarCoords i ActualizarObjetos**

En aquest subapartat es mostra com s'han realitzat les funcions de moure objectes, guardar coordenades i actualitzar objectes tant necessàries en el moviment de qualsevol objecte.

Així doncs, a continuació es mostra una figura amb imatges retallades del codi dels procediments anteriors:

```

3953 'Funció empleada per moure tots els objectes en relació a la posició del ratolí i delimitada per els marges del panell:
3954 Private Sub MoverObjetos()
3955     If Panel1.Contains(etapa0) Then
3956         If etapa0.Tag = True Then
3957             etapa0.Top = MousePosition.Y - coordsEtapa0.Y
3958             etapa0.Left = MousePosition.X - coordsEtapa0.X
3959             If etapa0.Top <= 0 Then
3960                 etapa0.Top = 0
3961             End If
3962             If etapa0.Left <= 35 Then
3963                 etapa0.Left = 35
3964             End If
3965             If (etapa0.Top + etapa0.Height) >= Panel1.Height Then
3966                 etapa0.Top = Panel1.Height - etapa0.Height
3967             End If
3968             If (etapa0.Left + etapa0.Width) >= Panel1.Width Then
3969                 etapa0.Left = Panel1.Width - etapa0.Width
3970             End If
3971             AjusteLabsYMarcaEtapa0()
3972         End If
3973     End If
3974
3975     For ind = 0 To listaEtapa.LongCount - 1
3976         If listaEtapa(ind).Tag = True Then
3977             listaEtapa(ind).Top = MousePosition.Y - listaCoordsEtapa(ind).Y
3978             listaEtapa(ind).Left = MousePosition.X - listaCoordsEtapa(ind).X
3979             If listaEtapa(ind).Top <= 0 Then
3980                 listaEtapa(ind).Top = 0
3981             End If
3982             If listaEtapa(ind).Left <= 35 Then
3983                 listaEtapa(ind).Left = 35
3984             End If
3985             If (listaEtapa(ind).Top + listaEtapa(ind).Height) >= Panel1.Height Then
3986                 listaEtapa(ind).Top = Panel1.Height - listaEtapa(ind).Height
3987             End If
3988             If (listaEtapa(ind).Left + listaEtapa(ind).Width) >= Panel1.Width Then
3989                 listaEtapa(ind).Left = Panel1.Width - listaEtapa(ind).Width
3990             End If
3991             AjusteLabsYMarcaEtapa(ind)
3992         End If
3993     Next ind
3994
4386 'Funció que guarda la posició de els objectes, és a dir, les coordenades per poder fer bé el moviment de cada un
de els objectes:
4387 Private Sub GuardarCoords()
4388     If Panel1.Controls.Contains(etapa0) Then
4389         coordsEtapa0 = MousePosition - etapa0.Location
4390     End If
4391     For ind = 0 To listaEtapa.LongCount - 1
4392         If listaEtapa(ind).Tag = True Then
4393             listaCoordsEtapa(ind) = MousePosition - listaCoordsEtapa(ind)
4394         End If
4395     Next ind
4396     For ind = 0 To TransicionesForm.listaTrans.LongCount - 1
4397         If TransicionesForm.listaTrans(ind).Tag = True Then
4398             listaCoordsTrans(ind) = MousePosition - listaCoordsTrans(ind)
4399         End If
4400     Next ind
4401
4498 'Actualiza la coordenada de els objectes per evitar errors en el proximo moviment.
4499 Private Sub ActualizarObjetos()
4500     If Panel1.Contains(etapa0) Then
4501         coordsEtapa0 = etapa0.Location
4502     End If
4503     For ind = 0 To listaEtapa.LongCount - 1
4504         listaCoordsEtapa(ind) = listaEtapa(ind).Location
4505     Next ind
4506     For ind = 0 To TransicionesForm.listaTrans.LongCount - 1
4507         listaCoordsTrans(ind) = TransicionesForm.listaTrans(ind).Location
4508     Next ind

```

Figura 3.7.1.2.8.1. Captures d’imatge part del codi annex A1. (Font: Pablo Mañas)

La primera funció que es mostra a la figura anterior consisteix en realitzar el moviment dels objectes mitjançant el ratolí ja que sempre en els esdeveniments del tipus MouseMove s’acaba trucant a aquest mètode com ja s’ha mostrat en el subapartat anterior. En primera instància es condiona si existeix una etapa inicial creada al panell i si aquesta està seleccionada utilitzant l’atribut tag. En cas afirmatiu es procedeix a moure l’objecte segons la posició relativa al cursor. Per fer-ho, s’actualitza l’atribut Top de l’objecte referent a la posició Y de l’objecte respecte el seu contenidor que és el panell mitjançant una diferència entre la posició Y del ratolí respecte la finestra principal i la coordenada emmagatzemada del mateix eix. Després s’actualitza l’atribut Left amb les coordenades corresponents a l’eix X. D’aquesta manera s’aconsegueix tenir l’objecte sempre on es troba el cursor durant el moviment. Ja que no es pot moure objectes fora del panell, perquè visualment desapareixerien, s’han de delimitar les posicions relatives dels objectes respecte el seu contenidor d’una manera similar a la que s’ha explicat anteriorment amb els esdeveniments de teclat. Així doncs quatre delimitacions, una per cada costat del panell, per exemple entre les línies 3979 i 3990 de la imatge anterior s’observen, retenen l’objecte dins del seu contenidor que és el panell. Mentre es fa el moviment també s’han de

more les etiquetes que van lligades amb ells i per tant, també es truca a les funcions d'ajust ja mencionades anteriorment com el procediment `AjusteLabsYMarcaEtapa0` a l'apartat anterior.

Per als objectes que es troben en llistes caldrà recórrer totes les llistes que poden existir però amb la mateixa idea que la explicada al paràgraf anterior.

La segona funció de la imatge consisteix en guardar les coordenades dels objectes en les variables corresponents utilitzades durant el moviment dels mateixos. Així com en el procediment anterior, es recorren totes les llistes d'objectes per guardar en cada variable la coordenada necessària. Aquesta coordenada és guardada com un punt i s'obté de la diferència entre la posició del ratolí respecte la finestra principal i la localització de l'objecte respecte el panell. Es fa d'aquesta manera ja que si l'usuari mou la finestra filla on es troba el panell no es perdin les referències de la situació de l'objecte i no es produeixin errors al moure.

Per últim, existeix la funció d'actualitzar objectes la qual guarda les coordenades dels objectes sense tenir en compte la posició del ratolí. Això es abans que es produeixi el moviment de l'objecte i al final per tenir sempre la referència d'on es troba l'objecte. En cas que s'obviés aquest procediment un cop s'acabara el moviment probablement la funció de moure objectes fallaria en cas que es mogué el panell de lloc per la mateixa raó que la segona funció, és a dir, perquè utilitzaria una variable que es va actualitzar al moviment anterior.

Amb aquestes tres funcions es poden moure tots els objectes existents al panell en grup i de forma individual sempre i quan es respectin els límits d'aquest, ja que si es desplacen en grup cap a un dels extrems s'apilarien tots als extrems i es perdria la forma del diagrama creat.

#### **3.7.1.2.9 Procediment Deseleccionar**

En referència als esdeveniments de ratolí aplicats al panell, quan l'usuari fa un clic únicament es crida a la funció `Deseleccionar`, la qual consisteix en recórrer totes les llistes i objectes del panell per canviar totes les imatges i els atributs `tag` de seleccionats a no seleccionats. Per fer-ho es creen matrius addicionals que contenen llistes on els objectes comparteixen la mateixa imatge quan es seleccionen. La següent imatge mostra una part del codi d'aquesta funció on es veu el que prèviament s'ha explicat:

```

4568 'Función que deselecciona todos los objetos que hay en el panel, cuando el usuario clica en cualquier parte que no sea
objeto.
4569 Private Sub Deseleccionar()
4570 Dim matrizListaTrans = {TransicionesForm.listaTrans, TransicionesForm.listaTransAlto, TransicionesForm.listaTransBajo,
TimerForm.listaTransTimer, TimerForm.listaTransTimerAsociado, TimerForm.listaTransTimerWegado,
TimerForm.listaTransTimerAsociadoWegado}
4571 Dim matrizListaConv = {ConvergenciaForm.listaTransConvergencia, ConvergenciaForm.listaTransAltoConvergencia,
ConvergenciaForm.listaTransBajoConvergencia}
4572 Dim matrizListaDiv = {DivergenciaForm.listaTransDivergencia, DivergenciaForm.listaTransAltoDivergencia,
DivergenciaForm.listaTransBajoDivergencia}
4573 Panel1.CreateGraphics().Clear(Color.White)
4574 If Panel1.Contains(etapa0) Then
4575     etapa0.Image = Image.FromFile("Imagenes\Etapa0.jpg")
4576     Label0.BackColor = Color.White
4577     etapa0.Tag = False
4578 End If
4579 For ind = 0 To listaEtapa.LongCount - 1
4580     listaEtapa(ind).Image = Image.FromFile("Imagenes\Etapa.png")
4581     listaEtapa(ind).Tag = False
4582 Next ind
4583 For ind = 0 To matrizListaTrans.LongCount - 1
4584     For ind1 = 0 To matrizListaTrans(ind).LongCount - 1
4585         matrizListaTrans(ind)(ind1).Image = Image.FromFile("Imagenes\Transicion.png")
4586         matrizListaTrans(ind)(ind1).Tag = False
4587     Next ind1
4588 Next ind

```

Figura 3.7.1.2.9.1. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Així com s'observa a la imatge anterior el recorregut per la matriu es fa mitjançant dos bucles amb la fi d'arribar a tots els objectes. També es podria haver programat sense l'ús de matrius i utilitzant les llistes ja definides, però com l'ús de matrius estalviava lleugerament línies de codi es va decidir fer-ho així. La imatge queda retallada i continuaria fent els bucles necessaris per recórrer les matrius i les llistes necessàries.

### 3.7.1.2.10 Procediment EscribirArchivoGuardado

Aquest procediment és el que permet sobreescrivre sobre una variable del tipus string totes les dades del document generat. És molt important ja que es truca gairebé a cada acció com ja s'ha mencionat al subapartat 3.7.1.2.1. A continuació s'observa una petita mostra d'aquesta funció:

```

4878 'Función que escribe todos los datos guardados en una variable de tipo string
4879 Public Sub EscribirArchivoGuardado(ByRef datos As String)
4880     datos = NuevoDocumentoForm.GuardarTamañoPapel + "|"
4881     Try
4882         If Panel1.Contains(etapa0) Then
4883             datos &= "Etapa0|"
4884             datos &= etapa0.Left.ToString + "|"
4885             datos &= etapa0.Top.ToString + "|"
4886         Else
4887             datos &= "NoEtapa0|"
4888             datos &= "NoEtapa0|"
4889             datos &= "NoEtapa0|"
4890         End If
4891     Catch
4892         datos &= "NoEtapa0|"
4893         datos &= "NoEtapa0|"
4894         datos &= "NoEtapa0|"
4895     End Try
4896     For ind = 0 To listaEtapa.LongCount - 1
4897         datos &= "Etapa" + "|"
4898         datos &= listaEtapa(ind).Left.ToString + "|"
4899         datos &= listaEtapa(ind).Top.ToString + "|"
4900         datos &= listaLabsEtapa(ind).Text.ToString + "|"
4901         datos &= listaMarcaLabsEtapa(ind).Text.ToString + "|"
4902         datos &= ind.ToString + "|"
4903     Next ind
4904     For ind = 0 To TransicionesForm.listaTrans.LongCount - 1
4905         datos &= "Transicion" + "|"
4906         datos &= TransicionesForm.listaTrans(ind).Left.ToString + "|"
4907         datos &= TransicionesForm.listaTrans(ind).Top.ToString + "|"
4908         datos &= listaLabsTrans(ind).Text.ToString + "|"
4909         datos &= ind.ToString + "|"
4910     Next ind

```

Figura 3.7.1.2.10.1. Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

En primer lloc cal dir que es tracta d'un procediment de tipus públic ja que s'utilitza en altres classes a part de la del menú principal i que modifica el paràmetre quan es fa la funció perquè es passen les dades per referència. A la primera línia dintre de la funció es sobreescriveix a sobre del paràmetre una

variable de la classe NuevoDocumentoForm que conté el tamany de full del document creat. Després es troba una instrucció del tipus Try-Catch que intenta afegir al paràmetre una cadena "Etapa0|" i seguit la localització de l'objecte de l'etapa inicial separat per barres ("|"), en el cas que hi hagi un error o no existeixi l'etapa s'afegirà a la cadena una sèrie de tres cadenes separades també per barres que identifiquen que no existeix l'etapa inicial. Continuant amb el codi, es recorren totes les llistes d'objectes i es va afegint sobre la cadena ja existent el nom identificatiu de l'objecte com pot ser "Etapa" la localització convertida en cadena, el text de l'etiqueta associada, el text de l'etiqueta de la marca associada i l'índex de la llista. Cada objecte s'identificarà per el seu nom, índex i localització en el panell. En alguns casos es poden afegir altres dades com l'amplada o l'alçada de les línies horitzontals o verticals. Cadascuna de les característiques de cada objecte i les seves associacions es queden emmagatzemades en el paràmetre del tipus cadena separades entre sí per barres. Finalment s'habrà modificat el paràmetre de tal manera que contengui totes les dades del panell guardades en una sola cadena. Cada característica es separa per barres perquè serà possible separar les dades després en les funcions necessàries amb la fi d'obtenir aquesta informació. Al final d'aquest mètode es truca al procediment d'actualitzar la taula d'enllaços que es detallarà més endavant.

#### **3.7.1.2.11 Procediment EliminarSeleccion**

En aquest subapartat es centra en el procediment EliminarSeleccion que ja s'ha mencionat als esdeveniments de teclat, més concretament a la part on s'explica la tecla de suprimir. En sí el mètode consisteix en eliminar tot aquell objecte que estigui seleccionat i, per tant, és necessari recorre de nou tots els objectes del panell i veure quins estan seleccionats. A continuació es mostra una imatge retallada del codi al que fa referència aquest capítol:



```

5214 'Función borra elementos seleccionados:
5215 Private Sub EliminarSeleccion()
5216     If Me.Panel1.Contains(etapa0) Then
5217         If etapa0.Tag = True Then
5218             Me.Panel1.Controls.Remove(etapa0)
5219             Me.Panel1.Controls.Remove(Label0)
5220             Me.Panel1.Controls.Remove(LabelMarca0)
5221             etapa0 = Nothing
5222             Label0 = Nothing
5223             LabelMarca0 = Nothing
5224             BotonEtapa0.Enabled() = True
5225             Try
5226                 EnlacesForm.DataGridView1.Rows.RemoveAt(0)
5227             Catch
5228             End Try
5229         End If
5230     End If
5231
5232     For ind = listaEtapa.LongCount - 1 To 0 Step -1 'Recorre la lista de las etapas de mayor a menor
5233         If listaEtapa(ind).Tag = True Then 'Si hay alguna etapa seleccionada
5234             Me.Panel1.Controls.Remove(listaEtapa(ind)) 'Borra la etapa del panel
5235             Me.Panel1.Controls.Remove(listaLabsEtapa(ind)) 'Borra su etiqueta
5236             Me.Panel1.Controls.Remove(listaMarcaLabsEtapa(ind)) 'Borra su número
5237             listaEtapa.RemoveAt(ind) 'Elimina de la lista de etapas el elemento
5238             listaLabsEtapa.RemoveAt(ind) 'Elimina de la lista de etiquetas el elemento
5239             listaMarcaLabsEtapa.RemoveAt(ind) 'Elimina de la lista de marcas el elemento
5240             If Me.Panel1.Contains(etapa0) Then
5241                 EnlacesForm.DataGridView1.Rows.RemoveAt(ind + 1) 'Borra la fila de enlaces si existe etapa0
5242             Else
5243                 EnlacesForm.DataGridView1.Rows.RemoveAt(ind) 'Borra la fila de enlaces si no existe etapa0
5244             End If
5245         End If
5246     Next ind
5247
5248     'Se aplica lo anterior para los demás objetos:
5249     For ind = TransicionesForm.listaTrans.LongCount - 1 To 0 Step -1
5250         If TransicionesForm.listaTrans(ind).Tag = True Then
5251             Me.Panel1.Controls.Remove(TransicionesForm.listaTrans(ind))
5252             Me.Panel1.Controls.Remove(listaLabsTrans(ind))
5253             TransicionesForm.listaTrans.RemoveAt(ind)
5254             listaLabsTrans.RemoveAt(ind)
5255         End If
5256     Next

```

**Figura 3.7.1.2.11.1.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Desglossant el codi anterior s'observa com es manipulen les variables corresponents. Per exemple, si es mira la línia 5218 en endavant, on s'arriba un cop l'etapa inicial ha estat seleccionada i s'ha provat d'esborrar, es pot apreciar que primerament s'elimina l'element del panell i el control d'aquest mitjançant el mètode del panell de Controls.Remove(objecte) en aquest cas l'objecte és la PictureBox de l'etapa inicial. A les dues línies següents es fa el mateix amb l'etiqueta de l'etapa i l'etiqueta de la marca. Encara que s'hagin eliminat aquests objectes del panell és necessari restablir les variables perquè es puguin tornar a utilitzar. Així doncs, les variables s'iguen a Nothing o sigui, a res. En aquest cas, també s'habilita el botó que permet crear l'etapa inicial. Per últim, a la línia 5226 es fa una estructura try-catch que prova d'eliminar la fila 0 de la taula d'enllaços ja que és on es troba l'etapa inicial i les seves dades amb els enllaços que té.

En el cas de la resta d'objectes seleccionats del panell cal fer un recorregut des de l'índex major al menor de les llistes corresponents. A cada objecte de la llista s'observa si està seleccionat i en el cas afirmatiu s'esborra primer del panell de forma similar a l'exemple anterior i després s'elimina de la llista aquell element. Si el recorregut es fes de menor a major el programa donaria problemes perquè no es podrien eliminar els elements i caldria utilitzar variables auxiliars per fer-ho. En el cas de les etapes, la part d'esborrar les files de la taula d'enllaços es fa en funció a si existeix o no l'etapa inicial.

### 3.7.1.2.12 Procediments relacionats amb la pestanya Archivo

Dintre d'aquest apartat es pretén mostrar a nivell de codi cadascuna de les funcions que s'efectuen quan l'usuari fa clic als botons de la pestanya Archivo que es troba dintre del ToolStripMenuItem, o sigui la barra principal sota el nom del programa. Seguint un ordre de dalt a baix en primer lloc es situa la funció Nuevo. Quan l'usuari fa clic l'esdeveniment que es genera és el següent:

```

2655     'Función al hacer clic sobre Archivo -> Nuevo
2656     Private Sub NuevoToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles NuevoToolStripMenuItem.Click
2657         Dim resultado As DialogResult "Crea una respuesta de diálogo
2658         If archivo = False Then 'Si no hay archivo creado
2659             NuevoDocumentoForm.ShowDialog() 'Se llama a la clase NuevoDocumento para crear el nuevo documento
2660             If ChildForm.Controls.Contains(Panell1) Then 'Si existe un panel dentro de la ventana hija que se ha creado
2661                 EventosPanel() 'Se habilitan los eventos del panel
2662             End If
2663         Else 'Si existe ya un archivo creado, se pregunta si se quiere crear otro nuevo, no pueden existir dos documentos al
2664             mismo tiempo
2665             resultado = MessageBox.Show("Quieres crear un archivo nuevo?" + Chr(13) + "(si lo haces perderás el progreso
2666             anterior)", "Nuevo archivo", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning)
2667             If resultado = DialogResult.OK Then 'Si el usuario acepta:
2668                 BorrarTodo() 'Se borra todo lo que hay en el panel y se limpian las variables modificadas.
2669                 Me.Controls.Remove(Panell1) 'Se elimina el panel
2670                 Me.Controls.Remove(ChildForm) 'Se elimina la ventana hija
2671                 Me.ChildForm.Close() 'Se cierra la misma
2672                 NuevoDocumentoForm.ShowDialog() 'Se llama a la clase NuevoDocumento para crear el nuevo documento
2673                 If ChildForm.Controls.Contains(Panell1) Then 'Si existe un panel dentro de la ventana hija que se ha creado
2674                     EventosPanel() 'Se habilitan los eventos del panel
2675                 End If
2676             End If
2677         End Sub

```

**Figura 3.7.1.2.12.1.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Observant la capçalera es veu una estructura diferent però que és fàcil d'entendre. Quan es fa clic al botó, NuevoToolStripMenuItem.Click es tornaria true ja que s'ha fet clic i per tant, es truca a la funció NuevoToolStripMenuItem\_Click que és la que es mostra. En aquest cas els paràmetres no s'utilitzen dintre de la funció però el propi Visual Studio genera la funció d'aquesta manera. Centrant-nos en l'inici de la funció es defineix una variable anomenada resultado que guardarà el resultat d'un quadre de diàleg que es generarà en cas que ja existeixi un arxiu creat. Per tant, si no existeix un arxiu (línia 2658) es truca a la classe NuevoDocumentoForm que generarà la finestra mostrada a la figura 3.1.3. Quan s'hagi creat el panell habilitarà els events de ratolí del panell ja explicats dintre del subapartat 3.7.1.2.7.

En cas que ja n'hi hagi un arxiu creat, o sigui, un panell i una finestra filla, com no s'ha permès que es pugui crear un altre perquè complicava algunes de les funcions i no es considerava molt útil tenir diversos GRAFCETS oberts al mateix temps, es crea un quadre de diàleg (figura 3.1.6) amb l'avertència de que es perdrà l'arxiu creat. Si s'accepta el quadre de diàleg, s'esborrarà tot el contingut de l'arxiu amb la funció BorrarTodo, s'eliminarà el panell, juntament amb la finestra filla que es tancarà. Després es produeix el mateix que quan no hi ha un arxiu creat. En cas que es produeixin errors en els diàlegs, l'usuari podrà crear dues finestres buides que caldrà que tanqui per tornar a fer el diagrama novament.

El pròxim procediment és el de la funció que es deriva quan es fa clic a l'ítem Abrir. El codi que controla aquesta situació és el següent:

```

4699 'Función abrir archivo:
4700 Private Sub AbrirToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles AbrirToolStripMenuItem.Click
4701 Dim resultado As DialogResult 'Se define una variable de respuesta de diálogo
4702 If archivo = False Then 'Si no existe ya un documento:
4703 AbrirArchivo() 'Se abre el archivo sin ningún aviso
4704 Else
4705 resultado = MessageBox.Show("Quieres crear un archivo nuevo?" + Chr(13) + "(si lo haces perderás el progreso
anterior)", "Nuevo archivo", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning) 'Sino se advierte ya que se cerrara el
anterior
4706 If resultado = DialogResult.OK Then 'En el caso que el usuario acepte
BorrarTodo() 'Se borra todo lo del anterior grafcet
4708 Me.Controls.Remove(ChildForm) 'Se elimina la ventana
4709 Me.ChildForm.Close() 'Se cierra la ventana para liberar
AbrirArchivo() 'Se abre el archivo
4711 End If
4712 End If
4713 End Sub

```

Figura 3.7.1.2.12.2. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Així com en la funció anterior la capçalera és molt similar i l'interior del procediment també a excepció de que en comptes de trucar a la classe NuevoDocumentoForm i habilitar els esdeveniments de ratolí del panell es deriva al procediment AbrirArchivo el codi del qual es troba a continuació:

```

4716 Private Sub AbrirArchivo()
4717 'Se crea un cuadro de dialogo para seleccionar el archivo txt a abrir
4718 Dim OpenFileDialog As New OpenFileDialog With {.InitialDirectory =
My.Computer.FileSystem.SpecialDirectories.MyDocuments, .Filter = "Archivos de texto (*.txt)*.txt[Todos los archivos
(*.*)*.*"}
4719 If (OpenFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then
4720 Dim FileName As String = OpenFileDialog.FileName 'Se guarda el nombre del archivo abierto
4721 Dim LeerArchivo As IO.StreamReader 'Se define el archivo a leer
4722 Dim LecturaDatos As String 'Se define una variable que almacena los datos leídos del fichero
4723 Dim SepararDatos() As String 'Se define una lista que separara los datos leídos
4724 LeerArchivo = My.Computer.FileSystem.OpenTextFileReader(FileName) 'Se abre el archivo en la variable
4725 LecturaDatos = LeerArchivo.ReadToEnd() 'Se lee todo el archivo y se guarda en la variable
4726 SepararDatos = LecturaDatos.Split("|") 'Se separan todos los datos guardados
4727 Try
4728 'En función del primer dato guardado el tamaño del nuevo panel será diferente:
4729 If SepararDatos(0) = "595842" Then
4730 NuevoDocumentoForm.CrearPanel(611, 881, 595, 842) 'Se crea el panel
4731 NuevoDocumentoForm.Activar() 'Se activan los botones necesarios
4732 EventosPanel() 'Se activan los eventos de ratón del panel
4733 ElseIf SepararDatos(0) = "842595" Then
4734 NuevoDocumentoForm.CrearPanel(858, 634, 842, 595)
4735 NuevoDocumentoForm.Activar()
4736 EventosPanel()
4737 ElseIf SepararDatos(0) = "1191842" Then
4738 NuevoDocumentoForm.CrearPanel(1207, 881, 1191, 842)
4739 NuevoDocumentoForm.Activar()
4740 EventosPanel()
4741 ElseIf SepararDatos(0) = "8421191" Then
4742 NuevoDocumentoForm.CrearPanel(858, 1230, 842, 1191)
4743 NuevoDocumentoForm.Activar()
4744 EventosPanel()
4745 End If
4746 RecuperarPosicionAnterior(SepararDatos) 'Se recuperan todos los objetos de la lista con los datos guardados
4747 Deseleccionar()
4748 ChildForm.Text = FileName.Replace(".txt", "") 'Se reemplaza el nombre de la ventana hija por el del fichero
abierto
4749 LeerArchivo.Close() 'Se cierra el archivo de lectura
4750 Catch
4751 MessageBox.Show("No se ha podido leer el archivo") 'Si hay algun error se avisa
4752 BorrarTodo() 'Se borra todo lo generado
4753 ChildForm.Close() 'Se cierra la ventana hija
4754 LeerArchivo.Close() 'Se cierra el archivo de lectura
4755 End Try
4756 End If
4757 End Sub

```

Figura 3.7.1.2.12.3. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Dintre d'aquest procediment es crea un quadre de diàleg on es poden seleccionar els arxius .txt on es guarden els grafcets generats pel programa (figura 3.1.8). Un cop s'accepta el diàleg, o sigui, es pretén obrir un fitxer de tipus .txt, primer de tot es guarda el nom de l'arxiu en una variable, s'emmagatzema tot el contingut de l'arxiu en una variable que contindrà totes les dades guardades del mateix estil que es guardaven les dades amb el procediment explicat al subapartat 3.7.1.2.10. Així doncs, cal separar totes aquestes dades per obtenir l'informació necessària per obrir el diagrama utilitzant el mètode Split("|") el qual emmagatzema en una llista del tipus string cadascun dels elements emmagatzemats prèviament a l'arxiu. Un cop es separen les dades ja es pot extreure l'informació i, per tant, en primer lloc es prova de mirar la dimensió del paper del GRAFCET guardat. Gràcies a la funció Try-Catch en cas

que es produeixi un error, s'avisarà mitjançant un missatge i es borrarà tot el que s'hagi pogut crear de manera accidental. En el cas que tot vagi bé es crearà el panell trucant a la funció de la classe necessària, en aquest cas NuevoDocumentoForm.CrearPanel, s'activaran els botons necessaris els events del panell. Després amb la funció RecuperarPosicionAnterior es crearan els objectes amb les dades obtingudes al llegir. Per últim es farà una deselecció dels objectes es canviarà el nom de la finestra filla per el de l'arxiu obert i es tancarà la comunicació amb el fitxer .txt.

En tercer lloc de la pestanya es troba la funció de Cerrar. El codi que es deriva d'aquest procediment és el següent:

```

4759 'Función al hacer clic en cerrar
4760 Private Sub CerrarToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles CerrarToolStripMenuItem.Click
4761     For Each ChildForm As Form In Me.MdiChildren
4762         ChildForm.Close() 'Se cierran todas las ventanas hijas.
4763     Next
4764 End Sub

```

Figura 3.7.1.2.12.4. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

En resum el codi recorre totes les finestres filles existents i les tanca a través de la funció Close. Quan es fa el tancament, de les finestres el programa adverteix, però ho fa a través d'altra classe que s'explica en altre subapartat.

A la quarta pestanya començant per dalt de la pestanya en qüestió es troba la funció Guardar. Com aquesta i la quinta (Guardar Como...) estan molt relacionades s'ha decidit comentar-les en conjunt.

```

4651 'Función Guardar normal:
4652 Private Sub Guardar()
4653     Deseleccionar() 'Todos los tags en false y los objetos deseleccionados
4654     Dim archivo As IO.StreamWriter 'Define un archivo donde se escribirá el documento txt
4655     Dim Ancho As Integer = Panel1.Width() 'Define ancho del panel
4656     Dim Alto As Integer = Panel1.Height() 'Define altura del panel
4657     Dim Mapa As New Bitmap(Ancho, Alto) 'Define un Mapa con el tamaño del panel
4658     RedistribuirObjetos() 'Redistribuye objetos para guardar el mapa tal y como se ve en el panel
4659     Panel1.DrawToBitmap(Mapa, Panel1.Bounds) 'Crea la imagen del panel
4660     ReestructurarObjetos() 'Reestructura los objetos para que vuelvan a estar como antes
4661     Mapa.Save(RutaArchivoGuardado, Imaging.ImageFormat.Png) 'Guarda la imagen en la ruta definida
4662     If My.Computer.FileSystem.FileExists(RutaArchivoGuardadoTxt) Then 'Si existe el texto
4663         My.Computer.FileSystem.DeleteFile(RutaArchivoGuardadoTxt) 'Elimina el texto
4664     End If
4665     archivo = My.Computer.FileSystem.OpenTextFileWriter(RutaArchivoGuardadoTxt, True) 'Abre un archivo de texto en la ruta
de guardado
4666     EscribirArchivoGuardado(DatosGuardados) 'Escribe los datos actuales del grafcet en la variable
4667     archivo.Write(DatosGuardados) 'Escribe los datos del grafcet en el archivo texto
4668     archivo.Close() 'Cierra el archivo de texto
4669 End Sub
4670
4671 'Función Guardar diagrama:
4672 Private Sub GuardarToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles GuardarToolStripMenuItem.Click
4673     If PrimerGuardado = True Or ChildForm.Text = "Sin Título" Then 'Si es la primera vez que se guarda:
4674         GuardarComo() 'Abre el diálogo de guardar como
4675     Else
4676         Guardar() 'Se llama a la función de guardar
4677     End If
4678 End Sub
4679
4680 'Función Guardar Como... del grafcet:
4681 Private Sub GuardarComo()
4682     'Se genera un cuadro de diálogo de guardado
4683     Dim SaveFileDialog As New SaveFileDialog With {.InitialDirectory =
My.Computer.FileSystem.SpecialDirectories.MyDocuments, .Filter = "Archivos de grafcet (*.png)|*.png|Todos los archivos
(*.*)|*.*"}
4684     If (SaveFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then 'Si se acepta el diálogo:
4685         Dim FileName As String = SaveFileDialog.FileName 'Se define el nombre del archivo que ha dado el usuario
4686         Dim FileNameTxt = FileName.Replace(".png", ".txt") 'Se define el nombre del archivo en texto reemplazando la
string anterior por el cambio de formato
4687         ChildForm.Text = FileName.Replace(".png", "") 'Se define la ventana hija donde está el panel con el grafcet con
el nombre guardado
4688         RutaArchivoGuardado = FileName 'Se define la variable con el nombre del archivo png
4689         RutaArchivoGuardadoTxt = FileNameTxt 'Se define la variable con el nombre del archivo txt
4690         PrimerGuardado = False 'Se vuelve falso el booleano de primer guardado
4691         Guardar()
4692     End If
4693 End Sub
4694
4695 'Función al clicar sobre guardar como:
4696 Private Sub GuardarComoToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
GuardarComoToolStripMenuItem.Click
4697     GuardarComo()
4698 End Sub

```

Figura 3.7.1.2.12.5. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

En el moment que l'usuari clica sobre l'opció Guardar s'activa la funció observable a la línia 4672 i en cas que fos el primer guardat o la finestra filla s'anomeni "Sin Título" perquè no s'ha guardat aquell diagrama, s'obre el diàleg de guardar como (figura 3.1.7). L'usuari escull on guarda l'arxiu i un cop ho fa, es guarda el nom de l'arxiu que ha donat l'usuari i es guarda en diferents variables un amb .txt i l'altre .png; es modifica el nom de la finestra filla per la ruta de l'arxiu i s'actualitza el booleà que controlava el primer guardat. Després es procedeix a guardar el diagrama. Primer de tot, si n'hi ha algun objecte seleccionat es deixa de ser seleccionat i es crea un mapa amb les dimensions del panell on es dibuixa tot el que es troba a dins i es guarda a la ruta seleccionada anteriorment en format png, també es genera un fitxer de tipus .txt amb les últimes dades guardades de l'estat del grafcet. Simplement es guarda la variable DatosGuardados que s'acaba d'actualitzar trucant a la funció corresponent al fitxer de tipus .txt que té el mateix nom que el de .png i que en cas que ja existeixi s'elimina i es crea un de nou. Les funcions RedistribuirObjetos (línia 4658) i ReestructurarObjetos (línia 4660) que es defineixen a les línies 4767 i 4832 del programa consisteixen en portar totes les etiquetes al fons del panell per fer el mapa real i després tornar-les on es troben. No es creu necessari parlar en específic d'aquestes dues funcions ja que l'únic que fan és un recorregut de tots els objectes de la mateixa manera que es fa al subapartat 3.7.1.2.10. amb la funció de deseleccionar, però en aquest cas s'utilitzen les funcions de label.SendToBack i label.BringToFront.

En el cas que es cliqui sobre el botó de guardar i ja s'hagi guardat prèviament, sense cap avís el programa guardarà a la ruta on es va decidir per primer cop amb el quadre de diàleg. Obviament si es fa clic al botó de Guardar Como s'obrirà de nou un quadre de diàleg que permet escollir novament la ruta on es desitgi guardar els arxius.

Continuant amb la pestanya Archivo al fer clic en Exportar es troben dos funcions més que cal explicar. La primera de totes és l'exportació en format pdf i el codi relacionat amb ella és el que es mostra a continuació:

```

6115 'Función para exportar en PDF
6116 Private Sub ExportarEnFormatoPDFToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ExportarEnFormatoPDFToolStripMenuItem.Click
6117     Deseleccionar() 'Deselecciona objetos
6118     ReestructurarObjetos() 'Reestructura objetos por si acaso
6119     If Panel1.Width() = 595 And Panel1.Height() = 842 Then 'Si el panel es DinA4 en posicion vertical
6120         PrintDocument1.DefaultPageSettings.Landscape = False 'La hoja no es en modo paisaje
6121         PrintDocument1.PrinterSettings.PrinterName = "Microsoft Print to PDF" 'Impresión en PDF
6122     Try
6123         PrintDocument1.Print() 'Se prueba a crear el pdf
6124     Catch
6125         MessageBox.Show("El archivo está abierto y no se puede modificar") 'Si hay problemas no se puede crear el pdf
6126     End Try
6127 ElseIf Panel1.Width() = 842 And Panel1.Height() = 595 Then 'Si el panel es DinA4 en posicion horizontal
6128     PrintDocument1.DefaultPageSettings.Landscape = True 'La hoja es en modo paisaje
6129     PrintDocument1.PrinterSettings.PrinterName = "Microsoft Print to PDF" 'Impresión en PDF
6130     Try
6131         PrintDocument1.Print() 'Se prueba a crear el pdf
6132     Catch
6133         MessageBox.Show("El archivo está abierto y no se puede modificar") 'Si hay problemas no se puede crear el pdf
6134     End Try
6135 Else 'Si el panel es un DinA3 no hay vista preliminar, fallo con la configuración de las páginas.
6136     MessageBox.Show("No se puede crear un pdf de tamaño DinA3")
6137 End If
6138 End Sub

```

Figura 3.7.1.2.12.6. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Un cop es polsa el botó, es deseleccionen els objectes i es reestructuren per si hi ha algun problema i, acte seguit, es limita a que el graficet estigui fet en mida DIN A4 ja que en DIN A3 no es va aconseguir configurar les pàgines perquè es pogués obtenir un resultat bo. Si es DIN A4 en posició vertical, es configura la pàgina per a que sigui en mode de posició vertical modificant el booleà `PrintDocument1.DefaultPageSettings.Landscape` a fals. També es varia la propietat del control creat al menú principal del tipus `PrintDocument` que fa referència al tipus d'impressió. Aquesta variació es fa sobre el `PrintDocumen1.PrinterSettings.PrinterName` al qual se li dona la cadena "Microsoft Print to PDF" perquè generi un pdf quan es faci la impressió. Un cop s'ha fet tot això es prova de fer la impressió i si es produeix algun error s'avisava a l'usuari amb un missatge. La única diferència entre tenir un DIN A4 en posició vertical i horitzontal és el booleà que s'ha modificat en la línia anterior, que ara s'hauria de canviar a true. Ja que no es pot fer el pdf en DIN A3 en cas que s'intenti es mostra un missatge.

La segona funció dintre d'aquest nucli és la d'exportació en excel la qual consta de dos funcions que s'observen a la figura següent:

```

6140 'Función para exportar en Excel
6141 Private Sub ExportarEnFormatoExcelToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ExportarEnFormatoExcelToolStripMenuItem.Click
6142 'Se crea un diálogo de guardado
6143 Dim SaveFileDialog As New SaveFileDialog With {.InitialDirectory =
My.Computer.FileSystem.SpecialDirectories.MyDocuments, .Filter = "Archivos de graficet (*.xlsx)|*.xlsx|Todos los archivos
(*.*)|*.*"}
6144 If (SaveFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then
6145 Dim DatosActualizados = ""
6146 EscribirArchivoGuardado(DatosActualizados) 'Se guardan los datos actualizados
6147 EnlacesForm.Show() 'Se muestra la tabla de enlaces
6148 SimularToolStripMenuItem.PerformClick() 'Se obliga a clicar el botón simular para que se muestre la tabla de
entradas y salidas
6149 ExportarExcel(EnlacesForm.DataGridView1, SimulacionForm.DataGridView1, SaveFileDialog.FileName) 'Se llama a
exportar el excel
6150 EnlacesForm.Close() 'Se cierra la tabla de enlaces
6151 DetenerToolStripMenuItem.PerformClick() 'Se obliga a cerrar la simulación
6152 End If
6153 End Sub
6154
6155 'Función de exportación
6156 Private Sub ExportarExcel(ByVal tabla1 As DataGridView, ByVal tabla2 As DataGridView, ByVal ruta As String)
6157 Dim excelApp As Object = CreateObject("Excel.Application") 'Se define un objeto para contener la aplicación de tipo
excel
6158 Dim excelLibro As Object = excelApp.WorkBooks.add 'Se crea el libro del Excel
6159 Dim excelHoja As Object = excelLibro.WorkSheets(1) 'Se crea la hoja de cálculo del excel
6160 Dim espacio = 1 'Se define un espacio entre tablas de 1 casilla
6161 excelHoja.Cells(1, 1).Value = "Tabla de enlaces:" 'Se da un título en la posición A1 a la primera tabla
6162
6163 excelHoja.Cells(1, tabla1.Columns.Count + espacio + 1).Value = "Tabla de entradas, salidas y temporizadores:" 'Se da
un título a la segunda tabla
6164
6165 For ColumnasTabla1 = 0 To tabla1.Columns.Count - 1
6166 excelHoja.Cells(2, ColumnasTabla1 + 1).Value = tabla1.Columns(ColumnasTabla1).HeaderText 'Se escriben los títulos
de la primera tabla
6167 excelHoja.Cells(2, ColumnasTabla1 + 1).Interior.Color = RGB(64, 224, 208) 'Se da un fondo azul turquesa a los
títulos de la tabla
6168 excelHoja.Cells(2, ColumnasTabla1 + 1).HorizontalAlignment = 3 'Se alinean centradamente las celdas
6169 excelHoja.Cells(2, ColumnasTabla1 + 1).Font.Bold = 1 'Se define en negrita el texto
6170 Next ColumnasTabla1
6171
6172 For ColumnasTabla2 = 0 To tabla2.Columns.Count - 1 'Lo mismo que en las líneas anteriores pero ahora en la segunda
tabla
6173 excelHoja.Cells(2, tabla1.Columns.Count + espacio + ColumnasTabla2 + 1).Value =
tabla2.Columns(ColumnasTabla2).HeaderText
6174 excelHoja.Cells(2, tabla1.Columns.Count + espacio + ColumnasTabla2 + 1).Interior.Color = RGB(64, 224, 208)
6175 excelHoja.Cells(2, tabla1.Columns.Count + espacio + ColumnasTabla2 + 1).HorizontalAlignment = 3
6176 excelHoja.Cells(2, tabla1.Columns.Count + espacio + ColumnasTabla2 + 1).Font.Bold = 1
6177 Next ColumnasTabla2
6178
6179 For FilasTabla1 = 0 To tabla1.RowCount - 1 'Se recorre toda la tabla 1
6180 For ColumnasTabla1 = 0 To tabla1.Columns.Count - 1
6181 excelHoja.Cells(FilasTabla1 + 3, ColumnasTabla1 + 1).Value = tabla1.Item(ColumnasTabla1, FilasTabla1).Value
6182 'Se copian en las casillas correspondientes los valores de la tabla
excelHoja.Cells(FilasTabla1 + 3, ColumnasTabla1 + 1).HorizontalAlignment = 3 'Se alinean centradamente las
celdas
6183 Next ColumnasTabla1
6184 Next FilasTabla1
6185
6186 For FilasTabla2 = 0 To tabla2.RowCount - 1 'Se recorre toda la tabla 2
6187 For ColumnasTabla2 = 0 To tabla2.Columns.Count - 1
6188 excelHoja.Cells(FilasTabla2 + 3, tabla1.Columns.Count + espacio + ColumnasTabla2 + 1).Value =
tabla2.Item(ColumnasTabla2, FilasTabla2).Value 'Se copian en las casillas correspondientes los valores de la tabla
6189 excelHoja.Cells(FilasTabla2 + 3, tabla1.Columns.Count + espacio + ColumnasTabla2 + 1).HorizontalAlignment = 3
'Se alinean centradamente las celdas
6190 Next ColumnasTabla2
6191 Next FilasTabla2
6192 excelHoja.Columns.AutoFit 'Las columnas se adaptan al tamaño del contenido
6193 Try
6194 excelLibro.SaveAs(ruta) 'Se prueba a guardar el excel en la ruta definida
6195 Catch
6196 MessageBox.Show("El archivo donde se está intentando guardar está abierto.") 'Si hay un error se cancela
6197 End Try
6198 excelHoja = Nothing 'Se resetea la variable que contiene la hoja excel
6199 excelLibro = Nothing 'Se resetea la variable que contiene el libro de excel
6200 excelApp.Quit() 'Se cierra la aplicación
6201 excelApp = Nothing 'Se resetea la variable de la aplicación excel
6202 End Sub

```

Figura 3.7.1.2.12.7. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)



La primera funció que s'observa a la figura anterior consisteix en generar un quadre de diàleg amb l'usuari on es guardarà l'arxiu en format excel de la mateixa manera que funciona el procediment de guardar. Un cop s'accepta el quadre de diàleg, s'actualitzen les dades del grafcet en aquell instant i es truca a la classe que conté la taula d'enllaços perquè es mostri i s'obliga a fer un clic sobre la simulació perquè es generi la taula d'entrades i sortides (línies 6147 i 6148). Després es truca a la segona funció important d'aquest procediment i per últim es tanquen les dues taules anteriors.

La segona funció rep per paràmetres dues taules i una ruta on generarà l'arxiu. Les dues taules són les que s'han mencionat anteriorment. En primer lloc, es defineixen variables dintre de la funció que guarden objectes del tipus excel, la primera conté l'aplicació en sí, la segona el llibre i la tercera la fulla de càlcul. La quarta variable es un enter de valor 1 que serà la separació d'una casella entre les dues taules. Després a la cel·la que es troba a la posició (1, 1) de la fulla d'excel que es vol generar s'escriu el títol de la primera taula. Segons les columnes de la taula 1 i l'espai que es vulgui deixar entre les taules, a la posició (1, columnesTaula1 + espai + 1) de la fulla d'excel s'escriu el títol de la segona taula.

Inmediatament a sota s'escriuen les capçaleres de les taules a les posicions corresponents amb la modificació del color de fons, l'estil de la lletra i el centrat per estètica. Això s'aconsegueix fent un recorregut de les columnes de les taules i afegint a la cel·la corresponent el valor de la capçalera de la taula. Per completar la resta de les taules es recorren les dues per files i columnes i es van copiant els valors a les posicions de les cel·les adequades. Per últim s'adapten totes les cel·les al contingut amb la funció AutoFit i es prova a guardar l'arxiu a la ruta establerta per l'usuari al quadre de diàleg anterior. Si hi ha algun error es comunica i no es guarda i finalment, es restauren les variables definides dintre de la funció y es tanca la comunicació amb l'aplicació excel.

Un cop explicada la pestanya d'exportació, a continuació es presenta la funció Imprimir i Vista Preliminar les quals es governen per les següents línies de codi:

```

6081 Private Sub ImprimirToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles ImprimirToolStripMenuItem.Click
6082 Deseleccionar() 'Deselecciona todo
6083 If (PrintDialog1.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then 'Crea diàleg de impressió
6084     PrintDocument1.DocumentName = ChildForm.Text 'El nombre del documento impreso es el nombre de la ventana hija
6085     PrintDocument1.Print() 'se imprime el documento
6086 End If
6087 PrintDocument1.Dispose() 'Se libera el documento
6088 End Sub
6089
6090 Private Sub PrintDocument1_PrintPage(ByVal sender As Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs)
6091 Handles PrintDocument1.PrintPage
6092 Dim Ancho As Integer = Panel1.Width() 'Define el ancho del panel
6093 Dim Alto As Integer = Panel1.Height() 'Define la altura del panel
6094 Dim Mapa As New Bitmap(Ancho, Alto) 'Define un mapa con el tamaño del papel
6095 RedistribuirObjetos() 'Redistribuye objetos
6096 Panel1.DrawToBitmap(Mapa, Panel1.Bounds) 'Se transmite el grafcet en el mapa
6097 e.Graphics.DrawImage(Mapa, (e.PageBounds.Width - Panel1.Width) \ 2, (e.PageBounds.Height - Panel1.Height) \ 2) 'Se
6098 definen margenes de pagina
6099 ReestructurarObjetos() 'Reestructura objetos
6100 End Sub
6101
6102 'Función en pulsar el botón de vista preliminar
6103 Private Sub VistaPreliminarToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
6104 VistaPreliminarToolStripMenuItem.Click
6105 Deseleccionar() 'Deselecciona todo
6106 PrintDocument1.DocumentName = ChildForm.Text 'El nombre del documento impreso es el nombre de la ventana hija
6107 If Panel1.Width() = 595 And Panel1.Height() = 842 Then 'Si el panel es DinA4 en posición vertical
6108     PrintDocument1.DefaultPageSettings.Landscape = False 'La hoja no es en modo paisaje
6109     PrintPreviewDialog1.ShowDialog() 'Se muestra el dialogo de vista previa a la impresión
6110 ElseIf Panel1.Width() = 842 And Panel1.Height() = 595 Then 'Si el panel es DinA4 en posición horizontal
6111     PrintDocument1.DefaultPageSettings.Landscape = True 'La hoja es en modo paisaje
6112     PrintPreviewDialog1.ShowDialog() 'Se muestra el dialogo de vista previa a la impresión
6113 Else 'Si el panel es un DinA3 no hay vista preliminar, fallo con la configuración de las páginas.
6114     MessageBox.Show("No hay vista preliminar DinA3")
6115 End If
6116 End Sub

```

Figura 3.7.1.2.12.8. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

En pulsar sobre Imprimir es truca a la funció que apareix a la línia 6081 on es deseleccionen els objectes, s'obre un quadre de diàleg d'impressió (figura 3.1.12) que un cop que s'accepta canvia el nom del document a imprimir pel nom de la finestra filla, o sigui, el nom de l'arxiu generat i imprimeix, quan s'acaba llibera el document d'impressió. La segona funció que es mostra a la figura defineix les propietats d'impressió i bàsicament consisteix en crear un mapa que conté el que s'imprimeix de la mateixa manera que es fa quan es guarda un document amb uns marges de pàgina concrets que l'usuari no pot modificar. La tercera funció correspon a la vista preliminar que es pot fer per veure com quedarà el graficet dissenyat. Funciona de manera idèntica a l'exportació en pdf, però en comptes de fer la impressió en pdf es truca al quadre de diàleg PrintPreviewDialog1 que mostra una finestra (figura 3.1.11) on l'usuari podrà imprimir el diagrama també.

Abans de concloure aquest subapartat cal parlar de la funció Salir la qual permet a l'usuari tancar l'aplicació directament, de la mateixa forma que ho faria fent clic a la creu del vèrtex superior dret. El codi corresponent amb aquest mètode és el següent:

```

2641     'Función al hacer clic sobre Archivo -> Salir
2642     Private Sub SalirToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles SalirToolStripMenuItem.Click
2643         Dim resultado As DialogResult 'Crea una respuesta de diálogo entre el usuario y el programa
2644         If archivo = True Then 'Si existe un archivo creado
2645             resultado = MessageBox.Show("Si cierras no se guardará el progreso", "Cerrar", MessageBoxButtons.OKCancel,
                MessageBoxIcon.Warning) 'Nueva ventana de alerta por si no se ha guardado anteriormente
2646         If resultado = DialogResult.OK Then 'Si se acepta, se cierra el programa
2647             Application.Exit()
2648         End If
2649         'Si no se acepta no pasa nada
2650     Else
2651         Application.Exit() 'Si no hay archivo creado se cierra directamente el programa
2652     End If
2653 End Sub

```

**Figura 3.7.1.2.12.9.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Per advertir a l'usuari de que pot perdre tot el progrés si tanca l'aplicació en cas que hi hagi un arxiu creat, es genera un quadre de diàleg que si s'accepta es tanca l'aplicació i sinó no passa res i es torna a la finestra principal. Si no existeix cap arxiu es tancarà directament l'aplicació.

### 3.7.1.2.13 Procediment BorrarTodo

En alguna de les funcions detallades en el subapartat anterior s'ha mencionat el procediment BorrarTodo. Aquest, com el seu nom indica borra tot el que aparegui en el panell i restaura totes les variables. Un fragment del codi d'aquest mètode es pot observar a continuació:



```

2678 'Función que elimina todos los elementos existentes en el panel y libera las variables
2679 Public Sub BorrarrTodo()
2680     DatosGuardados = ""
2681     DatosCopiados = ""
2682     DatosGuardadosDeshacer = ""
2683     LineaForm.GuardarEnLaces = ""
2684     Dim BorrarrListas = {listaEtapa, listaLabsEtapa, listaMarcaLabsEtapa, TransicionesForm.listaTrans, listaLabsTrans,
TransicionesForm.listaTransAlto, listaEstadoAlto, listaLabsTransAlto, TransicionesForm.listaTransBajo, listaEstadoBajo,
listaLabsTransBajo, AccionForm.listaAccion, listaLabsAccion, AccionAsociadaForm.listaAccionAsociada, listaLabsAccionAsociada,
listaEtiquetas, ConvergenciaForm.listaTransConvergencia, listaLabsTransConvergencia,
ConvergenciaForm.listaTransAltoConvergencia, listaEstadoAltoConvergencia, listaLabsTransAltoConvergencia,
ConvergenciaForm.listaTransBajoConvergencia, listaEstadoBajoConvergencia, listaLabsTransBajoConvergencia,
DivergenciaForm.listaTransDivergencia, listaLabsTransDivergencia, DivergenciaForm.listaTransAltoDivergencia,
listaEstadoAltoDivergencia, listaLabsTransAltoDivergencia, DivergenciaForm.listaTransBajoDivergencia,
listaEstadoBajoDivergencia, listaLabsTransBajoDivergencia, LineaForm.listaLineaVertical, LineaForm.listaLineaHorizontal,
TimerForm.listaTimer, TimerForm.listaTransTimer, TimerForm.listaTransTimerNegado, listaLabsTransTimer, listaLabsTimer,
listaLabsTransTimerNegado, TimerForm.listaTimerAsociado, TimerForm.listaTransTimerAsociado,
TimerForm.listaTransTimerAsociadoNegado, listaLabsTransTimerAsociado, listaLabsTimerAsociado,
listaLabsTransTimerAsociadoNegado}
2685     For ind = 0 To BorrarrListas.LongCount - 1
2686         BorrarrListas(ind).Clear()
2687     Next ind
2688     Try
2689         If Me.Contains(Panel1) Then
2690             Me.Controls.Remove(Panel1)
2691         End If
2692         If Me.Panel1.Contains(etapa0) Then
2693             Me.Panel1.Controls.Remove(etapa0)
2694             Me.Panel1.Controls.Remove(Label0)
2695             Me.Panel1.Controls.Remove(LabelMarca0)
2696             etapa0 = Nothing
2697             Label0 = Nothing
2698
2699             LabelMarca0 = Nothing
2700             BotonEtapa0.Enabled() = True
2701         End If
2702         For ind = 0 To listaEtapa.LongCount - 1
2703             Me.Panel1.Controls.Remove(listaEtapa(ind))
2704             Me.Panel1.Controls.Remove(listaLabsEtapa(ind))
2705             Me.Panel1.Controls.Remove(listaMarcaLabsEtapa(ind))
2706         Next ind

```

Figura 3.7.1.2.13.1. Captura d’imatge part del codi annex A1. (Font: Pablo Mañas)

Dintre d’aquesta funció inicialment s’esborren les variables globals del tipus cadena igualant-les a una cadena buida i totes les llistes amb tots els objectes a dintre també es netegen utilitzant una matriu que es recorre en un bucle i amb la funció Clear elimina tots els ítems i deixa la variable nova. A més s’esborren del panell tots els objectes possibles existents al panell de la mateixa manera que es fa quan es fa l’eliminació de la selecció (subapartat 3.7.1.2.13).

### 3.7.1.2.14 Procediments relacionats amb la pestanya Editar

A la dreta de la pestanya explicada anteriorment es troba Editar, aquesta permet funcions com desfer, refer, retallar, copiar, enganxar i seleccionar tots el elements. Seguint amb un ordre de dalt a baix es comença amb l’opció de desfer. En fer clic es dirigeix al mètode següent:

```

5669 'Función del botón deshacer
5670 Private Sub DeshacerToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles DeshacerToolStripMenuItem.Click
5671     DeshacerToolStripMenuItem.Enabled = False 'Inhabilita el botón deshacer
5672     RehacerToolStripMenuItem.Enabled = True 'Habilita el botón rehacer
5673     EscribirArchivoGuardado(DatosGuardadosDeshacer) 'Guarda datos antes de deshacer
5674     Dim SepararDatos() As String 'Define variable que separa datos anteriores
5675     If DatosGuardados <> "" Then 'Si hay datos guardados anteriores
5676         SepararDatos = DatosGuardados.Split("|") 'Separa datos anteriores
5677         RecuperarPosicionAnterior(SepararDatos) 'Recupera los objetos con los datos anteriores
5678     End If
5679 End Sub

```

Figura 3.7.1.2.14.1. Captura d’imatge part del codi annex A1. (Font: Pablo Mañas)

En primer lloc s’inhabilita el propi botó en fer clic ja que només es permet desfer una acció anterior i s’habilita el botó de refer per a que es pugui tornar abans de l’acció desfeta. Després es guarda a la variable DatosGuardadosDeshacer les dades del GRAFCET que s’utilitzaran a l’acció de refer. A continuació es defineix una llista del tipus cadena (de fet és una matriu de base 0, similar a una llista) que permet separar les dades actuals, o sigui, les dades de la variable DatosGuardados amb un Split.

Finalment es truca a la funció `RecuperarPosicionAnterior` amb el paràmetre de la llista anterior per tornar a l'acció anterior. En cas que es volgués desfer més d'una acció, en comptes d'emmagatzemar les dades en una sola variable com pot ser `DatosGuardados`, s'hauria de crear una llista amb un número d'accions determinades que s'anés actualitzant a cada acció i esborrant l'acció més antiga cada cop que s'arribés al límit de la llista. Fer-ho complica el programa pel fet del maneig de les dades i es va descartar l'opció tot i que es podria implementar més endavant.

Ja que s'ha parlat en dues ocasions sobre el procediment de `RecuperarPosicionAnterior`, fent un petit incís, en aquest moment s'explica com és la seva estructura i com funciona. Per començar, es pot observar un extracte del codi a la figura següent:

```

5681 'Función que recupera las posiciones anteriores y datos anteriores, si no existe el objeto porque ha sido borrado se crea
de nuevo con las características de los datos anteriores.
5682 Private Sub RecuperarPosicionAnterior(datos() As String)
5683     For ind = 0 To datos.LongCount - 1
5684         If datos(ind) = "Etapa0" Then
5685             Try
5686                 etapa0.Left = datos(ind + 1)
5687             Catch
5688                 ConstruirEtapa0()
5689                 ConstruirTextoEtapa0()
5690                 ConstruirTextoMarcaEtapa0()
5691                 RehacerToolStripMenuItem.Enabled = False
5692             End Try
5693             etapa0.Left = datos(ind + 1)
5694             etapa0.Top = datos(ind + 2)
5695             Label0.Left = etapa0.Left + (etapa0.Width() / 3)
5696             Label0.Top = etapa0.Top + (etapa0.Height() / 3)
5697             LabelMarca0.Left = etapa0.Left - 1.2 * etapa0.Width()
5698             LabelMarca0.Top = etapa0.Top + (etapa0.Height() / 3)
5699             BotonEtapa0.Enabled = False
5700         End If
5701         If datos(ind) = "Etapa" Then
5702             Try
5703                 listaEtapa(CInt(datos(ind + 5))).Left = CInt(datos(ind + 1))
5704             Catch
5705                 ConstruirEtapa()
5706                 ConstruirTextoMarcaEtapa()
5707                 ConstruirTextoEtapa()
5708                 RehacerToolStripMenuItem.Enabled = False
5709                 EventosRatonEtapa()
5710                 EventosRatonLabEtapa()
5711             End Try
5712             listaEtapa(CInt(datos(ind + 5))).Left = CInt(datos(ind + 1))
5713             listaEtapa(CInt(datos(ind + 5))).Top = CInt(datos(ind + 2))
5714             listaLabsEtapa(CInt(datos(ind + 5))).Text = datos(ind + 3)
5715             listaMarcaLabsEtapa(CInt(datos(ind + 5))).Text = datos(ind + 4)
5716             listaLabsEtapa(CInt(datos(ind + 5))).Left = listaEtapa(CInt(datos(ind + 5))).Left + (listaEtapa(CInt(datos(ind
+ 5))).Width() / 3)
5717             listaLabsEtapa(CInt(datos(ind + 5))).Top = listaEtapa(CInt(datos(ind + 5))).Top + (listaEtapa(CInt(datos(ind +
5718             listaMarcaLabsEtapa(CInt(datos(ind + 5))).Left = listaEtapa(CInt(datos(ind + 5))).Left - 1.3 *
5719             listaEtapa(CInt(datos(ind + 5))).Width()
5720             listaMarcaLabsEtapa(CInt(datos(ind + 5))).Top = listaEtapa(CInt(datos(ind + 5))).Top +
5721             (listaEtapa(CInt(datos(ind + 5))).Height() / 3)
5722         End If

```

**Figura 3.7.1.2.14.2.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

A la capçalera del procediment s'observa que cal proporcionar un paràmetre del tipus matriu de cadenes de base 0. Amb un bucle es recorre aquest argument i si va trobant les paraules clau que identifiquen als objectes com pot ser `Etapa0`, `Etapa`, `Transicion`, etc; s'estableix una estructura `Try-Catch` que consisteix en provar a moure l'objecte on la dada li ordena (línia 5686) i si no es pot (voldria dir que l'objecte s'ha esborrat) es torna a construir l'objecte juntament amb les seves etiquetes (línies 5688, 5689, 5690). Com en cas que es torni a crear no es vol que al refer s'esborri l'objecte, s'inhabilita aquest botó. Un cop s'acaba l'estructura `try-catch` es situa l'objecte en la posició que estava anteriorment juntament amb les seves etiquetes. En el cas de les etapes també cal activar els esdeveniments de ratolí i el moviment és més complex ja que cal cercar a les dades primerament l'índex de l'etapa que es guardava a la posició 5 després de l'etiqueta identificadora.

Per exemple, si es mira la línia 5712 del codi a la llista d'etapes es pot observar com funciona aquest procediment. La llista d'etapes conté tots els objectes PictureBox amb les etapes creades. Cada objecte, per tant, té un índex de la llista. Aquest índex quan es guarden les dades queda emmagatzemat a una posició 5 a partir de l'etiqueta "Etapa". Així doncs, quan es vol recuperar l'estat anterior sempre s'ha de dirigir a aquest índex. Seguint amb l'exemple, l'atribut Left es va guardar a la primera posició després de l'etiqueta identificadora, per això, s'igualava:

```
llistaObjectes(índex de l'objecte).Left = Cint(índex posició Left de l'objecte)
```

L'índex de l'objecte sempre serà l'índex que es recorre en bucle + 5 per la posició i l'índex de la posició Left serà l'índex del bucle + 1. Això es manté per cadascuna de les propietats emmagatzemades necessàries per restablir l'acció anterior. Tot el mostrat a la captura 3.7.1.2.14.2 es continua amb cadascun del tipus d'objecte, però el funcionament és el mateix encara que els índexs dels objectes poden canviar en funció del tipus.

Cal dir que aquesta funció és la mateixa que s'utilitza per carregar les dades inicials quan s'obre un arxiu ja que quan s'obre un document es pot obtenir la matriu en base 0 de la cadena a partir de les dades que s'han escrit al document guardat de tipus .txt.

Després d'aquest parèntesis es continua amb l'explicació del funcionament de la funció refer. Així doncs, a continuació s'observa el codi corresponent:

```
6071 'Función del botón rehacer
6072 Private Sub RehacerToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles RehacerToolStripMenuItem.Click
6073     RehacerToolStripMenuItem.Enabled = False 'Inhabilita el botón rehacer
6074     DeshacerToolStripMenuItem.Enabled = True 'Habilita el botón deshacer
6075     Dim SepararDatosDeshacer() As String 'Define variable que separa los datos guardados al antes de deshacer
6076     SepararDatosDeshacer = DatosGuardadosDeshacer.Split("|") 'Separa los datos antes de deshacer
6077     RecuperarPosicionAnterior(SepararDatosDeshacer) 'Recupera los datos antes de deshacer
6078 End Sub
```

**Figura 3.7.1.2.14.3.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Simplement quan es clica sobre el botó s'inverteix l'habilitació que s'havia fet amb el desfer. Es defineix una variable del tipus matriu de cadena en base 0 que permet separar la variable DatosGuardadosDeshacer on es guardaven les dades del diagrama abans de realitzar l'acció de desfer. Finalment es truca al procediment descrit al incís realitzat anteriorment per tornar a la posició abans de produir-se el desfer. Si s'hagués creat la llista amb diferents accions anteriors s'hauria d'anar recorrent des de l'última acció fins la més antiga.

Continuant ara sense parèntesis en mig amb la pestanya Editar, es troba el botó Cortar. El seu codi corresponent és el següent:

```

5205 'Función cortar (se eliminan los objetos previamente se copian):
5206 Private Sub CortarToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles CortarToolStripMenuItem.Click
5207 My.Computer.Clipboard.Clear()
5208 DatosCopiados = ""
5209 CopiarPortafolio()
5210 If Me.Contains(Panel1) Then
5211     EliminarSeleccion()
5212 End If
5213 End Sub

```

**Figura 3.7.1.2.14.4.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

En primer lloc s'esborra tot el que hi hagi al portafolis i la variable que conté les dades copiades, es copien les dades al portafolis i s'eliminen els elements seleccionats. Així doncs, el botó Copiar funciona de la mateixa manera però sense eliminar els elements seleccionats.

```

5047 'Función para copiar en el portafolios los objetos seleccionados:
5048 Private Sub CopiarToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles CopiarToolStripMenuItem.Click
5049     Clipboard.Clear() 'Se limpia el portafolios
5050     DatosCopiados = ""
5051     CopiarPortaFolio()
5052 End Sub
5053
5054 Private Sub CopiarPortaFolio()
5055     If Me.Contains(Panel1) Then
5056         If Me.Panel1.Contains(etapa0) Then
5057             If etapa0.Tag = True Then
5058                 DatosCopiados &= "etapa0|"
5059             End If
5060         End If
5061         For ind = 0 To listaEtapa.LongCount - 1
5062             If listaEtapa(ind).Tag = True Then
5063                 DatosCopiados &= "etapa" + "|"
5064                 DatosCopiados &= listaLabsEtapa(ind).Text.ToString + "|"
5065                 DatosCopiados &= listaMarcalabsEtapa(ind).Text.ToString + "|"
5066             End If
5067         Next ind
5068         For ind = 0 To TransicionesForm.listaTrans.LongCount - 1
5069             If TransicionesForm.listaTrans(ind).Tag = True Then
5070                 DatosCopiados &= "Transicion" + "|"
5071                 DatosCopiados &= listaLabsTrans(ind).Text.ToString + "|"
5072             End If
5073         Next ind
5074
5075         If DatosCopiados <> "" Then
5076             My.Computer.Clipboard.SetText(DatosCopiados) 'Se copia la variable en el portafolios
5077         End If
5078     End Sub

```

**Figura 3.7.1.2.14.5.** Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

Així com s'observa a la figura anterior, la funció de copiar al portafolis consisteix en guardar a la variable DatosCopiados la informació dels objectes existents sense guardar posicions ja que quan s'enganxin ho faran a la posició inicial. Finalment s'emmagatzema la informació copiada al portafolis. En cas que es vulguessin enganxar els elements copiats sobre la posició del ratolí mantenint l'estructura s'haurien de guardar les posicions dels objectes també i complicar la part de codi en que es realitza la funció d'enganxar. Finalment es va decidir copiar únicament els objectes existents sense tenir en compte posicions i que a l'hora d'enganxar s'enganxessin a la posició original on es creen els objectes segons el tipus.

El cinquè ítem de la pestanya és la casella de Pegar que consisteix com ja s'acaba de mencionar en enganxar les dades copiades en el procediment anterior. El codi elaborat és el següent:

```

5487 'Funcion pegar objetos copiados/cortados:
5488 Private Sub PegarToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles PegarToolStripMenuItem.Click
5489     EscribirArchivoGuardado(DatosGuardados) 'Guarda datos anteriores
5490     RehacerToolStripMenuItem.Enabled = False 'Desactiva botón Rehacer
5491     DeshacerToolStripMenuItem.Enabled = True 'Activa botón Deshacer
5492     If DatosCopiados <> "" Then 'si la variable tiene algun dato copiado
5493         Dim LeerPortafolio As String 'Define una variable para leer el portafolio
5494         Dim SepararDatosCopiados() As String 'Define una variable que separa los datos
5495         LeerPortafolio = My.Computer.Clipboard.GetText 'Lee y almacena en la variable el contenido del portafolio
5496         SepararDatosCopiados = LeerPortafolio.Split("|") 'Se separan los datos en la lista
5497         For ind = 0 To SepararDatosCopiados.LongCount - 1 'Se recorren todos los datos
5498             If SepararDatosCopiados(ind) = "etapa0" Then 'En función de los datos copiados se crean nuevas etapas,
                    transiciones, acciones, etc...Aparecen en el sitio definido por la Clase en cuestión pero se mantienen los textos
                    MessageBox.Show("No es pot copiar l'etapa inicial")
5499             End If
5500             If SepararDatosCopiados(ind) = "Etapa" Then
5501                 ConstruirEtapa()
5502                 ConstruirTextoEtapa()
5503                 listaLabsEtapa(listaLabsEtapa.LongCount - 1).Text = SepararDatosCopiados(ind + 1)
5504                 ConstruirTextoMarcaEtapa()
5505                 listaMarcaLabsEtapa(listaMarcaLabsEtapa.LongCount - 1).Text = SepararDatosCopiados(ind + 2)
5506                 EventosRatonEtapa()
5507                 EventosRatonLabEtapa()
5508             End If
5509             If SepararDatosCopiados(ind) = "Transicion" Then
5510                 TransicionesForm.ConstruirTransicion()
5511                 TransicionesForm.CrearLabelTransicion()
5512                 listaLabsTrans(listaLabsTrans.LongCount - 1).Text = SepararDatosCopiados(ind + 1)
5513                 EventosRatonTrans()
5514                 EventosRatonLabTrans()
5515             End If
5516         End If

```

Figura 3.7.1.2.14.6. Captura d’imatge part del codi annex A1. (Font: Pablo Mañas)

Per evitar problemes d’execució es guarden les dades i s’inhabilita el botó de refer i s’habilita el botó de desfer. Després, si n’hi ha hagut alguna dada copiada es defineix una variable que llegeix el contingut del portafolis, el qual es separa i es construeixen els objectes de nou, mantenint els textos de les etiquetes associades. En el cas de l’etapa inicial es considera que no es pot copiar i s’avisava amb un missatge.

Finalment dintre d’aquest subapartat es troba el botó Seleccionar Todo. Aquest funciona de la mateixa manera que el Deseleccionar ja explicat anteriorment però ara deixant seleccionats tots els elements i canviant els atributs tag dels quals en true. Un fragment del codi elaborat es mostra a la imatge següent:

```

3654 'Función para seleccionar todos los elementos del panel en el caso que existan en el panel:
3655 Private Sub SeleccionarTodoToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
                    SeleccionarTodoToolStripMenuItem.Click
3656     Dim matrizListaTrans = {TransicionesForm.listaTrans, TransicionesForm.listaTransAlto, TransicionesForm.listaTransBajo,
                    TimerForm.listaTransTimer, TimerForm.listaTransTimerAsociado, TimerForm.listaTransTimerNegado,
                    TimerForm.listaTransTimerAsociadoNegado}

3657     Dim matrizListaConv = {ConvergenciaForm.listaTransConvergencia, ConvergenciaForm.listaTransAltoConvergencia,
                    ConvergenciaForm.listaTransBajoConvergencia}
3658     Dim matrizListaDiv = {DivergenciaForm.listaTransDivergencia, DivergenciaForm.listaTransAltoDivergencia,
                    DivergenciaForm.listaTransBajoDivergencia}
3659     If Panel1.Contains(etapa0) Then
3660         etapa0.Image = Image.FromFile("Imágenes\Etapa0Sel.jpg")
3661         etapa0.Tag = True
3662     End If
3663     If Panel1.Contains(etapa) Then
3664         For ind = 0 To listaEtapa.LongCount - 1
3665             listaEtapa(ind).Image = Image.FromFile("Imágenes\EtapaSel.jpg")
3666             listaEtapa(ind).Tag = True
3667         Next ind
3668     End If
3669     If Panel1.Contains(TransicionesForm.transicion) Then
3670         For ind = 0 To matrizListaTrans.LongCount - 1
3671             For ind1 = 0 To matrizListaTrans(ind).LongCount - 1
3672                 matrizListaTrans(ind)(ind1).Image = Image.FromFile("Imágenes\TransicionSel.png")
3673                 matrizListaTrans(ind)(ind1).Tag = True
3674             Next ind1
3675         Next ind
3676     End If

```

Figura 3.7.1.2.14.7. Captures d’imatge part del codi annex A1. (Font: Pablo Mañas)

Així com en altres procediments, s’uneixen les llistes amb característiques similars en matrius i es recorren totes les llistes i objectes que es troben al panell. Es canvien les propietats de tag a true i les imatges a objectes seleccionats.

### 3.7.1.2.15 Procediments pestanya Ver

En un principi a aquesta pestanya es trobaven les funcions de Zoom, però per problemes amb la cerca d'enllaços i la reestructuració dels objectes es van descartar. Tot i així dintre d'aquesta pestanya es troba un element important el qual obre la finestra per veure la taula d'enllaços. Aquesta casella és la de primera la qual s'anomena Ver enllaces.

```

6205 'Función al clicar el botón de ver enlaces
6206 Private Sub VerEnlacesToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles VerEnlacesToolStripMenuItem.Click
6207     Dim DatosActualizados = ""
6208     EscribirArchivoGuardado(DatosActualizados) 'Se guardan los datos actuales
6209     EnlacesForm.Show() 'Se llama a la clase enlacesForm para mostrar la tabla de enlaces
6210     EnlacesForm.Top = ToolStrip1.Top + ToolStrip1.Height() + 40 'Se sitúa en una zona específica de la pantalla
6211     EnlacesForm.Left = Splitter2.Left - EnlacesForm.Width() - 10
6212 End Sub

```

**Figura 3.7.1.2.15.1.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Així com es pot observar a la figura anterior, en fer clic al botó corresponent el programa defineix una variable nova, emmagatzema les dades del diagrama al seu interior, truca a la classe EnlacesForm per mostrar la taula d'enllaços i la desplaça a una zona de la pantalla perquè es vegi millor.

En aquest instant és necessari fer un incís, ja que no s'ha explicat com es crea la taula d'enllaços. Bé doncs, ara és el moment de fer-ho ja que s'ha parlat anteriorment i en aquest punt és interessant. Sempre que es truca al procediment EscribirArchivoGuardado al final d'aquest es desenvolupa el procediment ActualizarTablaEnlaces amb les mateixes dades que han sigut guardades. El codi d'aquesta funció és el següent:

```

7564 'Función que actualiza la tabla de enlaces llamando a las funciones anteriores para ello. Escribe las líneas en la tabla
7565 de enlaces obtenidas con las funciones anteriores.
7566 Public Sub ActualizarTablaEnlaces(sender As String)
7567     Dim SepararDatos() As String = {}
7568     Dim contadorEtapa As Integer = 0
7569     If sender <> "" Then
7570         SepararDatos = sender.Split("|")
7571         Try
7572             For ind = 0 To SepararDatos.LongCount - 1
7573                 If SepararDatos(ind) = "Etapa0" Then
7574                     Dim filaEtapa0() As String = {"", "", "", "", "", "", "", "", "", ""}
7575                     Try
7576                         filaEtapa0(0) = Label0.Text.ToString
7577                     Catch
7578                         filaEtapa0(0) = ""
7579                     End Try
7580                     Try
7581                         filaEtapa0(1) = LabelMarca0.Text
7582                     Catch
7583                         filaEtapa0(1) = ""
7584                     End Try
7585                     BuscarEnlacesEtapa(filaEtapa0, SepararDatos, ind)
7586                     Try
7587                         EnlacesForm.DataGridView1.Rows(0).SetValues(filaEtapa0)
7588                     Catch
7589                         EnlacesForm.DataGridView1.Rows.Add()
7590                         EnlacesForm.DataGridView1.Rows(0).SetValues(filaEtapa0)
7591                     End Try
7592                 End If
7593             End For
7594         End Try
7595     End If

```

```

7592         If SepararDatos(ind) = "Etapa" Then
7593             contadorEtapa += 1
7594             Dim filaEtapa() As String = {"", "", "", "", "", "", "", "", ""}
7595             Try
7596                 filaEtapa(0) = SepararDatos(ind + 3)
7597             Catch
7598                 filaEtapa(0) = ""
7599             End Try
7600             Try
7601                 filaEtapa(1) = SepararDatos(ind + 4)
7602             Catch
7603                 filaEtapa(1) = ""
7604             End Try
7605             BuscarEnlacesEtapa(filaEtapa, SepararDatos, ind)
7606             If Panel1.Contains(etapa0) Then
7607                 Try
7608                     EnlacesForm.DataGridView1.Rows(contadorEtapa).SetValues(filaEtapa)
7609                 Catch
7610                     EnlacesForm.DataGridView1.Rows.Add()
7611                     EnlacesForm.DataGridView1.Rows(contadorEtapa).SetValues(filaEtapa)
7612                 End Try
7613             Else
7614                 Try
7615                     EnlacesForm.DataGridView1.Rows(contadorEtapa - 1).SetValues(filaEtapa)
7616                 Catch
7617                     EnlacesForm.DataGridView1.Rows.Add()
7618                     EnlacesForm.DataGridView1.Rows(contadorEtapa - 1).SetValues(filaEtapa)
7619                 End Try
7620             End If
7621         End If
7622     Next ind
7623 Catch
7624     MessageBox.Show("No hay ningún enlace") 'Si hay un error se muestra que no hay enlaces
7625 End Try
7626 End If

7627 ActualizarEnlacesManuales() 'Se llama a la función de enlaces manuales
7628 End Sub

```

Figura 3.7.1.2.15.2. Captura d’imatge part del codi annex A1. (Font: Pablo Mañas)

En primer lloc la funció anterior és pública perquè es pugui trucar des d’altres classes i el paràmetre que rep sempre són les dades actuals del diagrama que inicialment es separen com ja s’ha fet en procediments explicats anteriorment. També es defineix un comptador que guardarà el número d’etapes existents. Davant de tot es troba una estructura Try-Catch que mai deuria d’activar-se, però que s’ha deixat per si es produeix algun error al programa. Inicialment es fa un recorregut amb totes les dades i en cas que es trobi una etiqueta d’etapa0 (línia 7573) es genera una llista buida de cadenes de dimensió 9 corresponent a les columnes de la taula d’enllaços. Després es prova d’omplir la primera dada amb l’etiqueta de l’etapa i la segona amb la marca, en cas que no existeixi etapa inicial es deixen com a cadenes buides. Després es truca a la funció de buscar enllaços la qual s’exposa immediatament després d’aquest incís. Dintre d’aquest mètode s’acaba d’omplir la cadena definida anteriorment i més endavant s’escriu a la primera fila de la taula el contingut de la cadena. En cas que no es pugui primerament es crea la fila a la taula (línia 7588). Així com es fa amb l’etapa inicial es farà amb totes les etapes de la mateixa manera però amb el comptador que controla les files creades a la taula en funció de si existeix etapa inicial o no. Finalment s’actualitzarien els enllaços que l’usuari ha decidit de forma manual trucant al procediment ActualizarEnlacesManuales, el qual s’explicarà quan sigui el moment.

Abans de continuar cal explicar la funció BuscarEnlacesEtapa, la qual agafa els paràmetres de les llistes de cadenes generades, les dades i l’índex del bucle per trobar tots els enllaços de les etapes. Ja que la funció és molt robusta i anida nombroses funcions més s’intentarà explicar de forma detallada des de l’aspecte general a les petites singularitats. El codi inicialment presenta el següent:

```

7232 'Función que busca enlaces etapas según posición de elementos y escribe una fila de la tabla de enlaces
7233 Private Sub BuscarEnlacesEtapa(filaEtapa0() As String, SepararDatos() As String, ind As Integer)
7234 Dim MúltiplesEnlacesSigüientes As Boolean = False
7235 Dim MúltiplesEnlacesAnteriores As Boolean = False
7236 Dim MúltiplesEtapasAnteriores As Boolean = False
7237 For ind2 = 0 To SepararDatos.LongCount - 1
7238     If SepararDatos(ind2) = "Transicion" Then
7239         If (CInt(SepararDatos(ind2 + 1)) = CInt(SepararDatos(ind + 1))) And (CInt(SepararDatos(ind2 + 2)) =
7240             CInt(SepararDatos(ind + 2)) + 30) Then
7241             If MúltiplesEnlacesSigüientes = False Then
7242                 filaEtapa0(6) = "Trans.Nivel: " + SepararDatos(ind2 + 3) + ", "
7243                 MúltiplesEnlacesSigüientes = True
7244             Else
7245                 filaEtapa0(6) &= SepararDatos(ind2 + 3) + ", "
7246             End If
7247         End If
7248         If (CInt(SepararDatos(ind2 + 1)) = CInt(SepararDatos(ind + 1))) And (CInt(SepararDatos(ind2 + 2)) =
7249             CInt(SepararDatos(ind + 2)) - 30) Then
7250             If MúltiplesEnlacesAnteriores = False Then
7251                 filaEtapa0(7) = "Trans.Nivel: " + SepararDatos(ind2 + 3) + ", "
7252                 BusquedaEnlaceEtapa(filaEtapa0, SepararDatos, ind2, MúltiplesEtapasAnteriores)
7253                 MúltiplesEnlacesAnteriores = True
7254             Else
7255                 filaEtapa0(7) &= SepararDatos(ind2 + 3) + ", "
7256                 BusquedaEnlaceEtapa(filaEtapa0, SepararDatos, ind2, MúltiplesEtapasAnteriores)
7257             End If
7258         End If
7259     End If
7260 End Sub

```

Figura 3.7.1.2.15.3. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Inicialment s'observa que apareixen 3 noves variables de tipus booleà que inicialitzen a fals. Aquestes serveixen per observar si n'hi ha varis enllaços en una mateixa etapa. Es fa un segon recorregut amb les dades inicials de la funció anterior per comprovar objecte per objecte les posicions relatives entre etapa i objecte. Si n'hi ha una bona coincidència per crear l'enllaç s'escriu a la posició de la llista de la cadena que després es mostra a la taula d'enllaços. En alguns casos, que n'hi hagi coincidència en posició no cal, ja que també fa falta observar si després de l'enllaç es troba una etapa la qual també caldria afegir a la taula. Així doncs, es truca a la segona funció BusquedaEnlaceEtapa el codi de la qual té un aspecte com el següent:

```

6330 'Función que busca los enlaces probables con una etapa según la posición de los elementos:
6331 Private Sub BusquedaEnlaceEtapa(filaEtapa0() As String, SepararDatos() As String, ind2 As Integer, ByRef
6332     MúltiplesEtapasAnteriores As Boolean)
6333     For ind3 = 0 To SepararDatos.LongCount - 1
6334         If SepararDatos(ind3) = "Etapa" Then
6335             If (CInt(SepararDatos(ind2 + 1)) = CInt(SepararDatos(ind3 + 1))) And (CInt(SepararDatos(ind2 + 2)) =
6336                 CInt(SepararDatos(ind3 + 2)) + 30) Then
6337                 If MúltiplesEtapasAnteriores = False Then
6338                     filaEtapa0(5) = "Etapa: " + SepararDatos(ind3 + 3) + ", "
6339                     MúltiplesEtapasAnteriores = True
6340                 Else
6341                     filaEtapa0(5) &= SepararDatos(ind3 + 3) + ", "
6342                 End If
6343             End If
6344             If SepararDatos(ind3) = "Etapa0" Then
6345                 If (CInt(SepararDatos(ind2 + 1)) = CInt(SepararDatos(ind3 + 1))) And (CInt(SepararDatos(ind2 + 2)) =
6346                 CInt(SepararDatos(ind3 + 2)) + 30) Then
6347                     If MúltiplesEtapasAnteriores = False Then
6348                         filaEtapa0(8) = "Etapa: 0, "
6349                         MúltiplesEtapasAnteriores = True
6350                     Else
6351                         filaEtapa0(8) &= "0, "
6352                     End If
6353                 End If
6354             End If
6355             If SepararDatos(ind3) = "LíneaVertical" Then
6356                 If (CInt(SepararDatos(ind2 + 1)) = CInt(SepararDatos(ind3 + 1)) - 15) And (CInt(SepararDatos(ind2 + 2)) =
6357                 CInt(SepararDatos(ind3 + 2)) + CInt(SepararDatos(ind3 + 3))) Then
6358                 For ind4 = 0 To SepararDatos.LongCount - 1
6359                     If SepararDatos(ind4) = "Etapa" Then
6360                         If (CInt(SepararDatos(ind4 + 1)) >= CInt(SepararDatos(ind3 + 1))) And (CInt(SepararDatos(ind3 +
6361                         1)) < CInt(SepararDatos(ind4 + 1)) + 30) And (CInt(SepararDatos(ind3 + 2)) = CInt(SepararDatos(ind4 + 2)) + 30) Then
6362                             If MúltiplesEtapasAnteriores = False Then
6363                                 filaEtapa0(8) = "Etapa: " + SepararDatos(ind4 + 3) + ", "
6364                                 MúltiplesEtapasAnteriores = True
6365                             Else
6366                                 filaEtapa0(8) &= SepararDatos(ind4 + 3) + ", "
6367                             End If
6368                         End If
6369                     End If
6370                 End If
6371             End If
6372         End If
6373     End Sub

```

Figura 3.7.1.2.15.4. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Aquesta funció torna a recórrer les dades i busca si n'hi ha una etapa anterior a l'etapa que s'està tractant, per tant, primerament ha d'haber un enllaç i després una etapa enllaçada en aquest enllaç. En cas que aparegui una línia vertical o una línia horitzontal es produeix un altre bucle en que es busca l'etapa i així es van encadenant bucles i bucles fins que es considera que no és necessari cercar més.



Per ajudar a l'explicació es recomana mirar la figura 3.4.2. Fixant-nos en l'etapa 3 té un enllaç anterior immediatament a sobre, però després, per arribar a enllaçar-se amb l'etapa inicial té 2 línies verticals i una d'horitzontal, per això és important recórrer tants cops les dades i encadenar bucles ja que per cada element es necessita fer un bucle.

Continuant amb el codi de la cerca d'enllaços (figura 3.7.1.2.15.3) es continuaria amb cadascun dels objectes i en cas que l'etapa trobés una línia vertical amunt o avall enllaçada es descriuria el següent:

```

7464         If SepararDatos(ind2) = "LineaVertical" Then
7465             'Línea vertical debajo de la etapa:
7466             If (CInt(SepararDatos(ind2 + 1)) >= CInt(SepararDatos(ind + 1))) And (CInt(SepararDatos(ind2 + 1)) <
CInt(SepararDatos(ind + 1)) + 30) And (CInt(SepararDatos(ind2 + 2)) = CInt(SepararDatos(ind + 2)) + 30) Then
7467                 For ind5 = 0 To SepararDatos.LongCount - 1
7468                     EscribirEnlacesSiguientes(filaEtap0, SepararDatos, ind2, ind5, MúltiplesEnlacesSiguientes)
7469                     If SepararDatos(ind5) = "LineaHorizontal" Then
7470                         'Línea horizontal unida con línea vertical
7471                         If ((CInt(SepararDatos(ind5 + 2)) >= CInt(SepararDatos(ind2 + 2))) And (CInt(SepararDatos(ind5 +
2)) < CInt(SepararDatos(ind2 + 2)) + CInt(SepararDatos(ind2 + 3))) And (CInt(SepararDatos(ind5 + 1)) = CInt(SepararDatos(ind2
+ 1)) + 1) Or ((CInt(SepararDatos(ind5 + 2)) >= CInt(SepararDatos(ind2 + 2))) And (CInt(SepararDatos(ind5 + 2)) <
CInt(SepararDatos(ind2 + 2)) + CInt(SepararDatos(ind2 + 3))) And (CInt(SepararDatos(ind5 + 1)) + CInt(SepararDatos(ind5 + 3))
= CInt(SepararDatos(ind2 + 1))) Or ((CInt(SepararDatos(ind2 + 1)) >= CInt(SepararDatos(ind5 + 1))) And
(CInt(SepararDatos(ind2 + 1)) < CInt(SepararDatos(ind5 + 1)) + CInt(SepararDatos(ind5 + 3))) And (CInt(SepararDatos(ind5 + 2))
= CInt(SepararDatos(ind2 + 2)) + CInt(SepararDatos(ind2 + 3)))) Then
7472                             BucleLineaHorizontalMasLineaVertical(filaEtap0, SepararDatos, ind5,
MúltiplesEnlacesSiguientes, MúltiplesEnlacesAnteriores, MúltiplesEtapasAnteriores)
7473                             End If
7474                         End If
7475                     Next ind5
7476                 End If
7477             'Línea vertical encima de la etapa:
7478             If (CInt(SepararDatos(ind2 + 1)) >= CInt(SepararDatos(ind + 1))) And (CInt(SepararDatos(ind2 + 1)) <
CInt(SepararDatos(ind + 1)) + 30) And (CInt(SepararDatos(ind2 + 2)) + CInt(SepararDatos(ind2 + 3)) = CInt(SepararDatos(ind +
2))) Then
7479                 For ind5 = 0 To SepararDatos.LongCount - 1
7480                     EscribirEnlacesAnteriores(filaEtap0, SepararDatos, ind2, ind5, MúltiplesEnlacesAnteriores,
MúltiplesEtapasAnteriores)
7481                     If SepararDatos(ind5) = "LineaHorizontal" Then
7482                         'Línea horizontal unida con línea vertical
7483                         If ((CInt(SepararDatos(ind5 + 2)) >= CInt(SepararDatos(ind2 + 2))) And (CInt(SepararDatos(ind5 +
2)) < CInt(SepararDatos(ind2 + 2)) + CInt(SepararDatos(ind2 + 3))) And (CInt(SepararDatos(ind5 + 1)) = CInt(SepararDatos(ind2
+ 1)) + 1) Or ((CInt(SepararDatos(ind5 + 2)) >= CInt(SepararDatos(ind2 + 2))) And (CInt(SepararDatos(ind5 + 2)) <
CInt(SepararDatos(ind2 + 2)) + CInt(SepararDatos(ind2 + 3))) And (CInt(SepararDatos(ind5 + 1)) + CInt(SepararDatos(ind5 + 3))
= CInt(SepararDatos(ind2 + 1))) Or ((CInt(SepararDatos(ind2 + 1)) >= CInt(SepararDatos(ind5 + 1))) And
(CInt(SepararDatos(ind2 + 1)) < CInt(SepararDatos(ind5 + 1)) + CInt(SepararDatos(ind5 + 3))) And (CInt(SepararDatos(ind5 + 2))
= CInt(SepararDatos(ind2 + 2)) - 1)) Then
7484                             BucleLineaHorizontalMasLineaVertical(filaEtap0, SepararDatos, ind5,
MúltiplesEnlacesSiguientes, MúltiplesEnlacesAnteriores, MúltiplesEtapasAnteriores)
7485                             End If
7486                         End If
7487                     Next ind5
7488                 End If
7489             End If
7490         Next ind2

```

Figura 3.7.1.2.15.5. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Si la línia vertical està immediatament a sota i immediatament a sota de la línia vertical n'hi ha un enllaç, es truca a la funció EscribirEnlacesSiguientes. Un fragment del codi és el següent:

```

6548 'función que busca los enlaces siguientes a una etapa según la posición de los elementos:
6549 Private Sub EscribirEnlacesSiguientes(filaEtap0() As String, SepararDatos() As String, ind2 As Integer, ind5 As Integer,
ByRef MúltiplesEnlacesSiguientes As Boolean)
6550     If SepararDatos(ind5) = "Transición" Then
6551         If (CInt(SepararDatos(ind5 + 2)) = CInt(SepararDatos(ind2 + 2)) + CInt(SepararDatos(ind2 + 3))) And
(CInt(SepararDatos(ind5 + 1)) = CInt(SepararDatos(ind2 + 1)) - 15) Then
6552             If MúltiplesEnlacesSiguientes = False Then
6553                 filaEtap0(6) = "Trans.Nivel: " + SepararDatos(ind5 + 3) + ", "
6554             Else
6555                 MúltiplesEnlacesSiguientes = True
6556                 filaEtap0(6) &= SepararDatos(ind5 + 3) + ", "
6557             End If
6558         End If
6559     End If

```

Figura 3.7.1.2.15.6. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

En aquesta funció únicament es mira si la línia vertical està enllaçada amb la transició, convergència, transició del temporitzador, etc, per poder escriure l'enllaç a la columna adient.

Si la línia vertical està immediatament a munt de l'etapa es truca a la funció EscribirEnlacesAnteriores que funciona de la mateixa manera que la figura 3.7.1.2.15.6. però amb la columna corresponent i trucant a la funció descrita a la figura 3.7.1.2.15.4. amb la recerca de les etapes anteriors.

En el cas que es produeixi una unió entre la línia vertical i una horitzontal caldrà recórrer de nou un bucle (línies 7472 i 7484 de la figura 3.7.1.2.15.5). Dintre d'aquest nou bucle es tornarà a buscar enllaços a sobre i a sota de la línia horitzontal i en cas que es trobi amb una línia vertical caldrà fer un segon bucle. No es creu necessari detallar més sobre el funcionament d'aquests bucles ja que simplement provenen que l'enllaç sigui correcte i sinó no es crea l'enllaç.

Les convergències i les divergències en "Y" tenen una particularitat i per això dintre dels bucles anteriors quan hi ha enllaç truquen a les seves funcions determinades. Centrant-se amb les convergències la funció és EscribirEtapasConvergencia i el codi que es deriva del qual és aquest:

```

6682 'Función que busca los enlaces probables con una etapa al haber una convergencia según la posición de los elementos:
6683 Private Sub EscribirEtapasConvergencia(filaEtapa0() As String, SepararDatos() As String, ind2 As Integer)
6684     If filaEtapa0(8) = "" Then
6685         filaEtapa0(8) = "MúltiplesEtapas: "
6686     End If
6687     For ind3 = 0 To SepararDatos.LongCount - 1
6688         If SepararDatos(ind3) = "LineaVertical" Then
6689             If (CInt(SepararDatos(ind3 + 2)) + CInt(SepararDatos(ind3 + 3)) = CInt(SepararDatos(ind2 + 2))) And
6690             (CInt(SepararDatos(ind3 + 1)) >= CInt(SepararDatos(ind2 + 1))) And (CInt(SepararDatos(ind3 + 1)) < CInt(SepararDatos(ind2 +
6691             1)) + 330) Then
6692                 For ind4 = 0 To SepararDatos.LongCount - 1
6693                     If SepararDatos(ind4) = "Etapa" Then
6694                         If (CInt(SepararDatos(ind3 + 1)) >= CInt(SepararDatos(ind4 + 1))) And (CInt(SepararDatos(ind3 +
6695                         1)) < CInt(SepararDatos(ind4 + 1)) + 30) And (CInt(SepararDatos(ind3 + 2)) = CInt(SepararDatos(ind4 + 2)) + 30) Then
6696                             filaEtapa0(8) &= SepararDatos(ind4 + 3) + ", "
6697                         End If
6698                     If SepararDatos(ind4) = "Etapa0" Then
6699                         If (CInt(SepararDatos(ind3 + 1)) >= CInt(SepararDatos(ind4 + 1))) And (CInt(SepararDatos(ind3 +
6700                         1)) < CInt(SepararDatos(ind4 + 1)) + 30) And (CInt(SepararDatos(ind3 + 2)) = CInt(SepararDatos(ind4 + 2)) + 30) Then
6701                             filaEtapa0(8) &= "0, "
6702                         End If
6703                     End If
6704                 Next ind4
6705             End If
6706         Next ind3
6707     End Sub

```

Figura 3.7.1.2.15.7. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Únicament cal destacar que es torna a fer un bucle cercant les línies verticals i les etapes que puguin estar sobre la convergència per crear l'enllaç.

En el cas de les divergències la funció s'anomena BusquedaEnlaceEtapaDivergencia i un fragment del codi es mostra a continuació.

```

6215 Private Sub BusquedaEnlaceEtapaDivergencia(filaEtapa0() As String, SepararDatos() As String, ind2 As Integer, ByRef
6216     MúltiplesEtapasAnteriores As Boolean)
6217     For ind3 = 0 To SepararDatos.LongCount - 1
6218         If SepararDatos(ind3) = "Etapa" Then
6219             If (CInt(SepararDatos(ind2 + 1)) >= CInt(SepararDatos(ind3 + 1)) + 135) And (CInt(SepararDatos(ind3 + 1)) <
6220             CInt(SepararDatos(ind2 + 1)) + 195) And (CInt(SepararDatos(ind2 + 2)) = CInt(SepararDatos(ind3 + 2)) + 30) Then
6221                 If MúltiplesEtapasAnteriores = False Then
6222                     filaEtapa0(8) = "Etapa: " + SepararDatos(ind3 + 3) + ", "
6223                     MúltiplesEtapasAnteriores = True
6224                 Else
6225                     filaEtapa0(8) &= SepararDatos(ind3 + 3) + ", "
6226                 End If
6227             End If
6228             If SepararDatos(ind3) = "Etapa0" Then
6229                 If (CInt(SepararDatos(ind2 + 1)) >= CInt(SepararDatos(ind3 + 1)) + 135) And (CInt(SepararDatos(ind3 + 1)) <
6230                 CInt(SepararDatos(ind2 + 1)) + 195) And (CInt(SepararDatos(ind2 + 2)) = CInt(SepararDatos(ind3 + 2)) + 30) Then
6231                     If MúltiplesEtapasAnteriores = False Then
6232                         filaEtapa0(8) = "Etapa: 0, "
6233                         MúltiplesEtapasAnteriores = True
6234                     Else
6235                         filaEtapa0(8) &= "0, "
6236                     End If
6237                 End If
6238             End If
6239         Next ind3
6240     End Sub

```

Figura 3.7.1.2.15.8. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Com en el cas de les convergències es cerquen les etapes que poden estar enllaçades amb la divergència i si n'hi ha línies verticals o horitzontals es fan els bucles adients per trobar els enllaços.

Recapitulant i continuant amb la funció BusquedaEnlaceEtapa, un cop es tenen tots els enllaços el codi que s'executa és el següent:

```

7491 Dim numeroComasEnlaceSiguiente As Integer = 0
7492 Dim numeroComasEnlaceAnterior As Integer = 0
7493 Dim numeroComasEtapaAnterior As Integer = 0
7494 For Each caracter As Char In filaEtapa0(6)
7495     If caracter = "," Then
7496         numeroComasEnlaceSiguiente += 1
7497     End If
7498 Next caracter
7499 For Each caracter As Char In filaEtapa0(7)
7500     If caracter = "," Then
7501         numeroComasEnlaceAnterior += 1
7502     End If
7503 Next caracter
7504 For Each caracter As Char In filaEtapa0(8)
7505     If caracter = "," Then
7506         numeroComasEtapaAnterior += 1
7507     End If
7508 Next caracter
7509 If numeroComasEnlaceSiguiente > 1 Then
7510     MultiplesEnlaces(filaEtapa0, 6, "MúltiplesEnlaces: ")
7511 End If
7512 If numeroComasEnlaceAnterior > 1 Then
7513     MultiplesEnlaces(filaEtapa0, 7, "MúltiplesEnlaces: ")
7514 End If
7515 If numeroComasEtapaAnterior > 1 Then
7516     MultiplesEnlaces(filaEtapa0, 8, "MúltiplesEtapas: ")
7517 End If
7518 filaEtapa0(6) = filaEtapa0(6).TrimEnd(", ", " ")
7519 filaEtapa0(7) = filaEtapa0(7).TrimEnd(", ", " ")
7520 filaEtapa0(8) = filaEtapa0(8).TrimEnd(", ", " ")
7521 If filaEtapa0(2) = "" Then
7522     filaEtapa0(2) = "No hay acción asociada"
7523 End If
7524 If filaEtapa0(3) = "" Then
7525     filaEtapa0(3) = "No hay acción asociada"
7526 End If
7527 If filaEtapa0(4) = "" Then
7528     filaEtapa0(4) = "No hay timer asociado"
7529 End If
7530 If filaEtapa0(5) = "" Then
7531     filaEtapa0(5) = "No hay timer asociado"
7532 End If
7533 If filaEtapa0(6) = "" Then
7534     filaEtapa0(6) = "No hay transición asociada"

```

Figura 3.7.1.2.15.9. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

En aquest instant es generen noves variables que permeten comptar les comes i modificar els textos per a que siguin múltiples enllaços o etapes fent ús de la funció MultiplesEnlaces. Després utilitzant el TrimEnd s'eliminen les comes finals que es creen cada cop que es troba un enllaç i sinó s'ha trobat cap enllaç en alguna de les columnes es determina que no hi ha enllaç d'aquell tipus. La funció MultiplesEnlaces consisteix en eliminar els elements que es poden haver duplicat. El codi corresponent aquest procediment s'observa a continuació:

```

7547 'Función para escribir múltiples enlaces y eliminar los repetidos
7548 Private Sub MultiplesEnlaces(filaEtapa0() As String, num As Integer, tipo As String)
7549     Dim Espacio = InStr(filaEtapa0(num), " ")
7550     Dim Recorte = filaEtapa0(num).Substring(0, Espacio - 1)
7551     Dim enlaces = filaEtapa0(num).Substring(Espacio, Len(filaEtapa0(num)) - Espacio)
7552     filaEtapa0(num) = filaEtapa0(num).Replace(enlaces, "")
7553     enlaces = enlaces.Replace(" ", "")
7554     Dim enlacesSiguietes() = enlaces.Split(",")
7555     Dim enlacesNoRepetidos = enlacesSiguietes.Distinct().ToArray()
7556     Dim enlacesFinales As New List(Of String)
7557     EliminarRepeticiones(enlacesNoRepetidos, enlacesFinales, "")
7558     filaEtapa0(num) = filaEtapa0(num).Replace(Recorte, tipo)
7559     For ind = 0 To enlacesFinales.LongCount - 1
7560         filaEtapa0(num) = filaEtapa0(num) + enlacesFinales(ind) + ", "
7561     Next ind
7562 End Sub

```

Figura 3.7.1.2.15.10. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Aquesta funció agafa com a paràmetres la fila d'enllaços, un número relacionat amb la columna i una cadena que utilitzarà per definir el nou de múltiples enllaços o etapes. El procediment consisteix en trobar l'espai dintre de la cadena de la columna de la llista definida pel número, retallar la part anterior a l'espai i substituir-la per la cadena nova. La part no retallada és on es troben els enllaços els quals estan separats per comes (","). Es separen els enllaços i s'eliminen els repetits utilitzant la funció Split i Distinct.ToArray. Finalment trucant a la funció EliminarRepeticiones, s'ordenen sobre una nova llista de cadenes els enllaços que després es monten de forma adient sobre la cadena retallada amb el nou títol. El procediment d'EliminarRepeticiones es mostra a continuació:

```

8091      'Función que elimina las entradas repetidas:
8092      Private Sub EliminarRepeticiones(sender As String(), e As List(Of String), punto As String)
8093          Dim NumeroGrupoEliminar As New List(Of Integer)
8094          Dim encontrado As Boolean = False
8095          For ind = 0 To sender.LongCount - 1 'Recorre los elementos de la matriz de cadenas que le llega a la función
8096              For Each caracter As Char In sender(ind) 'Recorre cada caracter
8097                  IF (caracter = punto Or caracter = " " Or caracter = "") And encontrado = False Then 'Si se encuentra un . un
espacio o un elemento vacío se borra
8098                      NumeroGrupoEliminar.Add(ind) 'Se añade el numero que se debe eliminar
8099                      encontrado = True 'Se ha encontrado elemento a borrar el booleano es verdadero
8100                  End If
8101              Next caracter
8102              If sender(ind) = "" Then 'Si directamente está vacío se borra
8103                  NumeroGrupoEliminar.Add(ind) 'Se añade el numero que se debe eliminar
8104              End If
8105              encontrado = False 'Si no se encuentra nada el booleano es falso
8106          Next ind
8107
8108          For ind = 0 To sender.LongCount - 1
8109              e.Add(sender(ind)) 'Se guarda en la nueva lista todo el contenido viejo
8110          Next ind
8111
8112          'Se ordenan los números de mayor a menor, podría recorrerse la lista de números de mayor a menor también
8113          NumeroGrupoEliminar.Sort()
8114          NumeroGrupoEliminar.Reverse()
8115
8116          For ind = 0 To NumeroGrupoEliminar.LongCount - 1
8117              e.RemoveAt(NumeroGrupoEliminar(ind)) 'Se elimina en la nueva lista los elementos encontrados anteriormente
8118          Next ind
8119      End Sub

```

**Figura 3.7.1.2.15.11.** Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

El seu funcionament és el següent, es recorre la matriu en base 0 que li arriba i es recorre cada caracter i s'escriu sobre una llista d'enters l'índex en cas que s'hagi trobat un espai o una cadena buida o algun caracter especial com podria ser el punt. Si la cadena està buida s'esborra directament. Si no es troba res la variable booleana es falsa. Després es copia el contingut de la matriu a la nova llista i s'esborren els elements on hi apareixia una cadena buida o un espai.

D'aquesta manera es finalitza l'explicació de la cerca d'enllaços, cal dir que el sistema és rudimentari i requereix de molts bucles per trobar totes les connexions entre els elements. Tot i així, a la fi el resultat és adequat i s'obtenen els enllaços necessaris.

Dintre de la pestanya de Ver també està la casella de Ocultar Marcas. A continuació s'analitza el codi realitzat:

```

8144      'Función para ver u ocultar marcas
8145      Private Sub MarcasToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles MarcasToolStripMenuItem.Click
8146          Try
8147              If LabelMarca0.Visible = True Then 'Si la marca de la etapa0 es visible
8148                  LabelMarca0.Visible = False 'Se vuelve invisible la marca0
8149                  For ind = 0 To listaMarcaLabsEtapa.LongCount - 1
8150                      listaMarcaLabsEtapa(ind).Visible = False 'Se vuelven invisibles todas las marcas de las etapas
8151                  Next ind
8152                  MarcasToolStripMenuItem.Text = "Mostrar marcas" 'Se cambia el texto de ocultar a mostrar
8153              Else
8154                  LabelMarca0.Visible = True 'Se vuelve visible la marca0
8155                  For ind = 0 To listaMarcaLabsEtapa.LongCount - 1
8156                      listaMarcaLabsEtapa(ind).Visible = True 'Se vuelven visibles todas las marcas de las etapas
8157                  Next ind

```

```

8158         MarcasToolStripMenuItem.Text = "Ocultar marcas" 'Se cambia el texto de mostrar a ocultar
8159     End If
8160     Catch
8161     MsgBox("No hay etapas creadas o no existe etapa inicial", MessageBoxIcon.Error, "Error") 'Si hay error se comunica
8162     End Try
8163 End Sub
    
```

**Figura 3.7.1.2.15.12.** Captures d’imatge part del codi annex A1. (Font: Pablo Mañas)

En fer clic es realitza un try-catch per si no existeix cap etapa incloent l’etapa inicial o únicament l’etapa inicial. En cas que les marques siguin visibles (línia 8147) es tornen invisibles modificant la propietat Visible. L’etiqueta de l’etapa inicial es torna invisible de manera individual i després es recorre la llista on es troben les etiquetes de les etapes per anar canviant la propietat en cada marca. També es canvia el text del botó que canvia de “Ocultar Marcas” a “Mostrar Marcas”. Si les marques són invisibles al clicar es tornen visibles de la mateixa manera que abans, modificant la propietat Visible a true. El text es torna a canviar a com estava a l’inici.

### 3.7.1.2.16 Procediment ActualizarEnlacesManuales

Tot i que s’ha mencionat al subapartat anterior, no s’ha acabat d’explicar com es creen els enllaços manuals. Per tant, ara es mostra el codi referent a aquesta part:

```

7630 'Función que apartir de los datos manuales introducidos con la función correspondiente se sobrescribe en la tabla de
7631 enlaces (se pierden datos anteriores).
7632 Private Sub ActualizarEnlacesManuales()
7633 Dim SepararEnlaces() As String
7634 If LineaForm.GuardarEnlaces <> "" Then
7635     SepararEnlaces = LineaForm.GuardarEnlaces.Split("|")
7636     For ind = 0 To SepararEnlaces.LongCount - 1
7637         If SepararEnlaces(ind) = "anterior" Then
7638             Try
7639                 EnlacesForm.DataGridView1.Rows(SepararEnlaces(ind + 1)).Cells(7).Value = SepararEnlaces(ind + 2)
7640             Catch
7641             End Try
7642         End If
7643         If SepararEnlaces(ind) = "siguiente" Then
7644             Try
7645                 EnlacesForm.DataGridView1.Rows(SepararEnlaces(ind + 1)).Cells(6).Value = SepararEnlaces(ind + 2)
7646             Catch
7647             End Try
7648         End If
7649     Next ind
7650 End Sub
    
```

**Figura 3.7.1.2.16.1.** Captura d’imatge part del codi annex A1. (Font: Pablo Mañas)

Com es pot veure a la figura anterior s’utilitza una variable de la classe LineaForm, en concret GuardarEnlaces, que conté els enllaços que s’han creat de forma manual. Havent separat prèviament aquestes dades és recorre la matriu i si es troba l’etiqueta anterior referent a què es tracta d’un enllaç anterior, es situa a la filera de la taula guardada a una posició més de l’etiqueta identificadora y s’estableix l’enllaç manual guardat a la segona posició a partir d’aquesta mateixa etiqueta. El mateix succeeix si l’enllaç és següent però amb la diferència que canvia la columna de la taula. Per si es produeix algun error es troba dins un try-catch que si falla no fa res.

### 3.7.1.2.17 Procediments pestanya Simular

Un dels ítems de la barra de menús és el de Simular. En clicar s'obren usualment el botó de Simular habilitat i el de Detener inhabilitat. El primer botó consisteix en crear un nou panell on es fa la simulació i en generar una taula d'entrades, sortides i temporitzadors. Així doncs, es dividirà l'explicació en diferents parts. El codi de la primer part és el següent:

```

7672 Private Sub SimularToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles SimularToolStripMenuItem.Click
7673     SimularToolStripMenuItem.Enabled = False 'Se deshabilita el botón de simular
7674     DetenerToolStripMenuItem.Enabled = True 'Se habilita el botón de detener
7675     Dim Simulacion As New Form 'Se crea una nueva ventana
7676     Dim PanelSimulacion As New Panel 'Se crea un nuevo panel
7677     Dim DatosActuales As String 'Se definen datos actuales
7678     Dim SepararDatosActuales() As String 'Define variable que separa datos actuales
7679     DatosActuales = ""
7680     EscribirArchivoguardado(DatosActuales) 'Se escriben los datos en la variable
7681     ActualizarTablaEnlaces(DatosActuales) 'Se actualizan los datos de la tabla de enlaces
7682     'Se crea la ventana de simulación en función del tamaño del panel:
7683     If ChildForm.Width = 611 Then
7684         Simulacion.Width = 611
7685         Simulacion.Height = 881
7686     ElseIf ChildForm.Width = 858 Then
7687         Simulacion.Width = 858
7688         Simulacion.Height = 634
7689     ElseIf ChildForm.Width = 1287 Then
7690         Simulacion.Width = 1287
7691         Simulacion.Height = 881
7692     Else
7693         Simulacion.Width = 842
7694         Simulacion.Height = 1191
7695     End If
7696     Simulacion.MdiParent = Me 'El padre de la simulación es la ventana principal
7697     Simulacion.Text = "Simulación" 'Se define el título de la simulación en Simulación
7698     Simulacion.BackColor = Color.Gray 'Fondo gris
7699     Simulacion.AutoScroll = True 'Desplazamiento automático activado
7700     Simulacion.MaximizeBox = False 'Maximización desactivada
7701     Simulacion.MinimizeBox = False 'Minimización desactivada
7702     Simulacion.FormBorderStyle = FormBorderStyle.FixedSingle 'Borde fijo
7703     AddHandler Simulacion.FormClosing, AddressOf SimulacionClose 'Se añade la función de cierre de simulación
7704     Simulacion.Show() 'Se muestra la ventana de simulación
7705     EnlacesForm.Show() 'Se muestran los enlaces si no lo están ya
7706     EnlacesForm.Top = MenuStrip1.Top + MenuStrip1.Height() + 40 'Se sitúa la ventana en cierta posición del programa
7707     EnlacesForm.Left = Splitter2.Left - EnlacesForm.Width() - 10
7708     PanelSimulacion.Width = Panel1.Width 'Se definen las variables del nuevo panel de simulación
7709     PanelSimulacion.Height = Panel1.Height
7710     PanelSimulacion.BackColor = Color.White
7711     PanelSimulacion.Top = 0
7712     PanelSimulacion.Left = 0
7713     Simulacion.Controls.Add(PanelSimulacion)

```

Figura 3.7.1.2.17.1. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

En el moment en que es fa clic, s'inhabilita el botó de simular i s'habilita el de detenir. Després es crea un nou panell amb una nova finestra que guarden les mateixes característiques que la finestra filla creada quan es genera un arxiu. Se li afegeix una funció de tancament per què al tancar-se la finestra es realitzin certes coses. Més endavant es mostra la finestra de simulació, la finestra d'enllaços que es situa en cert lloc. Per últim es situa de forma adequada el panell de simulació a la finestra de simulació.

La segona part consisteix en observar com es copien les dades del grafet creat al panell de simulació. La part del codi corresponent és la següent:

```

7714     SepararDatosActuales = DatosActuales.Split("|") 'Se obtienen los datos actuales fragmentados, en función de estos se
7715     crea la simulación como copia de los datos actuales en el panel del grafet:
7716     For ind = 0 To SepararDatosActuales.LongCount - 1
7717         If SepararDatosActuales(ind) = "Etapa0" Then

```

```

7717         Dim etapa0simulacion As New PictureBox With {.Image = Image.FromFile("Imágenes\Etapa0.jpg"), .SizeMode =
PictureBoxSizeMode.StretchImage, .Width = 30, .Height = 30, .Left = Cint(SepararDatosActuales(ind + 1)), .Top =
CInt(SepararDatosActuales(ind + 2)), .Tag = True}
7718     PanelSimulacion.Controls.Add(etapa0simulacion)
7719     listaEtapasSimulacion.Add(etapa0simulacion)
7720     Dim Label0simulacion As New Label With {.AutoSize = True, .Text = "0", .Left = etapa0simulacion.Left +
(etapa0simulacion.Width() / 3), .Top = etapa0simulacion.Top + (etapa0simulacion.Height() / 3), .BackColor = Color.Transparent}
7721     Label0simulacion.Font = New Font(Label0simulacion.Font.Name, 8)
7722     PanelSimulacion.Controls.Add(Label0simulacion)
7723     Label0simulacion.BringToFront()
7724     Dim LabelMarca0simulacion As New Label With {.AutoSize = True, .Text = "M0.0", .Left = etapa0simulacion.Left -
1.2 * etapa0simulacion.Width(), .Top = etapa0simulacion.Top + (etapa0simulacion.Height() / 3), .BackColor = Color.Transparent,
.Visible = False}
7725     LabelMarca0simulacion.Font = New Font(LabelMarca0simulacion.Font.Name, 7)
7726     PanelSimulacion.Controls.Add(LabelMarca0simulacion)
7727     End If
7728     If SepararDatosActuales(ind) = "Etapa" Then
7729         Dim etapaSimulacion As New PictureBox With {.Image = Image.FromFile("Imágenes\Etapa.png"), .SizeMode =
PictureBoxSizeMode.StretchImage, .Width = 30, .Height = 30, .Left = Cint(SepararDatosActuales(ind + 1)), .Top =
CInt(SepararDatosActuales(ind + 2)), .Tag = False}
7730     PanelSimulacion.Controls.Add(etapaSimulacion)
7731     listaEtapasSimulacion.Add(etapaSimulacion)
7732     Dim LabelEtapaSimulacion As New Label With {.AutoSize = True, .Text = SepararDatosActuales(ind + 3), .Left =
etapaSimulacion.Left + (etapaSimulacion.Width() / 3), .Top = etapaSimulacion.Top + (etapaSimulacion.Height() / 3), .BackColor
= Color.Transparent, .Width = 20, .Height = 13}
7733     PanelSimulacion.Controls.Add(LabelEtapaSimulacion)
7734     LabelEtapaSimulacion.BringToFront()
7735     Dim LabelMarcaEtapaSimulacion As New Label With {.AutoSize = True, .Text = SepararDatosActuales(ind + 4),
.Left = etapaSimulacion.Left - 1.3 * etapaSimulacion.Width(), .Top = etapaSimulacion.Top + (etapaSimulacion.Height() / 3),
.BackColor = Color.Transparent, .Visible = False}
7736     LabelMarcaEtapaSimulacion.Font = New Font(LabelMarcaEtapaSimulacion.Font.Name, 7)
7737     LabelMarcaEtapaSimulacion.Cursor = Cursors.SizeAll
7738     PanelSimulacion.Controls.Add(LabelMarcaEtapaSimulacion)
7739     End If
7740     If SepararDatosActuales(ind) = "Transicion" Then
7741         Dim transicionSimulacion As New PictureBox
7742         Dim LabelTransSimulacion As New Label
7743         CrearTransicionSimulacion(transicionSimulacion, LabelTransSimulacion, SepararDatosActuales, ind,
PanelSimulacion)
7744     listaLabsSimulacion.Add(LabelTransSimulacion)
7745     End If

```

Figura 3.7.1.2.17.2. Captura d’imatge part del codi annex A1. (Font: Pablo Mañas)

Fent ús de les dades guardades es fa un recorregut i es van creant noves etapes, transicions i tota sèrie d’objectes que s’afegeixen al nou panell de simulació i a les variables de les llistes corresponents definides per a la simulació. D’aquesta manera s’aconsegueix tenir una copia sencera del panell inicial en el panell simulat però sense modificar les variables de l’original i fent servir les de la simulació. Potser és necessari aclarar la funció CrearTransicionSimulacion ja que només s’utilitza dintre d’aquest procediment del qual s’està parlant. Aquesta funció únicament utilitza les variables per generar les transicions i les etiquetes associades amb les característiques adequades i afegir-les al panell. Es va crear per reduir línies de codi, si es vol veure en detall comença a la línia 7653 del codi. Cal dir que la figura anterior està retallada i faltaria la resta d’objectes, però com el funcionament és el mateix no cal especificar-lo.

La tercera part que cal explicar és la formació de la taula d’entrades, sortides i temporitzadors. El codi d’aquesta part és el següent:

```

7902     'Se definen variables con los enlaces de la tabla de enlaces para crear la tabla de entradas/salidas y temporizadores.
7903     Dim Enlaces As String
7904     Dim EnlacesCortados As String
7905     Dim ReconstruccionEnlaces() As String
7906     Dim ReconstruccionEnlaces2() As String
7907     Dim Espacio As Integer
7908     Dim Salidas As String
7909     Dim SalidasCortadas As String
7910     Dim ReconstruccionSalidas() As String
7911     Dim Temporizadores As String
7912     Dim TemporizadoresCortados As String
7913     Dim ReconstruccionTemporizadores() As String
7914     ReconstruccionEnlaces = Nothing
7915     ReconstruccionSalidas = Nothing
7916     ReconstruccionTemporizadores = Nothing
7917     Enlaces = ""
7918     EnlacesCortados = ""
7919     Salidas = ""
7920     SalidasCortadas = ""
7921     Temporizadores = ""
7922     TemporizadoresCortados = ""

```

Figura 3.7.1.2.17.3. Captura d’imatge part del codi annex A1. (Font: Pablo Mañas)

Primerament, es defineixen una sèrie de variables que es necessiten per transformar els enllaços creats a la taula en entrades, sortides i temporitzadors. Després es fa un recorregut sobre la taula d'enllaços:

```

7924 'Se recorre la tabla de enlaces
7925 For Each Fila As DataGridViewRow In EnlacesForm.DataGridView1.Rows
7926 For Columna = 6 To 7 'Se recorren todos los enlaces (anteriores y siguientes)
7927 Enlaces = Fila.Cells(Columna).Value.ToString + "+" 'Se guarda en una variable el valor y se le añade un + como
separación
7928 If Enlaces <> "haytransiciónasociada+" And Enlaces <> "No hay transición asociada+" Then
7929 Espacio = InStr(Enlaces, " ") 'Encuentra el espacio
7930 Enlaces = Enlaces.Substring(Espacio, Len(Enlaces) - Espacio) 'Separa los datos posteriores al espacio
7931 Enlaces = Enlaces.Replace(",", "+") 'Reemplaza las comas por símbolos + en caso que haya múltiples enlaces
7932 Enlaces = Enlaces.Replace(" ", "") 'Elimina los espacios
7933 EnlacesCortados += Enlaces 'Va añadiendo los enlaces a otra variable
7934 ReconstruccionEnlaces = EnlacesCortados.Split("+") 'Se separan todos los enlaces
7935 If ReconstruccionEnlaces(0) = EnlacesCortados Then 'Si solo hay un enlace con un AND
7936 ReconstruccionEnlaces = EnlacesCortados.Split(".") 'Se separan los puntos, para obtener las entradas
(por ejemplo: FC1-FC2 -> FC1 (1 entrada) FC2 (2 entrada))
7937 Else 'Si hay más
7938 For ind = 0 To ReconstruccionEnlaces.LongCount - 1 'Se recorren todos
7939 For Each caracter As Char In ReconstruccionEnlaces(ind) 'Se busca caracter por caracter los ., ya
que los + ya se han separado
7940 If caracter = "." Then
7941 ReconstruccionEnlaces2 = ReconstruccionEnlaces(ind).Split(".") 'Se define una lista con
los enlaces reconstruidos
7942 Try
7943 ReDim Preserve ReconstruccionEnlaces(ReconstruccionEnlaces.LongCount +
ReconstruccionEnlaces2.LongCount - 1) 'Se redefine la dimensión de la lista
7944 Catch
7945 ReDim Preserve ReconstruccionEnlaces(ReconstruccionEnlaces.LongCount - 1) 'Se redefine
la dimensión de la lista
7946 End Try
7947 Try
7948 Array.Copy(ReconstruccionEnlaces2, 0, ReconstruccionEnlaces,
ReconstruccionEnlaces.LongCount - ReconstruccionEnlaces2.LongCount, ReconstruccionEnlaces2.LongCount) 'Se copia en la segunda
matriz el contenido de la primera
7949 Catch
7950 End Try
7951 End If
7952 Next caracter
7953 Next ind
7954 End If
7955 End If
7956 Next Columna
7957 For Columna = 2 To 3 'Se recorren las acciones
7958 Salidas = Fila.Cells(Columna).Value.ToString
7959 If Salidas <> "No hay acción asociada" Then
7960 Espacio = InStr(Salidas, " ")
7961 Salidas = Salidas.Substring(Espacio, Len(Salidas) - Espacio)
7962 SalidasCortadas += Salidas + "|"
7963 ReconstruccionSalidas = SalidasCortadas.Split("|") 'Se obtienen todas las salidas separadas
7964 End If
7965 Next Columna
7966 For Columna = 4 To 5 'Se recorren los temporizadores
7967 Temporizadores = Fila.Cells(Columna).Value.ToString
7968 If Temporizadores <> "No hay timer asociado" Then
7969 Espacio = InStr(Temporizadores, " ")
7970 Temporizadores = Temporizadores.Substring(Espacio, Len(Temporizadores) - Espacio)
7971 TemporizadoresCortados += Temporizadores + "|"
7972 ReconstruccionTemporizadores = TemporizadoresCortados.Split("|") 'Se obtienen todos los temporizadores
7973 End If
7974 Next Columna
7975 Next Fila

```

Figura 3.7.1.2.17.4. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Es recorren totes les files de les columnes 6 i 7 i es guarden els enllaços que apareixen en la variable Enlaces on s'eliminen els espais i es reemplacen les comes per símbols "+" per poder fer la partició i es van afegint sobre la variable EnlacesCortados. Els enllaços són separats en una matriu de base 0 i si només hi apareix un enllaç AND es separen de nou amb el símbol ".". En cas que n'hi hagi més d'un enllaç AND s'han de recórrer i afegint aquells que contenen el caràcter "." sobre una nova variable. Després es tracta de redimensionar la matriu de la llista amb la nova mida (línia 7943) i sinó es possible es manté amb la mateixa mida (línia 7945). Després es copia el contingut de la primera matriu a la segona matriu la qual tindrà els enllaços ja ben formats. Per les accions i els temporitzadors es recorren les columnes 2 i 3 i 3 i 4 respectivament. En aquest cas és més fàcil ja que no intervé cap operador



lògic (/,+ o ·). Finalment s’obté en les diferents variables unes matrius amb tots els enllaços, altra amb totes les accions i una amb els temporitzadors. És molt probable que n’hi hagin enllaços i accions repetides, per tant, cal eliminar les repeticions aparegudes. Així doncs, el codi continua de la següent forma:

```

7977 'Se definen nuevas variables para construir la tabla de entradas/salidas y temporizadores
7978 Dim EliminarEntradasRepetidas() As String
7979 Dim EnlacesFinales As New List(Of String)
7980 Dim EliminarSalidasRepetidas() As String
7981 Dim SalidasFinales As New List(Of String)
7982 Dim EliminarTemporizadoresRepetidos() As String
7983 Dim TemporizadoresFinales As New List(Of String)
7984
7985 If ReconstruccionEnlaces Is Nothing Then
7986     MessageBox.Show("No hay enlaces") 'En caso que no haya enlaces se muestra
7987 Else
7988     For ind = 0 To ReconstruccionEnlaces.LongCount - 1
7989
7990         For Each caracter As Char In ReconstruccionEnlaces(ind)
7991             If caracter = "(" Then
7992                 ReconstruccionEnlaces(ind) = ReconstruccionEnlaces(ind).TrimStart("(") 'Se eliminan los paréntesis
7993             End If
7994             If caracter = "/" Then
7995                 ReconstruccionEnlaces(ind) = ReconstruccionEnlaces(ind).TrimStart("/") 'Se eliminan las barras /
7996             End If
7997             If caracter = ")" Then
7998                 ReconstruccionEnlaces(ind) = ReconstruccionEnlaces(ind).TrimEnd(")") 'Se eliminan los paréntesis
7999             End If
8000         Next caracter
8001     Next ind
8002     EliminarEntradasRepetidas = ReconstruccionEnlaces.Distinct().ToArray() 'Se eliminan las entradas repetidas
8003     EliminarRepeticiones(EliminarEntradasRepetidas, EnlacesFinales, ".")
8004 End If
8005
8006 'Se repite para las salidas y para los temporizadores:
8007 If ReconstruccionSalidas Is Nothing Then
8008     MessageBox.Show("No hay acciones")
8009 Else
8010     EliminarSalidasRepetidas = ReconstruccionSalidas.Distinct().ToArray()
8011     EliminarRepeticiones(EliminarSalidasRepetidas, SalidasFinales, ".")
8012 End If
8013
8014 If ReconstruccionTemporizadores Is Nothing Then
8015     MessageBox.Show("No hay temporizadores")
8016 Else
8017     For ind = 0 To ReconstruccionTemporizadores.LongCount - 1
8018         ReconstruccionTemporizadores(ind) = ReconstruccionTemporizadores(ind).Replace(" ", "")
8019     Next ind
8020     EliminarTemporizadoresRepetidos = ReconstruccionTemporizadores.Distinct().ToArray()
8021     EliminarRepeticiones(EliminarTemporizadoresRepetidos, TemporizadoresFinales, ".")
8022 End If

```

Figura 3.7.1.2.17.5. Captura d’imatge part del codi annex A1. (Font: Pablo Mañas)

Es defineixen noves variables per esborrar entrades, sortides o temporitzadors repetits. En aquest cas si no hi ha enllaços o accions o temporitzadors es mostra un missatge d’advertència (línies 7986, 8007 i 8014). En el cas dels enllaços s’eliminen els parèntesis i les barres de negació amb la fi de deixar les entrades netes. Després en cada un dels casos s’eliminen les repeticions amb la funció ja comentada anteriorment EliminarRepeticiones i es guarden en aquestes noves variables. Ara ja serà el moment de crear la taula d’entrades, sortides i temporitzadors:

```

8023 'Se definen nuevas variables con el numero del temporizador y los temporizadores finalmente
8024 Dim TimersFinales As New List(Of String)
8025 Dim numeroTimers As New List(Of Integer)
8026 Dim numeroNoTimers As New List(Of Integer)
8027
8028 For ind = 0 To EnlacesFinales.LongCount - 1
8029     TimersFinales.Add(EnlacesFinales(ind))
8030 Next ind
8031
8032 For ind = 0 To TimersFinales.LongCount - 1
8033     If (InStr(1, TimersFinales(ind), "t1=", 1) > 0) Or (InStr(1, TimersFinales(ind), "t2=", 1) > 0) Or (InStr(1,
TimersFinales(ind), "t3=", 1) > 0) Or (InStr(1, TimersFinales(ind), "t4=", 1) > 0) Or (InStr(1, TimersFinales(ind), "t5=", 1)
> 0) Or (InStr(1, TimersFinales(ind), "t6=", 1) > 0) Or (InStr(1, TimersFinales(ind), "t7=", 1) > 0) Or (InStr(1,
TimersFinales(ind), "t8=", 1) > 0) Or (InStr(1, TimersFinales(ind), "t9=", 1) > 0) Then
8034         numeroTimers.Add(ind) 'Se guarda el número de enlaces finales donde se encuentra un temporizador
8035     Else
8036         numeroNoTimers.Add(ind) 'Se guarda el número donde no hay un temporizador
8037     End If
8038 Next
8039
8040 For ind = numeroTimers.LongCount - 1 To 0 Step -1
8041     EnlacesFinales.RemoveAt(numeroTimers(ind)) 'Se elimina de la lista de enlaces finales aquellos donde hay un
temporizador
8042 Next ind
8043
8044 For ind = numeroNoTimers.LongCount - 1 To 0 Step -1
8045     TimersFinales.RemoveAt(numeroNoTimers(ind)) 'Se elimina de la lista de temporizadores aquellos donde no hay un
temporizador
8046 Next ind
8047
8048 'Dependiendo el tamaño de las columnas (salidas, entradas, temporizadores, transiciones temporizadores) se crea la
dimension de la tabla de entradas/salidas
8049 If EnlacesFinales.LongCount >= SalidasFinales.LongCount And EnlacesFinales.LongCount >= TimersFinales.LongCount And
EnlacesFinales.LongCount >= TemporizadoresFinales.LongCount Then
8050     For ind = 0 To EnlacesFinales.LongCount - 1
8051         SimulacionForm.DataGridView1.Rows.Add()
8052     Next ind
8053 ElseIf SalidasFinales.LongCount >= TimersFinales.LongCount And SalidasFinales.LongCount >=
TemporizadoresFinales.LongCount Then
8054     For ind = 0 To SalidasFinales.LongCount - 1
8055         SimulacionForm.DataGridView1.Rows.Add()
8056     Next ind
8057 ElseIf TimersFinales.LongCount >= TemporizadoresFinales.LongCount Then
8058     For ind = 0 To TimersFinales.LongCount - 1
8059         SimulacionForm.DataGridView1.Rows.Add()
8060     Next ind
8061 Else
8062     For ind = 0 To TemporizadoresFinales.LongCount - 1
8063         SimulacionForm.DataGridView1.Rows.Add()
8064     Next ind
8065 End If

```

Figura 3.7.1.2.17.6. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Les transicions dels temporitzadors s'avaluen als enllaços i, per tant, cal diferenciar-los en la seva columna adient. Per això, sobre la variable d'enllaços finals es busquen els índexs on realment hi ha una transició de temporitzador (línies 8032-8038) i on no hi ha. A partir d'aquests números s'eliminen de les llistes corresponents, és a dir, de la llista d'enllaços s'eliminen les transicions de temporitzadors i de la llista de transicions de temporitzadors que era una copia de l'altra llista s'eliminen els enllaços normals. Finalment, es disposarà de 4 llistes diferents amb longituds diferents. Es cerca la que té una longitud més llarga i es generen les files de la taula de la classe simulació en funció a aquesta (línies 8049 – 8065).

Finalment s'escriuen les dades columna per columna a la taula de la següent forma:

```

8066 'Se escriben los datos en la tabla de entradas y salidas con los datos obtenidos
8067 EscribirDatosTablaINOUT(EnlacesFinales, 0)
8068 EscribirDatosTablaINOUT(SalidasFinales, 1)
8069 EscribirDatosTablaINOUT(TimersFinales, 2)
8070 EscribirDatosTablaINOUT(TemporizadoresFinales, 3)
8071 Panel1.Enabled = False 'Se deshabilita el panel1 donde esta el grafcet original para que no se modifique durante la
simulación
8072 SimulacionForm.InicioEntradasFalse() 'Se llama a la función correspondiente para iniciar la simulación
8073 SimulacionForm.Timer2.Enabled = True 'Se habilita el temporizador de la simulación
8074 SimulacionForm.Show() 'Se muestra la simulación
8075 End Sub

```

Figura 3.7.1.2.17.7. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Es fa ús de la funció EscribirDatosTablaINOUT la qual agafa la llista amb les dades i la columna adient i escriu sobre la taula d'entrades, sortides i temporitzadors. A més es deshabilita el panell original perquè no es pugui modificar durant la simulació, es truca a un mètode de la classe simulació perquè s'iniciï aquesta, s'habilita un temporitzador propi de la classe i es mostra la taula de simulació. Pel que fa al codi de la funció descrita a l'inici d'aquest paràgraf és el següent:

```

8078 Private Sub EscribirDatosTablaINOUT(sender As List(Of String), numero As Integer)
8079     'Según la entrada que reciba escribe en la tabla el dato en la columna adecuada.
8080     For ind = 0 To sender.LongCount - 1
8081         If numero = 1 Then
8082             SimulacionForm.DataGridView1.Rows(ind).Cells(numero).Value = sender(ind) + " - Q" + (ind \ 8).ToString +
8083             "." + (ind - 8 * (ind \ 8)).ToString
8084             ElseIf numero = 0 Then
8085                 SimulacionForm.DataGridView1.Rows(ind).Cells(numero).Value = sender(ind) + " - I" + (ind \ 8).ToString +
8086                 "." + (ind - 8 * (ind \ 8)).ToString
8087             Else
8088                 SimulacionForm.DataGridView1.Rows(ind).Cells(numero).Value = sender(ind)
8089             End If
8090         Next
8091     End Sub

```

Figura 3.7.1.2.17.8. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

Es veu com s'utilitza la llista d'entrada a la funció per fer el recorregut guardant a la fila i columna adients. En el cas de les entrades i les sortides s'afegeix una lletra I i Q respectivament seguit d'un número identificatiu similar a les marques.

Dintre de la pestanya de la que es tracta en aquest subapartat es troba la casella Detener. Si s'observa el codi es veu el següent:

```

8121 'Función al cerrar la simulación
8122 Private Sub SimulacionClose(sender As Object, ByVal e As FormClosingEventArgs)
8123     SimularToolStripMenuItem.Enabled = True 'Se habilita el botón de simular
8124     DetenerToolStripMenuItem.Enabled = False 'Se deshabilita el botón de detener
8125     SimulacionForm.SimulacionActivada = False 'Se vuelve falso el valor del booleano de simulación activada
8126     Panel1.Enabled = True 'Se habilita el panel del graficet
8127     'Se define una lista con todas los datos de simulación
8128     Dim BorrarListas = {listaLabsSimulacion, listaLabsAccionSimulacion, listaLabsAccionAsociadaSimulacion,
8129     listaEtapasSimulacion, listaAccionesSimulacion, listaAccionesAsociadasSimulacion, listaTimerSimulacion,
8130     listaTimerAsociadoSimulacion, listaLabsTimerSimulacion, listaLabsTimerAsociadoSimulacion, listaLabsTransTimerSimulacion,
8131     listaLabsTransTimerAsociadoSimulacion, listaLabsTransTimerNegadoSimulacion, listaLabsTransTimerAsociadoNegadoSimulacion}
8132     For ind = 0 To BorrarListas.LongCount - 1
8133         BorrarListas(ind).Clear() 'Se borran los elementos de la lista
8134     Next ind
8135     SimulacionForm.Timer2.Enabled = False 'Se para el temporizador de la simulación
8136     SimulacionForm.Close() 'Se cierra la simulación
8137 End Sub
8138 'Función que cierra la simulación al clicar en detener
8139 Private Sub DetenerToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles DetenerToolStripMenuItem.Click
8140     If (Me.MdiChildren.Last.Text.ToString) = "Simulación" Then
8141         SimulacionForm.Timer2.Enabled = False 'Se para el temporizador de la simulación
8142         Me.MdiChildren.Last.Close() 'Se cierra la simulación
8143     End If
8144 End Sub

```

Figura 3.7.1.2.17.9. Captura d'imatge part del codi annex A1. (Font: Pablo Mañas)

La funció Detener en sí es el que s'observa a partir de la línia 8137 on es comprova si el text de la finestra filla es Simulación. En cas afirmatiu es tanca la finestra i s'inhabilita el temporitzador de la simulació. El que es veu a les línies de codi superiors és el que succeeix quan la taula de la simulació es tanca. En aquest cas s'habilita el botó de simular i s'inhabilita el de detenir. El panell que s'havia deshabilitat també s'habilita i acte seguit es restableixen totes les variables de simulació. També es deté el temporitzador i es tanca la simulació.

### 3.7.1.2.18 Procediments pestanya Ayuda

Dintre d'aquesta casella es troben dos botons el primer referent a l'ajuda de posicionament ja mencionada anteriorment i el segon que versa sobre assumptes de dades del creador del programa.

El codi qüestionat al primer cas es tracta a continuació:

```

8165 'Función activar/desactivar ayuda de posicionamiento
8166 Private Sub DesactivarAyudaPosicionamientoToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
DesactivarAyudaPosicionamientoToolStripMenuItem.Click
8167     If AyudaPosicionamiento = True Then 'Si se pulsa y está activada
8168         AyudaPosicionamiento = False 'Se desactiva la ayuda
8169         DesactivarAyudaPosicionamientoToolStripMenuItem.Text = "Activar ayuda posicionamiento" 'Se cambia desactivar por
activar
8170     Else 'Sino:
8171         AyudaPosicionamiento = True 'Se activa la ayuda
8172         DesactivarAyudaPosicionamientoToolStripMenuItem.Text = "Desctivar ayuda posicionamiento" 'Se cambia activar por
desactivar
8173     End If
8174 End Sub
8175 End Class

```

Figura 3.7.1.2.18.1. Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

Així doncs, quan es fa clic sobre la casella en qüestió si el booleà que controla l'ajuda de posicionament és true es torna fals i es modifica el text de la casella. Si, en canvi, l'ajuda està desactivada, en fer clic s'activarà de nou el booleà i tornarà a modificar-se el text de la casella que tornarà a ser igual a com estava en un inici.

Pel que fa a l'altre botó, únicament es mostra un quadre de missatge amb les dades corresponents així com es veu a la figura 3.1.13. El codi creat és el que s'observa a continuació:

```

4646 'Función acerca de...Muestra un mensaje con los datos del programa:
4647 Private Sub AcercaDeToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles AcercaDeToolStripMenuItem.Click
4648     MessageBox.Show("Trabajo de fin de grado: Editor de GRAFCETS Versión 1.0" + Chr(13) + Chr(13) + "Autor:
Pablo Mañas Pellejero" + Chr(13) + Chr(13) + "Tutor: Joan Domingo Peña" + Chr(13) + Chr(13) + "© Escola d'Enginyeria
Barcelona Est (EEBE) -" + Chr(13) + Chr(13) + "Universitat Politècnica de Catalunya (UPC)", "Información del programa")
4649 End Sub

```

Figura 3.7.1.2.18.2. Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

Únicament caldria destacar el significat de Chr(13) que són salts de línia per a que es vegin millor les dades.

### 3.7.1.2.19 Procediments botons de la barra esquerra

Finalment, és necessari explicar com actuen els botons de la barra esquerra del programa. Com tots funcionen de la mateixa manera s'explicaran només un parell d'ells. A nivell de codi en fer clic al dibuix de l'etapa inicial s'executa el següent:

```

3502 'Funciones clic sobre botones de la barra de elementos de la izquierda:
3503 Private Sub BotonEtapa0_Click(sender As Object, e As EventArgs) Handles BotonEtapa0.Click
3504     If (archivo = True) Then
3505         GuardarDatos()
3506         ConstruirEtapa0()
3507         ConstruirTextoEtapa0()
3508         ConstruirTextollanacaEtapa0()
3509         BotonEtapa0.Enabled = False
3510     Else
3511         MessageBox.Show("No hay ningún archivo creado")
3512     End If
3513 End Sub

```

**Figura 3.7.1.2.19.1.** Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

En el cas que no existeixi cap arxiu en teoria no es podria clicar sobre el botó, però tot i així es limita per si succeeixen coses descontrolades i es mostra un missatge d'alerta. En cas que si existeixi un arxiu, es truca a les funcions ja explicades amb anterioritat de guardar dades, construir l'objecte, el text, la marca i, en el cas de l'etapa inicial es deshabilita el propi botó perquè no es generin més etapes inicials.

Altres exemples de botó pot ser el de les transicions. El seu codi és el següent:

```

3531 Private Sub BotonTransicion_Click(sender As Object, e As EventArgs) Handles BotonTransicion.Click
3532     If (archivo = True) And BotonEtapa0.Enabled = False Then
3533         GuardarDatos()
3534         ConstruirTextoTransicion()
3535         EventosRatonTransAlta()
3536         EventosRatonTransBaja()
3537         EventosRatonTrans()
3538         EventosRatonLabTrans()
3539     ElseIf BotonEtapa0.Enabled = True Then
3540         MessageBox.Show("Primero crea una etapa inicial")
3541     Else
3542         MessageBox.Show("No hay ningún archivo creado")
3543     End If
3544 End Sub

```

**Figura 3.7.1.2.19.2.** Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

En la resta de botons es fa la comprovació de si existeix una etapa inicial o no, ja que es considera que sinó existeix una etapa inicial no té sentit fer el diagrama. A més en aquests botons s'afegeix els esdeveniments de ratolí i la funció de ConstruirTextoTransicion realment truca a altre classe on es genera la transició.

L'últim exemple de botó és el d'enllaços.

```

3611 Private Sub BotonEnlace_Click(sender As Object, e As EventArgs) Handles BotonEnlace.Click
3612     If (archivo = True) And BotonEtapa0.Enabled = False Then
3613         GuardarDatos()
3614         ConstruirEnlace()
3615         EventosRatonEnlaceLineaVertical()
3616         EventosRatonEnlaceLineaHorizontal()
3617     ElseIf BotonEtapa0.Enabled = True Then
3618         MessageBox.Show("Primero crea una etapa inicial")
3619     Else
3620         MessageBox.Show("No hay ningún archivo creado")
3621     End If
3622 End Sub

```

**Figura 3.7.1.2.19.3.** Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

A primera vista no hi ha cap diferència amb el botó anterior, tot i així cal observar la funció ConstruirEnlace, la qual és la següent:

```

2794 'Función para la construcción del enlace:
2795 Private Sub ConstruirEnlace()
2796     'Se construye la caja con los elementos del Grafcet
2797     For ind = 0 To listaLabsTrans.LongCount - 1
2798         LineaForm.ListBox1.Items.Add("Trans.Nivel: " + listaLabsTrans(ind).Text)
2799     Next ind
2800     For ind = 0 To listaLabsTransAlto.LongCount - 1
2801         LineaForm.ListBox1.Items.Add("Trans.F.Alto: " + listaLabsTransAlto(ind).Text)
2802     Next ind
2803     For ind = 0 To listaLabsTransBajo.LongCount - 1
2804         LineaForm.ListBox1.Items.Add("Trans.F.Bajo: " + listaLabsTransBajo(ind).Text)
2805     Next ind
2806     For ind = 0 To listaLabsTransConvergencia.LongCount - 1
2807         LineaForm.ListBox1.Items.Add("Convergencia.Nivel: " + listaLabsTransConvergencia(ind).Text)
2808     Next ind
2809     For ind = 0 To listaLabsTransAltoConvergencia.LongCount - 1
2810         LineaForm.ListBox1.Items.Add("Convergencia.F.Alto: " + listaLabsTransAltoConvergencia(ind).Text)
2811     Next ind
2812     For ind = 0 To listaLabsTransBajoConvergencia.LongCount - 1
2813         LineaForm.ListBox1.Items.Add("Convergencia.F.Bajo: " + listaLabsTransBajoConvergencia(ind).Text)
2814     Next ind
2815     For ind = 0 To listaLabsTransDivergencia.LongCount - 1
2816         LineaForm.ListBox1.Items.Add("Divergencia.Nivel: " + listaLabsTransDivergencia(ind).Text)
2817     Next ind
2818     For ind = 0 To listaLabsTransAltoDivergencia.LongCount - 1
2819         LineaForm.ListBox1.Items.Add("Divergencia.F.Alto: " + listaLabsTransAltoDivergencia(ind).Text)
2820     Next ind
2821     For ind = 0 To listaLabsTransBajoDivergencia.LongCount - 1
2822         LineaForm.ListBox1.Items.Add("Divergencia.Nivel.F.Bajo: " + listaLabsTransBajoDivergencia(ind).Text)
2823     Next ind
2824     LineaForm.ShowDialog() 'Se llama a la clase lineaForm para crear la línea, el enlace o ambas
2825 End Sub

```

**Figura 3.7.1.2.19.4.** Captures d'imatge part del codi annex A1. (Font: Pablo Mañas)

Tal i com es pot observar a la figura anterior es recorren les llistes de les diferents etiquetes existents al panell i es van afegint per ordre a la llista (ListBox1) de la classe Linea. Després es truca a la classe en sí per crear l'enllaç.

Això és tot pel que fa als botons del menú principal i dels procediments de la classe d'aquest menú. A continuació es farà el mateix amb la resta de classes, tot i que no serà tan extens ja que el nucli del programa s'ha centrat en aquesta classe.

### 3.7.2. Nou Document

Es truca quan es vol generar un nou document a partir del botó del menú principal Archivo -> Nuevo. Les seves característiques a nivell de codi es mostren a continuació.

#### 3.7.2.1. Variables

Únicament disposa d'una variable pública que guarda la mida del paper en una cadena. De fet és la variable que es guarda inicialment quan s'emmagatzemen dades amb la funció GuardarDatos.

#### 3.7.2.2. Procediments i funcions

L'aspecte de la classe quan es mostra és el que s'observa a la figura 3.1.3. Es tracta de 4 RadioButton i dos botons, un d'acceptar i altre de cancel·lar. A continuació es desglossen els diferents mètodes que apareixen dintre d'aquesta classe.

##### 3.7.2.2.1 Procediment Botó Acceptar

En fer clic a Acceptar s'executa el codi següent:

```

5      'Botón aceptar:
6      Public Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
7          If RadioButton1.Checked Then 'Si se selecciona la primera opción:
8              CrearPanel(611, 881, 595, 842) 'Se crea la ventana y el panel en el tamaño escogido
9              Activar() 'Se activan los botones convenientes
10             Me.Close() 'Se cierra la clase
11         ElseIf RadioButton2.Checked Then 'Si se selecciona la segunda opción:
12             CrearPanel(858, 634, 842, 595) 'Se crea la ventana y el panel en el tamaño escogido
13             Activar() 'Se activan los botones convenientes
14             Me.Close() 'Se cierra la clase
15         ElseIf RadioButton3.Checked Then 'Si se selecciona la tercera opción:
16             CrearPanel(1207, 881, 1191, 842) 'Se crea la ventana y el panel en el tamaño escogido
17             Activar() 'Se activan los botones convenientes
18             Me.Close() 'Se cierra la clase
19         ElseIf RadioButton4.Checked Then 'Si se selecciona la cuarta opción:
20             CrearPanel(858, 1230, 842, 1191) 'Se crea la ventana y el panel en el tamaño escogido
21             Activar() 'Se activan los botones convenientes
22             Me.Close() 'Se cierra la clase
23         Else
24             MessageBox.Show("Selecciona una opción") 'Si no se selecciona ninguna opción se muestra mensaje
25         End If
26     End Sub

```

**Figura 3.7.2.2.1.1.** Captura d'imatge part del codi annex A2. (Font: Pablo Mañas)

En funció de quin RadioButton estigui polsat, es crearà un tipus de panell o altre, és a dir, el panell tindrà una mida diferent. Si no es selecciona cap i s'intenta acceptar el programa avisarà amb un

missatge. En cas que es seleccioni algun RadioButton i s'accepti es crea el panell, s'activen certs botons del menú principal i es tanca la classe en qüestió.

### 3.7.2.2.2 Procediment CrearPanel

La funció crear panel consta de 4 paràmetres que determinaran l'alçada i amplada del panell i les mateixes característiques de la finestra filla creada. El seu codi és el següent:

```

68 'Función que crea la ventana y el panel en función a los números introducidos:
69 Public Sub CrearPanel(anchoForm As Integer, altoForm As Integer, anchoHoja As Integer, altoHoja As Integer)
70     Dim ChildForm As New Form With {MdiParent = MenuPrincipalForm, .Text = "Sin Titulo", .BackColor = Color.Gray, .Width
= anchoForm, .Height = altoForm, .AutoScroll = True, .MaximizeBox = False, .MinimizeBox = False, .FormBorderStyle =
FormBorderStyle.FixedSingle} 'Se define la ventana hija que contiene el panel
71     AddHandler ChildForm.FormClosing, AddressOf ChildClose 'Se le añade la función de cerrar la ventana hija
72     ChildForm.Show() 'Se muestra la ventana hija
73     Dim PanelNuevo As New Panel With {.Width = anchoHoja, .Height = altoHoja, .BackColor = Color.White, .Top = 0, .Left =
0} 'Se crea el panel con las características
74     GuardarTamañoPapel = PanelNuevo.Width.ToString + PanelNuevo.Height.ToString 'Se guarda en la variable el tamaño del
papel
75     MenuPrincipalForm.Panel1 = PanelNuevo 'Se guarda en la variable que controla en el módulo principal el panel
76     MenuPrincipalForm.ChildForm = ChildForm 'Se añade como ventana hija a la ventana principal
77     ChildForm.Controls.Add(MenuPrincipalForm.Panel1) 'Se añade el panel en la ventana hija
78 End Sub
    
```

Figura 3.7.2.2.2.1. Captura d'imatge part del codi annex A2. (Font: Pablo Mañas)

En executar-se es declara una finestra filla amb les característiques necessàries perquè el control recaigui sota la classe menú principal, amb una alçada i una amplada determinada pels dos primers paràmetres un títol com a text. També s'afegeix l'esdeveniment quan es tanca aquesta finestra filla. Més tard es crea el panell nou amb les dimensions dels arguments 3 i 4 i s'actualitza la variable d'aquesta classe amb aquestes mides. Es guarda la variable del panell a la variable panell del menú principal juntament amb la finestra filla. Per últim s'afegeix el panell a la finestra filla.

### 3.7.2.2.3 Procediment ChildClose

De la funció anterior es deriva la funció ChildClose que respon a l'esdeveniment de tancar la finestra filla. El codi creat per aquesta situació és el següent:

```

33 'Función al cerrar la ventana que contiene el panel
34 Private Sub ChildClose(sender As Object, ByVal e As FormClosingEventArgs)
35     Try
36         EnlacesForm.Close() 'Se intenta cerrar la ventana de los enlaces
37     Catch
38     End Try
39     Dim resultado As DialogResult 'Se abre diálogo al intentar cerrar
40     resultado = MessageBox.Show("Si cierras no se guardará el progreso", "Cerrar", MessageBoxButtons.OKCancel,
MessageBoxIcon.Warning)
41     If resultado = DialogResult.OK Then 'Si se acepta:
42         e.Cancel = False 'Se cierra
43         MenuPrincipalForm.archivo = False 'Se define la variable archivo a falso
44         MenuPrincipalForm.BorrarTodo() 'Se borra todo (variables creadas, se limpia el panel, etc...)
45         MenuPrincipalForm.Deshabilitar() 'Se deshabilitan los botones nuevamente
46         EnlacesForm.DataGridView1.Rows.Clear() 'Se borra la tabla de enlaces
47     Try
48         EnlacesForm.Close() 'Se vuelve a intentar cerrar la ventana de enlaces
49     Catch
50     End Try
51     Clipboard.Clear() 'Se elimina el portafolios
52     Else
53         e.Cancel = True 'Si se cancela no pasa nada
54     End If
55 End Sub
    
```

Figura 3.7.2.2.3.1. Captura d'imatge part del codi annex A2. (Font: Pablo Mañas)

En primera instància es prova de tancar la finestra dels enllaços ja que pot ser molest i no es veuria el missatge d'alerta. Sempre es genera el missatge d'alerta que diu que si no es tanca no es guardarà el progrés i en el cas que s'accepti es tancarà la finestra, la variable arxiu es tornarà falsa, es borraràn totes les variables, es deshabilitaran els botons, es netejarà la taula d'enllaços i es restablirà el

portafolis. En cas que es cancel·li no passarà res. Aquesta funció pot generar problemes ja que si es tanca l'aplicació o es crea un nou arxiu sempre mostrarà el missatge d'alerta que caldrà acceptar. Per arreglar-la caldria limitar-la abans de presentar el diàleg.

#### 3.7.2.2.4 Procediment Activar

Un cop es crea el panell nou i la finestra, cal activar els botons del menú principal que permeten fer el grafcet. Per tant, aquesta funció genera una llista amb tots els elements que s'habilitaran, es recorre amb un bucle i es modifica per cada un l'atribut Enabled. També es modifica la variable global archivo del menu principal per determinar que s'ha creat un arxiu. A més es deshabilita el botó de detenir. Com la variable és pública pot ser utilitzada per altres classes. Tot això s'observa al fragment de codi següent:

```

57     'Función que activa los diferentes botones
58     Public Sub Activar()
59         'Se define la lista con todos los botones a activar:
60         Dim ConjuntoDeBooleanos = {MenuPrincipalForm.DesactivarAyudaPosicionamientoToolStripMenuItem,
MenuPrincipalForm.CerrarToolStripMenuItem, MenuPrincipalForm.GuardarToolStripMenuItem,
MenuPrincipalForm.GuardarComoToolStripMenuItem, MenuPrincipalForm.ExportarToolStripMenuItem,
MenuPrincipalForm.ImprimirToolStripMenuItem, MenuPrincipalForm.VistaPreliminarToolStripMenuItem,
MenuPrincipalForm.DeshacerToolStripMenuItem, MenuPrincipalForm.CortarToolStripMenuItem,
MenuPrincipalForm.CopiarToolStripMenuItem, MenuPrincipalForm.PegarToolStripMenuItem,
MenuPrincipalForm.SeleccionarTodoToolStripMenuItem, MenuPrincipalForm.SimilarToolStripMenuItem,
MenuPrincipalForm.VerEnlacesToolStripMenuItem, MenuPrincipalForm.MarcasToolStripMenuItem, MenuPrincipalForm.BotonEtapa0,
MenuPrincipalForm.BotonEtapa, MenuPrincipalForm.BotonTransicion, MenuPrincipalForm.BotonAccion,
MenuPrincipalForm.BotonAccionAsociada, MenuPrincipalForm.BotonEtiqueta, MenuPrincipalForm.BotonEnlace,
MenuPrincipalForm.BotonConvergencia, MenuPrincipalForm.BotonDivergencia, MenuPrincipalForm.BotonTimer,
MenuPrincipalForm.BotonTimerAsociado}
61         For ind = 0 To ConjuntoDeBooleanos.LongCount - 1 'Se recorre la lista
62             ConjuntoDeBooleanos(ind).Enabled = True 'Se habilitan los elementos de la lista
63         Next
64         MenuPrincipalForm.archivo = True 'Se vuelve verdadera la variable archivo
65         MenuPrincipalForm.DetenerToolStripMenuItem.Enabled = False 'Se deshabilita el botón detener
66     End Sub

```

Figura 3.7.2.2.4.1. Captura d'imatge part del codi annex A2. (Font: Pablo Mañas)

#### 3.7.2.2.5 Procediment Botó Cancel·lar

Si es polsa a cancel·lar únicament tanca la classe sense fer res mes, no cal mostrar el codi.

### 3.7.3. Transicions

Aquesta classe es truca quan es vol crear una transició a partir d'un dels botons de la barra esquerra del menú principal. Quan es mostra s'observa una finestra com la de la figura 3.2.2. A continuació s'exposen les variables i els mètodes emprats.

#### 3.7.3.1. Variables

En aquesta classe es tracta una variable de tipus label que defineix l'etiqueta que es genera a la transició amb la condició introduïda per l'usuari. També es disposa de 5 variables globals dos de les quals són del tipus PictureBox que permeten generar la imatge dels flancs i les transicions i les altres 3 són les llistes que controlen cada tipus de transició.



### 3.7.3.2. Procediments i funcions

Així com en la classe anterior en aquesta també s'utilitzen RadioButtons i dos botons d'Acceptar i Cancel·lar. Els procediments són els següents:

#### 3.7.3.2.1 Procediment Botó Acceptar

El codi corresponent a aquesta funció s'observa a aquesta imatge:

```

11     'Botón aceptar:
12     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
13         If RadioButton1.Checked Then 'Si la primera opción se selecciona:
14             ConstruirTransicionAlta() 'Se construye la transición de flanco alto
15             CrearFlancoVLabel("Imágenes\FlancoSubida.png", MenuPrincipalForm.listaLabsTransAlto,
MenuPrincipalForm.listaEstadoAlto) 'Se construye flanco y la etiqueta
16             Me.Close() 'Se cierra la clase
17         ElseIf RadioButton2.Checked Then 'Si la segunda opción se selecciona:
18             ConstruirTransicionBaja() 'Se construye la transición de flanco bajo
19             CrearFlancoVLabel("Imágenes\FlancoBajada.png", MenuPrincipalForm.listaLabsTransBajo,
MenuPrincipalForm.listaEstadoBajo) 'Se construye flanco y la etiqueta
20             Me.Close() 'Se cierra la clase
21         ElseIf RadioButton3.Checked Then 'Si la tercera opción se selecciona:
22             ConstruirTransicion() 'Se construye la transición
23             CrearLabelTransicion() 'Se construye la etiqueta de la transición
24             Me.Close() 'Se cierra la clase
25         Else
26             MessageBox.Show("Selecciona un tipus de condició d'activació") 'Si no se selecciona nada se muestra un mensaje
27         End If
28     End Sub

```

Figura 3.7.3.2.1.1. Captura d'imatge part del codi annex A3. (Font: Pablo Mañas)

En funció de quin RadioButton estigui seleccionat es trucarà a unes funcions o a altres. Tot i així el funcionament en els tres casos és el mateix. Primer es crea la transició del tipus que s'hagi seleccionat i després l'etiqueta o a més de l'etiqueta els flancs. Si no hi ha res seleccionat s'avisarà amb un missatge.

#### 3.7.3.2.2 Procediments de Construir Transició

Com ja s'ha dit al subapartat anterior, els tres casos són molt similars i per això s'adjunten a continuació els codis de les diferents funcions:

```

35     Public Sub ConstruirTransicion()
36         CrearTransicion() 'Se crea la transición
37         listaTrans.Add(transicion) 'Se añade a la lista correspondiente
38         MenuPrincipalForm.listaCoordsTrans.Add(transicion.Location) 'Se añaden las coordenadas a la lista correspondiente
39     End Sub
40
41     Public Sub ConstruirTransicionAlta()
42         CrearTransicion() 'Se crea la transición
43         listaTransAlto.Add(transicion) 'Se añade a la lista correspondiente
44         MenuPrincipalForm.listaCoordsTransAlto.Add(transicion.Location) 'Se añaden las coordenadas a la lista correspondiente
45     End Sub
46
47     Public Sub ConstruirTransicionBaja()
48         CrearTransicion() 'Se crea la transición
49         listaTransBajo.Add(transicion) 'Se añade a la lista correspondiente
50         MenuPrincipalForm.listaCoordsTransBajo.Add(transicion.Location) 'Se añaden las coordenadas a la lista correspondiente
51     End Sub

```

Figura 3.7.3.2.2.1. Captura d'imatge part del codi annex A3. (Font: Pablo Mañas)

En els tres casos es truca a una nova funció que crea la transició i s'afegeix a la llista corresponent la transició i la coordenada.

#### 3.7.3.2.3 Procediment Crear Transició

Repetida en els tres casos aquesta funció permet crear la imatge de la transició al panell. El codi que ho demostra és el següent:

```

53     Public Sub CrearTransicion()
54         transicion = New PictureBox With {.Image = Image.FromFile("Imágenes\Transicion.png"), .SizeMode =
PictureBoxSizeMode.StretchImage, .Width = 30, .Height = 30, .Left = 200, .Top = 100, .Cursor = Cursors.SizeAll, .Tag = False}
'Define características de transición
55         MenuPrincipalForm.Panel1.Controls.Add(transicion) 'Se añade al panel la transición
56     End Sub

```

**Figura 3.7.2.3.1.3.** Captura d'imatge part del codi annex A3. (Font: Pablo Mañas)

Primer es defineix la transició amb les característiques inicials i després únicament s'afegeix al panell.

### 3.7.3.2.4 Procediment Crear Label Transició

En aquesta funció únicament es considera la creació de l'etiqueta que acompanya a la transició sense flancs. Així doncs, es defineix la variable de l'etiqueta amb les característiques, s'afegeix al panell i es col·loca al front. Finalment s'afegeix a la llista corresponent. El codi que s'executa és el que es mostra a la figura següent:

```

58     Public Sub CrearLabelTransicion()
59         LabelTrans = New Label With {.AutoSize = True, .Text = TextBox1.Text, .Left = transicion.Left + transicion.Width(),
.Top = transicion.Top + (transicion.Height() / 3), .BackColor = Color.Transparent, .Height = 13, .Cursor = Cursors.SizeAll}
'Define características de etiqueta de la transición
60         MenuPrincipalForm.Panel1.Controls.Add(LabelTrans) 'Se añade al panel la etiqueta
61         LabelTrans.BringToFront() 'Se coloca al frente en el panel
62         MenuPrincipalForm.listaLabsTrans.Add(LabelTrans) 'Se añade a la lista correspondiente
63     End Sub

```

**Figura 3.7.3.2.4.1.** Captura d'imatge part del codi annex A3. (Font: Pablo Mañas)

### 3.7.3.2.5 Procediment Crear Flanc i Label

En aquest mètode es crea el flanc amb la imatge i les característiques adjacents i es situa al panell. Després es crea l'etiqueta amb les característiques respecte el flanc, es situa al panell es porta al front i s'afegeix a la llista corresponent tant el flanc com l'etiqueta.

```

65     Public Sub CrearFlancoYLabel(ruta As String, listaLabs As List(Of Label), listaEstado As List(Of PictureBox))
66         Flanco = New PictureBox With {.Image = Image.FromFile(ruta), .SizeMode = PictureBoxSizeMode.StretchImage, .Width = 10,
.Height = 10, .Left = transicion.Left + transicion.Width(), .Top = transicion.Top + (transicion.Height() / 3)} 'Define
características del flanco
67         MenuPrincipalForm.Panel1.Controls.Add(Flanco) 'Se añade al panel el flanco
68         Flanco.BringToFront() 'Se coloca al frente en el panel
69         LabelTrans = New Label With {.AutoSize = True, .Text = TextBox1.Text, .Left = Flanco.Left + Flanco.Width(), .Top =
transicion.Top + Flanco.Height(), .Height = 13, .BackColor = Color.Transparent, .Cursor = Cursors.SizeAll} 'Define
características de etiqueta de la transición
70         MenuPrincipalForm.Panel1.Controls.Add(LabelTrans) 'Se añade al panel la etiqueta
71         LabelTrans.BringToFront() 'Se coloca al frente en el panel
72         listaLabs.Add(LabelTrans) 'Se añade a la lista correspondiente
73         listaEstado.Add(Flanco) 'Se añade a la lista correspondiente
74     End Sub

```

**Figura 3.7.3.2.5.1.** Captura d'imatge part del codi annex A3. (Font: Pablo Mañas)

Totes aquestes funcions són públiques ja que es criden des de el menú principal en algunes ocasions.

El procediment de Cancel·lar és igual que en el subapartat anterior, simplement tanca la classe.

### 3.7.4. Accions

Així com les classes anteriors aquesta es truca a través d'un dels botons de la barra de l'esquerra del menú. A la finestra principal es mostra el que s'observa a la figura 3.2.4. Les variables i els procediments que intervenen són els que es citen a continuació:

#### 3.7.4.1. Variables

Únicament s'utilitzen 3 variables. La primera conté l'etiqueta, la segona un objecte de tipus PictureBox i la tercera una llista del tipus anterior.

#### 3.7.4.2. Procediments i funcions

En aquest cas únicament es mostra un quadre amb dos botons d'acceptar i cancel·lar i un quadre de text on l'usuari introdueix el nom de l'acció.

##### 3.7.4.2.1 Procediment Botó Acceptar

Quan s'accepta es truca a construir l'acció en sí, o sigui l'objecte i l'etiqueta. Finalment es tanca la classe un cop s'ha creat l'acció. El fragment del codi és el següent:

```

8      'Botón aceptar:
9      Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
10         ConstruirAccion() 'Se llama a construir la acción
11         ConstruirLabelAccion() 'Se llama a construir la etiqueta de la acción
12         Me.Close() 'Se cierra la clase
13     End Sub

```

Figura 3.7.4.2.1.1. Captura d'imatge part del codi annex A4. (Font: Pablo Mañas)

##### 3.7.4.2.2 Procediment Construir Acció

La funció que es truca primerament executa el codi següent:

```

20     Public Sub ConstruirAccion()
21         Accion = New PictureBox With {.Image = Image.FromFile("Imágenes\Accion.png"), .SizeMode =
PictureBoxSizeMode.StretchImage, .Width = 90, .Height = 28, .Left = 200, .Top = 100, .Cursor = Cursors.SizeAll, .Tag = False}
'Se definen las características de la acción
22         MenuPrincipalForm.Panel1.Controls.Add(Accion) 'Se añade al panel la acción
23         ListaAccion.Add(Accion) 'Se añade a la lista correspondiente la acción
24         MenuPrincipalForm.listaCoordsAccion.Add(Accion.Location) 'Se añaden las coordenadas a la lista correspondiente
25     End Sub

```

Figura 3.7.4.2.2.1. Captura d'imatge part del codi annex A4. (Font: Pablo Mañas)

Es defineix la variable de tipus PictureBox amb les característiques corresponents i s'afegeix al panell. Després s'afegeix a la variable corresponent l'acció i la localització a la variable controlada al menú principal.

##### 3.7.4.2.3 Procediment Construir Label Acció

Pel que fa a la construcció de l'etiqueta el funcionament és el mateix que en totes les etiquetes i no cal reiterar l'explicació. El codi corresponent és el següent:

```

27     Public Sub ConstruirLabelAccion()
28         LabelAccion = New Label With {.AutoSize = True, .Text = TextBox1.Text, .Left = Accion.Left + (Accion.Width() / 3.6),
        .Top = Accion.Top + (Accion.Height() / 4), .Height = 13, .Width = 60, .Parent = Accion, .BackColor = Color.Transparent} 'Se
        definen las características de la etiqueta de la acción
29         MenuPrincipalForm.Panel1.Controls.Add(LabelAccion) 'Se añade al panel la etiqueta
30         LabelAccion.BringToFront() 'Se coloca al frente en el panel
31         LabelAccion.ResetBackColor() 'Se reinicia su color de fondo
32         MenuPrincipalForm.listaLabsAccion.Add(LabelAccion) 'Se añade a la lista correspondiente
33     End Sub

```

**Figura 3.7.4.2.3.1.** Captura d'imatge part del codi annex A4. (Font: Pablo Mañas)

El procediment de Cancel·lar com és habitual tanca la classe.

### 3.7.5. Accions associades

Aquesta classe funciona exactament igual que l'anterior, ja que es defineixen els mateixos tipus de variables i procediments. Les estructures utilitzades són les mateixes també i les úniques diferències es troben en les imatges dels PictureBox i en les llistes a les que es fan referència ja que no és el mateix una acció associada que una acció normal.

### 3.7.6. Canviar etiquetes etapes

Aquesta classe es truca quan es vol fer un canvi d'etiqueta al fer doble clic com ja s'ha vist anteriorment. El quadre que apareix al mostrar-se és el que s'observa a la figura 3.3.5. A continuació es mostrarà les variables i el codi dels procediments que intervenen:

#### 3.7.6.1. Variables

Únicament es declara una variable pública del tipus String que guarda el número escollit per l'usuari.

#### 3.7.6.2. Procediments i funcions

Únicament hi ha dos procediments, el que es genera en fer clic al botó d'acceptar i el que s'activa quan es cancel·la l'operació. El procediment de cancel·lar com sempre tan sols tanca la classe.

El codi del procediment del botó acceptar és el següent:

```

6     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
7         nuevaEtiquetaEtapa = NumericUpDown1.Text.ToString() 'Se guarda el número elegido en la variable
8         Me.Close() 'Se cierra la clase
9     End Sub

```

**Figura 3.7.6.2.1.** Captura d'imatge part del codi annex A6. (Font: Pablo Mañas)

Tal i com es veu es guarda a la variable global definida anteriorment el número escollit amb el quadre NumericUpDown1 que es troba en el quadre de diàleg. Després es tanca la classe ja que el menú principal avaluarà la variable pública d'aquesta classe.

### 3.7.7. Canviar etiquetes

En fer doble clic sobre una etiqueta que no sigui marca ni número d'etapa s'obre el quadre de diàleg que s'observa a la figura 3.3.7. En aquest quadre s'observa un TextBox i els dos botons d'acceptar i cancel·lar.

#### 3.7.7.1. Variables

Així com en el cas anterior únicament hi existeix una variable pública que guarda el que escriu l'usuari al text a través del procediment d'acceptar.

#### 3.7.7.2. Procediments i funcions

El codi del procediment del botó acceptar es mostra a la figura següent:

```

5      'Botón aceptar:
6      Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
7          nuevaEtiqueta = TextBox1.Text.ToString() 'Se guarda el texto de la nueva etiqueta en la variable
8          Me.Close() 'Se cierra la clase
9      End Sub

```

Figura 3.7.7.2.1. Captura d'imatge part del codi annex A7. (Font: Pablo Mañas)

L'única diferència amb el subapartat anterior és la variable en aquest cas i el NumericUpDown que ara no existeix i es un TextBox. La funció del botó cancel·lar com sempre tanca la classe.

### 3.7.8. Canviar etiquetes marques

Aquesta classe s'activa quan es fa doble clic sobre una marca que no sigui la de l'etapa inicial. El quadre de diàleg que apareix es pot veure a la figura 3.3.6. En aquest cas el procediment d'acceptar i cancel·lar són iguals que en el cas anterior i la variable torna a ser una sola. L'única diferència observable és la funció següent:

```

5      Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles
        TextBox1.KeyPress
6
7          'Limitación del cuadro de texto para que sólo se escriban números:
8          '97 - 122 = Código Ascii letras minúsculas
9          '65 - 90 = Código Ascii letras mayúsculas
10         '48 - 57 = Código Ascii números
11
12         If Asc(e.KeyChar) <> 8 Then
13             If Asc(e.KeyChar) < 46 Or Asc(e.KeyChar) > 57 Or Asc(e.KeyChar) = 47 Then 'Se limitan los números y el signo "."
14                 e.Handled = True
15             End If
16         End If
17
18     End Sub

```

Figura 3.7.8.1. Captura d'imatge part del codi annex A8. (Font: Pablo Mañas)

Tal i com es veu a la capçalera és un tipus de funció que s'activa quan es polsa una tecla del teclat si el TextBox1 està activat. Ja que una marca es defineix únicament per números, es limita les tecles que es puguin escriure. Tan sols es permet doncs, escriure números i el caràcter ".".

### 3.7.9. Convergències i Divergències

Degut a què els procediments i les variables utilitzades són les mateixes que a les transicions amb les modificacions de les imatges i alguns atributs diferents com les posicions on apareixen els objectes no es considera necessari repetir el mateix que al subapartat 3.7.3. Així doncs aquestes dues classes es consideren pràcticament iguals i si es vol veure el codi font de cada un dels procediments es troba a l'annex 9 i 10 respectivament.

### 3.7.10. Enllaços

És la classe que es crida quan es fa clic en veure els enllaços. Aquesta mostra la taula dels enllaços i un botó que la actualitza així com s'observa a la figura 3.4.1.

#### 3.7.10.1. Variables

Dintre de la classe és declara una sola variable privada que permet actualitzar les dades en el moment en que es fa clic al botó d'actualitzar enllaços.

#### 3.7.10.2. Procediments i funcions

Tan sols consta de dos procediments, un referent al botó de l'actualització i l'altra a quan es tanca la taula.

##### 3.7.10.2.1 Procediment Botó Actualitzar enllaços

En fer clic a actualitzar els enllaços es reinicia la variable privada que s'havia declarat i s'utilitza la funció de EscribirArchivoGuardado perquè al final d'aquesta sempre es fa l'actualització de la taula d'enllaços com ja s'ha explicat anteriorment. Així doncs, el codi relacionat amb aquesta funció és el següent:

```

5      'Clic a actualizar tabla:
6      Private Sub ActualizarTabla_Click(sender As Object, e As EventArgs) Handles ActualizarTabla.Click
7          DatosActualizados = ""
8          MenuPrincipalForm.EscribirArchivoGuardado(DatosActualizados) 'Se actualiza la tabla con los datos actuales al clicar
9      End Sub

```

Figura 3.7.10.2.1.1. Captura d'imatge part del codi annex A11. (Font: Pablo Mañas)

##### 3.7.10.2.2 Procediment Tancament de la classe

Degut a que si es tanca la taula d'enllaços es perd informació en alguna de les funcions i fallaria la simulació, és necessari afegir línies de codi que permetin que tot marxi de forma correcta.

```

11 Private Sub EnlacesForm_FormClosing(sender As Object, e As FormClosingEventArgs) Handles MyBase.FormClosing
12     If SimulacionForm.SimulacionActivada = True Then
13         SimulacionForm.Close() 'Cierra la simulación en cuanto se cierra la tabla de enlaces si está activa (evita
14         problemas)
15     End If
16 End Sub

```

**Figura 3.7.10.2.2.1.** Captura d'imatge part del codi annex A11. (Font: Pablo Mañas)

El codi font que es mostra en la imatge anterior mostra com en quan es tanca la taula d'enllaços, si existeix una simulació activa, es tanca també la simulació ja que sinó podria fallar aquesta.

### 3.7.11. Etiquetes

De forma similar a altres classes, aquesta en circular-se genera un quadre de diàleg amb un TextBox i dos botons, un d'acceptar i altre de cancel·lar tal i com es mostra a la figura 3.2.6.

#### 3.7.11.1. Variables

Es declara una sola variable pública del tipus Label que contindrà el text que l'usuari decideixi.

#### 3.7.11.2. Procediments i funcions

En aquest cas l'únic procediment interessant és el de CrearEtiqueta ja que la resta són els botons d'acceptar i cancel·lar. El primer truca a la funció que es considera a observar i tanca la classe i el segon, com sempre, tanca la classe.

El codi del procediment de CrearEtiqueta és el següent:

```

17 Public Sub CrearEtiqueta()
18     etiqueta = New Label With {.AutoSize = True, .Text = TextBox1.Text, .Left = 0, .Top = 0, .BackColor =
19     Color.Transparent, .Cursor = Cursors.SizeAll, .Tag = False}
20     MenuPrincipalForm.Panell.Controls.Add(etiqueta) 'Se añade al panel la etiqueta
21     etiqueta.BringToFront() 'Se expone al frente
22     MenuPrincipalForm.listaEtiquetas.Add(etiqueta) 'Se añade a la lista de las etiquetas
23     MenuPrincipalForm.listaCoordsEtiqueta.Add(etiqueta.Location) 'Se añade su posición a la lista de coordenadas
24 End Sub

```

**Figura 3.7.11.2.1.** Captura d'imatge part del codi annex A12. (Font: Pablo Mañas)

Així com la resta d'etiquetes ja explicades, aquestes funcionen de la mateixa manera. Es declara la nova etiqueta amb les noves característiques, s'afegeix al panell, s'exposa al front i se situa a les llistes corresponents.

### 3.7.12. Línies

Aquesta classe es mostra quan es vol crear un enllaç, és a dir quan es fa clic sobre el botó de la barra d'eines de l'esquerra del menú principal. Pel que fa al seu aspecte es pot observar a la figura 3.2.7. Es pot veure un NumericUpDown, una ListBox, dos RadioButton i dos botons d'acceptar i cancel·lar.

### 3.7.12.1. Variables

En aquesta classe es declaren 4 variables públiques. La primera és la línia en general que tindrà la imatge en un PictureBox, la segona i la tercera són llistes d'aquestes línies que es poden crear i l'última es una variable que guarda els enllaços que s'han fet de forma manual, o sigui que s'han seleccionat de la llista amb el número de l'etapa seleccionat al NumericUpDown de la part superior.

### 3.7.12.2. Procediments i funcions

Ara doncs s'explicaran de forma detallada els mètodes que apareixen dintre de la classe.

#### 3.7.12.2.1 Procediment Botó Acceptar

```

9      'Botón aceptar:
10     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
11     If RadioButton1.Checked Then 'Si se selecciona la primera opción:
12     If ListBox1.SelectedItems.Count > 0 Then 'Si se selecciona algún elemento de la lista
13     TipoEnlace.ShowDialog() 'Se abre la clase para elegir el tipo de enlace anterior o siguiente
14     CrearEnlace() 'Se crea el enlace manual
15     End If
16     ConstruirLinea("Imagenes\LineaVertical.png", MenuPrincipalForm.listaCoordsLineaVertical, listaLineaVertical, 1,
30) 'Se crea la línea vertical
17     RadioButton1.Checked = False 'Se deselecciona el botón seleccionado
18     Me.Close() 'Se cierra la clase
19     ElseIf RadioButton2.Checked Then 'Si se selecciona la segunda opción:
20     If ListBox1.SelectedItems.Count > 0 Then 'Si se selecciona algún elemento de la lista
21     TipoEnlace.ShowDialog() 'Se abre la clase para elegir el tipo de enlace anterior o siguiente
22     CrearEnlace() 'Se crea el enlace manual
23     End If
24     ConstruirLinea("Imagenes\LineaHorizontal.png", MenuPrincipalForm.listaCoordsLineaHorizontal, listaLineaHorizontal,
30, 1) 'Se crea la línea vertical
25     RadioButton2.Checked = False 'Se deselecciona el botón seleccionado
26     Me.Close() 'Se cierra la clase
27     ElseIf ListBox1.SelectedItems.Count > 0 Then 'Si se selecciona algún elemento de la lista
28     TipoEnlace.ShowDialog() 'Se abre la clase para elegir el tipo de enlace anterior o siguiente
29     CrearEnlace() 'Se crea el enlace manual
30     Else 'Si no se selecciona nada:
31     MessageBox.Show("Seleccione un tipo de condición de activación o un elemento de la lista") 'Se muestra un mensaje
32     End If
33 End Sub

```

**Figura 3.7.12.2.1.1.** Captura d'imatge part del codi annex A13. (Font: Pablo Mañas)

Quan es polsa el botó acceptar primer cal determinar si algun dels RadioButton està marcat o algun element de la llista està seleccionat. En cas contrari s'adverteix amb un missatge (línia 31). Bé doncs, si per exemple el primer RadioButton està seleccionat el qual correspon a la creació d'una línia vertical i no hi ha cap element seleccionat a la llista, simplement es construeix la línia, es deselecciona el RadioButton i es tanca la classe. Si s'ha seleccionat un element de la llista primer de tot s'obre un diàleg amb la classe TipoEnlace, la qual es menciona més endavant i segons com es resolgui aquest quadre, es truca a la classe CrearEnlace.

També es consideren totes les opcions possibles a les següents línies de codi, però funcionen de la mateixa manera així que no cal explicar-les.

#### 3.7.12.2.2 Procediment Botó Crear Enllaç

Es truca quan es necessita crear un enllaç manual perquè no s'ha produït de forma automàtica. Cal dir que no es podrà revertir aquest procés. En aquesta funció es recorre tota la taula d'enllaços generada i es mira la coincidència de la primera columna amb el NumericUpDown de la classe (línia 36). Si n'hi ha coincidència es prova de guardar l'element de la llista en una variable i en funció a la variable pública



de la classe TipoEnlace adient es situarà en una columna de la taula o en altra. Per mantenir constància d'aquest enllaç manual, finalment es guarda en la variable pública GuardarEnlaces tota la informació necessària aprofitable quan s'actualitzen els enllaços manuals al subapartat 3.7.1.2.16. El codi que s'ha explicat es pot trobar a la figura següent:

```

35 Private Sub CrearEnlace()
36     For Each Fila As DataGridViewRow In EnlacesForm.DataGridView1.Rows 'Se recorre la tabla de enlaces
37         If Me.NumericUpDown1.Value = Fila.Cells(0).Value.ToString Then 'Si coincide el número de la etapa con el valor
seleccionado por el usuario
38             Dim leerSelección As String 'Variable que lee la selección del ítem seleccionado
39             Try
40                 leerSelección = ListBox1.SelectedItem.ToString() 'Se guarda la selección
41             Catch
42                 leerSelección = ""
43             End Try
44             If leerSelección <> "" Then
45                 If TipoEnlace.RespuestaAnterior = True Then 'En función de la respuesta del usuario:
46                     Fila.Cells(7).Value = leerSelección 'Se coloca como enlace anterior en la tabla de enlaces.
47                     GuardarEnlaces &= "anterior|" + Fila.Cells(0).RowIndex.ToString + "|" + leerSelección + "|" 'Se guarda
para mantenerse en la tabla
48                 ElseIf TipoEnlace.RespuestaSiguiente = True Then
49                     Fila.Cells(6).Value = leerSelección 'Se coloca como enlace siguiente en la tabla de enlaces
50                     GuardarEnlaces &= "siguiente|" + Fila.Cells(0).RowIndex.ToString + "|" + leerSelección + "|" 'Se
guarda para mantenerse en la tabla
51                 End If
52             End If
53         Next Fila
54     End Sub

```

Figura 3.7.12.2.2.1. Captura d'imatge part del codi annex A13. (Font: Pablo Mañas)

### 3.7.12.2.3 Procediment Botó Construir línia

En funció als paràmetres que rep la funció de construir línia, es generarà la línia amb les característiques concretes que es guardarà a la llista adequada i a la llista de coordenades adequades. Funciona com moltes funcions ja explicades i el codi és el següent:

```

63 Public Sub ConstruirLínea(ruta As String, listaCoords As List(Of Point), listaLinea As List(Of PictureBox), ancho As
Integer, altura As Integer)
64     Línea = New PictureBox With {.Image = Image.FromFile(ruta), .SizeMode = PictureBoxSizeMode.StretchImage, .BackColor =
Color.Transparent, .Width = ancho, .Height = altura, .Left = 300, .Top = 100, .Cursor = Cursors.SizeAll, .Tag = False}
65     Define características de la línea
66     MenuPrincipalForm.Panel1.Controls.Add(Línea) 'Se añade la línea al panel
67     listaLinea.Add(Línea) 'Se añade la línea a la lista adecuada
68     listaCoords.Add(Línea.Location) 'Se guarda la posición de la línea en las coordenadas adecuadas
69     ListBox1.Items.Clear() 'Se borran los elementos de la lista
End Sub

```

Figura 3.7.12.2.3.1. Captura d'imatge part del codi annex A13. (Font: Pablo Mañas)

En aquest cas cal dir que la funció que s'executa quan es polsa el botó de Cancel·lar, a més de tancar la classe, també borra els elements de la ListBox amb un Clear.

## 3.7.13. Simulació

Continuant amb el desenvolupament de les classes i la seva explicació, és el torn de la classe simulació una de les més complexes pel que fa al codi. Ja s'ha explicat com es genera la taula d'entrades, sortides i temporitzadors al menú principal, però no com interactua aquesta amb la simulació. Així doncs, en aquest subapartat es tracta d'explicar tot el que envolta a aquesta classe.

### 3.7.13.1. Variables

En primer lloc la classe Simulació declara una variable pública de tipus booleana que determina si la simulació està activa o no. Després s'involucren una sèrie de llistes de booleans i expressions utilitzades per activar les transicions determinades. Més endavant també s'observen 6 llistes que controlen els

temporitzadors actius. Finalment es generen variables auxiliars que copien les dades de les llistes de les etiquetes de simulació ja que per alguna raó aquestes últimes perden el valor que contenen quan un temporitzador acaba i serveixen com a base de dades.

### 3.7.13.2. Procediments i funcions

Al tractar-se d'una classe important, a continuació es trobarà d'explicar el més detalladament com funcionen els mètodes emprats dintre de la classe.

#### 3.7.13.2.1 Procediment Inici Entrades Fals

La primera funció d'aquesta classe a la que es truca des de el menú principal quan es fa clic en Simular s'anomena InicioEntradasFalse. Consisteix en convertir totes les entrades del sistema mostrades a la taula en expressions booleanes falses però escrites en cadena. El codi que es genera per fer-ho és el següent:

```

30 Public Sub InicioEntradasFalse()
31     SimulacionActivada = True 'Se inicia la simulación
32     listaExpresiones.Clear() 'Se limpia la lista de expresiones
33     listaBooleanaExpresiones.Clear() 'Se limpia la lista booleana de expresiones
34     For ind = 0 To MenuPrincipalForm.listaLabsSimulacion.LongCount - 1 'Se recorre la lista de etiquetas de la simulación
35     listaExpresiones.Add(" " + MenuPrincipalForm.listaLabsSimulacion(ind).Text + " ") 'Se añaden espacios a los lados
36     de cada una y se guarda en la lista
37     AcomodarListaExpresiones(ind) 'Se reemplazan los signos por sus funciones logicas y se añaden mas espacios
38     For Each Fila As DataGridViewRow In DataGridView1.Rows 'Se recorre la tabla de simulación (entradas/salidas)
39     Try
40     Dim Entradas = Fila.Cells(0).Value.ToString 'Se guarda la entrada en una variable
41     Dim Espacio = InStr(Fila.Cells(0).Value.ToString, " ") 'Se encuentra el espacio
42     Entradas = Entradas.Substring(0, Espacio - 1) 'Se extrae la entrada, o sea, el nombre de la variable de la
43     entrada, por ejemplo (FC1 - 10.0 -> FC1)
44     Entradas = " " + Entradas + " " 'Se añaden espacios a los lados
45     If InStr(1, listaExpresiones(ind), Entradas, 1) > 0 Then 'Si coincide la el nombre de la entrada con el
46     nombre del elemento de la lista de expresiones
47     listaExpresiones(ind) = listaExpresiones(ind).Replace(Entradas, " False ") 'Se reemplaza el elemento
48     de la lista por "False" ya que inicialmente la variable esta en "False"
49     End If
50     Catch
51     End Try
52     Next Fila
53     Dim Resultado = Eval(listaExpresiones(ind)) 'Evaluó el resultado de la lista de expresiones
54     listaBooleanaExpresiones.Add(Resultado) 'Añado el resultado a la lista de booleanos
55     Next ind
56     For ind = 0 To MenuPrincipalForm.listaLabsTimerSimulacion.LongCount - 1 'Se recorre la lista de los timers
57     NumeroTimer.Add(0) 'Se ponen a 0 todas las listas
58     ContadoresTimer.Add(0)
59     ContadorTimerAcabado.Add(0)
60     Next
61     For ind = 0 To MenuPrincipalForm.listaLabsTimerAsociadoSimulacion.LongCount - 1 'Se recorre la lista de los timers
62     asociados
63     NumeroTimerAsociado.Add(0) 'Se ponen a 0 todas las listas
64     ContadoresTimerAsociado.Add(0)
65     ContadorTimerAsociadoAcabado.Add(0)
66     Next
67     ObtenerListaBooleana() 'Se obtiene la lista booleana
68     ActualizarSimulacion() 'Se actualiza la simulación
69 End Sub

```

Figura 3.7.13.2.1.1. Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

Tal i com s'observa a la imatge anterior en un principi es manipula la variable SimulacionActivada per posar-la a true ja que s'ha iniciat la simulació. Després es netegen les llistes d'expressions i les llistes d'expressions booleanes. Es fa un recorregut per tota la llista d'etiquetes de la simulació (línia 34) i s'afegeixen a la variable llista d'expressions amb espais als dos costats. Més endavant es truca a la funció AcomodarListaExpresiones amb la fi de reemplaçar els símbols per funcions lògiques. A la línia de codi següent es recorre la taula de simulació i s'extrau el nom de la variable de cada entrada. Si aquest nom coincideix amb el de la llista d'expressions, es substitueix aquest per la cadena " False " a la llista d'expressions. Després s'avalua l'expressió amb la funció Eval i s'afegeix el resultat a la llista

booleana d'expressions. A les línies 52 i 57 es recorren els temporitzadors per tal de restaurar els comptadors d'aquests a 0. Finalment es truca a la funció `ObtenerListaBooleana` i després a `ActualizarSimulacion`.

### 3.7.13.2.2 Procediment Acomodar Llista d'expressions

Aquest mètode serveix per convertir les expressions en signe en expressions lògiques. El codi és el següent:

```

66      'Función que reemplaza signos por funciones lógicas:
67      Private Sub AcomodarListaExpresiones(ind As Integer)
68          listaExpresiones(ind) = listaExpresiones(ind).Replace("-", " And ")
69          listaExpresiones(ind) = listaExpresiones(ind).Replace("+", " Or ")
70          listaExpresiones(ind) = listaExpresiones(ind).Replace("/", " Not ")
71          listaExpresiones(ind) = listaExpresiones(ind).Replace(")", " ")
72          listaExpresiones(ind) = listaExpresiones(ind).Replace("(", " ( ")
73      End Sub
    
```

**Figura 3.7.13.2.2.1.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

Ja que només s'utilitza per la llista d'expressions només necessita un enter per anar recorrent cada element de la llista. El que fa es reemplaçar la cadena amb signes per l'expressió lògica, però sense deixar de ser del tipus cadena.

### 3.7.13.2.3 Procediment Botó Activar Entrada

La taula d'entrades i sortides té dos botons un per activar l'entrada i altre per desactivar-la. En aquesta ocasió es parla del primer botó. El codi que s'ha creat i que s'executa quan es polsa el botó és el següent:

```

75      'Botón activar entrada:
76      Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
77          listaExpresiones.Clear() 'Limpia la lista de expresiones
78          listaBooleanaExpresiones.Clear() 'Limpia la lista de expresiones booleanas
79          Dim NumeroSeleccion As Integer 'Define un número con la seleccion manual de la tabla
80          NumeroSeleccion = Me.DataGridView1.GetCellCount(DataGridViewElementStates.Selected) 'Se guarda el numero de las
celulas seleccionadas
81          For ind = 0 To MenuPrincipalForm.listaLabsSimulacion.LongCount - 1 'Se recorre la lista de etiquetas
82              listaExpresiones.Add(" " + MenuPrincipalForm.listaLabsSimulacion(ind).Text + " ") 'Se añaden espacios y se
guarda en la liste de expresiones
83              AcomodarListaExpresiones(ind) 'Se reemplazan los signos por sus funciones logicas y se añaden mas espacios
84              For ind1 = 0 To NumeroSeleccion - 1 'Se recorre por cada una de las selecciones
85                  If Me.DataGridView1.SelectedCells Is Nothing Then
86                      MessageBox.Show("No se ha seleccionado ninguna entrada") 'Si no se selecciona nada se muestra mensaje
87                  Else
88                      If Me.DataGridView1.SelectedCells(ind1).ColumnIndex = 1 Then
89                          MessageBox.Show("No se pueden activar las salidas") 'Se se intenta activar una salida se muestra
mensaje
90                      Else
91                          Me.DataGridView1.SelectedCells(ind1).Style.BackColor = Color.Turquoise 'Si esta seleccionada una
entrada, se pone el color turquesa de fondo
92                      End If
93                  End If
94                  Next ind1
95              Next ind
96              ObtenerListaExpresiones(ind) 'Se obtiene la lista de expresiones
97              ObtenerListaBooleana() 'Se obtiene la lista booleana
98              ActualizarSimulacion() 'Se actualiza la simulación
99      End Sub
    
```

**Figura 3.7.13.2.3.1.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

En primer lloc es netegen les llistes adjacents i es defineix un número que guarda les cel·les seleccionades per l'usuari. Després es recorre la llista d'etiquetes les quals s'acomoden de la mateixa forma que al subapartat 3.7.13.2.1. Si no existeix cap selecció (línia 85) no es fa res i s'adverteix, si s'ha seleccionat la columna de les sortides tampoc es permet i s'avisava (línia 89). En cas que hi hagi hagut selecció, es

pinten de color turquesa les cel·les seleccionades. Més endavant es truca a la funció `ObtenerListaExpresiones` i fora del bucle a `ObtenerListaBooleana` i `ActualizarSimulacion`.

### 3.7.13.2.4 Procediment Botó Desactivar Entrada

En el cas del segon botó, el funcionament és el mateix. El codí corresponent és el següent.

```

101 'Botón desactivar entrada (funcionamiento igual que el anterior pero coloreando el fondo de la entrada en blanco al
desactivar):
102 Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
103     listaExpresiones.Clear()
104     listaBooleanaExpresiones.Clear()
105     Dim NumeroSeleccion As Integer
106     NumeroSeleccion = Me.DataGridView1.GetCellCount(DataGridViewElementStates.Selected)
107     For ind = 0 To MenuPrincipalForm.listaLabsSimulacion.LongCount - 1
108         listaExpresiones.Add(" " + MenuPrincipalForm.listaLabsSimulacion(ind).Text + " ")
109         AcomodarListaExpresiones(ind)
110         For ind1 = 0 To NumeroSeleccion - 1
111             If Me.DataGridView1.SelectedCells Is Nothing Then
112                 MessageBox.Show("No se ha seleccionado ninguna entrada")
113             Else
114                 If Me.DataGridView1.SelectedCells(ind1).ColumnIndex = 1 Then
115                     MessageBox.Show("No se pueden desactivar las salidas")
116                 Else
117                     Me.DataGridView1.SelectedCells(ind1).Style.BackColor = Color.White
118                 End If
119             End If
120         Next ind1
121         ObtenerListaExpresiones(ind)
122     Next ind
123     ObtenerListaBooleana()
124     ActualizarSimulacion()
125 End Sub

```

Figura 3.7.13.2.4.1. Captura d'imatge part del codí annex A14. (Font: Pablo Mañas)

L'única diferència amb la funció anterior es que el color de fons de les cel·les seleccionades es pinta de blanc. De fet es podria haver fet una funció similar que tingués el contingut i que amb un paràmetre cambiés de blanc o turquesa en funció del botó.

### 3.7.13.2.5 Procediment Obtenir Llista d'expressions

En aquest subapartat s'explica una de les funcions que s'ha mencionat anteriorment en diverses ocasions. El codí corresponent a aquest mètode es disposa a continuació:

```

127 Private Sub ObtenerListaExpresiones(ind As Integer)
128     For Each Fila As DataGridViewRow In DataGridView1.Rows 'Se recorre la tabla de entradas/salidas
129         If Fila.Cells(0).Style.BackColor = Color.Turquoise Then 'Si hay alguna entrada en fondo color turquesa
130             Try
131                 Dim EntradasSeleccionadas = Fila.Cells(0).Value.ToString 'Define entradas seleccionadas
132                 Dim Espacio = InStr(EntradasSeleccionadas, " ")
133                 EntradasSeleccionadas = EntradasSeleccionadas.Substring(0, Espacio - 1) 'Extrae la entrada seleccionada
134                 EntradasSeleccionadas = " " + EntradasSeleccionadas + " "
135                 If InStr(1, listaExpresiones(ind), EntradasSeleccionadas, 1) > 0 Then 'Si hay coincidencia con la lista de
expresiones
136                     listaExpresiones(ind) = listaExpresiones(ind).Replace(EntradasSeleccionadas, " True ") 'Se reemplaza
el elemento de la lista por "True"
137                 End If
138             Catch
139             End Try
140         Else
141             Try
142                 Dim EntradasNoSeleccionadas = Fila.Cells(0).Value.ToString 'Define entradas seleccionadas
143                 Dim Espacio2 = InStr(Fila.Cells(0).Value.ToString, " ")
144                 EntradasNoSeleccionadas = EntradasNoSeleccionadas.Substring(0, Espacio2 - 1) 'Extrae la entrada
seleccionada
145                 EntradasNoSeleccionadas = " " + EntradasNoSeleccionadas + " "
146                 If InStr(1, listaExpresiones(ind), EntradasNoSeleccionadas, 1) > 0 Then
147                     listaExpresiones(ind) = listaExpresiones(ind).Replace(EntradasNoSeleccionadas, " False ") 'Se
reemplaza el elemento de la lista por "False"
148                 End If
149             Catch
150             End Try
151         End If
152     Next Fila
153     Dim Resultado = Eval(listaExpresiones(ind)) 'Se evalua la expresion
154     listaBooleanaExpresiones.Add(Resultado) 'Se añade el resultado a la lista de expresiones booleana
155 End Sub

```

Figura 3.7.13.2.5.1. Captura d'imatge part del codí annex A14. (Font: Pablo Mañas)

Com s’aprecia a la primera línia de codi dintre de la funció de la figura anterior, es recorre tota la llista de la taula. En cas que existeixi una entrada activada (línia 129) s’observa si n’hi ha coincidència amb la llista d’expressions creada prèviament. En cas que sí, aquell ítem de la llista d’expressions es substitueix per la cadena “ True “ ja que significaria que s’ha activat. Per aquelles entrades que tenen el fons blanc es fa el mateix però la substitució es per “ False “. Finalment s’evalua l’ítem de la llista d’expressions i s’afegeix el resultat a la llista booleana d’expressions.

### 3.7.13.2.6 Procediment Obtenir Llista Booleana

Altre de les funcions també mencionades anteriorment és que permet obtenir la llista booleana. El codi a analitzar és el següent:

```

157     'Función que obtiene la lista booleana:
158     Private Sub ObtenerListaBooleana()
159         For ind = 0 To listaBooleanaExpresiones.LongCount - 1 'Se recorre la lista booleana de expresiones
160             If listaBooleanaExpresiones(ind) = "True" Then 'Si el resultado en la lista es "true"
161                 MenuPrincipalForm.listaLabsSimulacion(ind).BackColor = Color.Turquoise 'El color de fondo de la etiqueta que
coincide se pone de color turquesa
162                 AuxiliarListaLabsSimulacion(ind) = MenuPrincipalForm.listaLabsSimulacion(ind) 'Se guarda en el auxiliar el
cambio
163             Else
164                 MenuPrincipalForm.listaLabsSimulacion(ind).BackColor = Color.Transparent 'Sino el color de fondo es
transparente ya que no se valida la transición
165                 AuxiliarListaLabsSimulacion(ind) = MenuPrincipalForm.listaLabsSimulacion(ind) 'Se guarda en el auxiliar el
cambio
166             End If
167         Next ind
168     End Sub

```

Figura 3.7.13.2.6.1. Captura d’imatge part del codi annex A14. (Font: Pablo Mañas)

En principi es recorre tota la llista booleana d’expressions que s’ha anat creant i en cas que es trobi un “True”, es posa el color de fons blau turquesa de l’etiqueta de la transició, convergència o divergència que correspon a l’expressió. En cas contrari el color de fons es deixa transparent o sigui blanc perquè no s’ha validat l’expressió. En ambdós casos es guarden els auxiliars per si un cas.

### 3.7.13.2.7 Funció Eval

Aquesta funció extreta de l’enllaç [8] de la bibliografia permet evaluar una expressió boleana escrita en cadena i ser retornada també en aquest tipus. D’aquesta manera es pot crear la llista d’expressions booleanes de la que s’ha parlat en nombroses ocasions. El codi extret és el següent:

```

170 'Función extraída de https://social.msdn.microsoft.com/Forums/es-ES/8cda2e72-2dc1-44ce-9bb4-b6083b2e586c/necesito-guardar-
operadores-lgicos-en-variable-para-usarlos-en-operacin-de-clculo?forum=vbes [7] para obtener un booleano a partir de un
string.
171 Private Function Eval(ByVal Expresion As String) As Object
172 Dim vbcp As New Microsoft.VisualBasic.VBCodeProvider
173 Dim vbc As System.CodeDom.Compiler.ICodeCompiler = vbcp.CreateCompiler
174 Dim cpar As New System.CodeDom.Compiler.CompilerParameters
175 Dim res As System.CodeDom.Compiler.CompilerResults
176 cpar.GenerateExecutable = False 'Generar DLL
177 cpar.GenerateInMemory = True 'Generar en memoria
178 cpar.IncludeDebugInformation = True
179 'Agregar referencias
180 cpar.ReferencedAssemblies.Add("Microsoft.VisualBasic.dll")
181 'Compilar
182 res = vbc.CompileAssemblyFromSource(cpar,
183 "Imports Microsoft.VisualBasic" & vbCrLf &
184 "Namespace MiNamespace" & vbCrLf &
185 "Public Class MiClase" & vbCrLf &
186 "Public Shared Function Eval() As Object" & vbCrLf &
187 "Return " & Expresion & vbCrLf &
188 "End Function" & vbCrLf &
189 "End Class" & vbCrLf &
190 "End Namespace")
191 If res.Errors.Count = 0 Then
192 'Obtengo el Type de la clase recién compilada
193 Dim miClase As System.Type
194 miClase = res.CompiledAssembly.GetType("MiNamespace.MiClase")
195 'Obtengo el metodo Eval de la clase
196 Dim funcEval As System.Reflection.MethodInfo
197 funcEval = miClase.GetMethod("Eval")
198 'Ejecuto la función Eval recién creada
199 Return funcEval.Invoke(Nothing, Nothing)
200 Else
201 Return res.Errors(0).ErrorText
202 End If
203 End Function

```

Figura 3.7.13.2.7.1. Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

El funcionament del codi de la imatge anterior és una mica complex. Si s'observa la capçalera es pot veure que requereix un paràmetre del tipus String i que retornarà un Objecte. En sí la funció agafa l'expressió que li arriba per argument i la compila utilitzant la consola. Si no troba cap error, retorna allò que ha compilat amb el mateix tipus que observa a la crida. Si ni ha un error retorna l'error en text.

### 3.7.13.2.8 Procediment Recorregut temporitzador

Aquest procediment es truca dins de la funció ActulizarSimulacion i consisteix en guardar els índexs de les files de la taula d'enllaços que coincideixin amb les transicions de temporitzadors actives. Es va crear per estalviar línies de codi ja que sinó s'hauria de repetir aquesta funció 4 cops, una per cada tipus de transició de temporitzador. El codi desenvolupat és el següent:

```

205 'Función para recorrer las transiciones de los temporizadores:
206 Private Sub RecorridoTimer(lista As List(Of Integer), ByRef GuardadoFila As List(Of Integer), ByRef ListaEtapasAnteriores As
List(Of String), ByRef Convergencia As List(Of Boolean))
207 For ind = 0 To lista.LongCount - 1 'Se recorre la lista
208 If lista(ind).BackColor = Color.Turquoise Then 'Si el elemento es de color turquesa la transicion se ha cumplido:
209 For Each Fila As DataGridViewRow In EnlacesForm.DataGridView1.Rows 'Se recorre la lista de enlaces
210 Dim EnlacesAnteriores = Fila.Cells(7).Value.ToString 'Se guardan los enlaces anteriores
211 ObtenerSubstring(EnlacesAnteriores, " ")
212 Dim ReconstruirEnlacesAnteriores() As String
213 ReconstruirEnlacesAnteriores = EnlacesAnteriores.Split(",")
214 For ind4 = 0 To ReconstruirEnlacesAnteriores.LongCount - 1 'Se recorren todos los enlaces anteriores
215 If ReconstruirEnlacesAnteriores(ind4) = lista(ind).Text Then 'Si coincide el elemento de color
turquesa con el enlace anterior:
216 GuardadoFila.Add(Fila.Cells(0).RowIndex) 'Se guarda el indice de la fila coincidente
217 ListaEtapasAnteriores.Add(Fila.Cells(8).Value.ToString) 'Se guardan las etapas o etapa coincidente
218 Convergencia.Add(False) 'No es una convergencia
219 End If
220 Next ind4
221 Next Fila
222 End If
223 Next ind
224 End Sub

```

Figura 3.7.13.2.8.1. Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

Per referència modificarà els paràmetres amb les variables de GuardadoFila, ListaEtapasAnteriores i Convergencia. També necessita una llista d'etiquetes per funcionar. En un principi es recorre aquesta llista i si es troba algun element amb el color de fons de color turquesa (transició activa) es comença a

recórrer tota la llista d'enllaços. En una variable es guarden els enllaços anteriors de la taula que en cas que hi hagi múltiples es separen per comes. Es recorre aquesta separació i si n'hi ha coincidència amb l'element de color turquesa i l'enllaç anterior, es guarda la fila en una llista (línia 216). També es guarda l'etapa anterior o etapes on ha aparegut la coincidència. Com no és una convergència, a la llista s'afegeix un False.

### 3.7.13.2.9 Procediment Actualitzar Simulació

Aquesta funció és llarga per la seva complexitat i, per tant, s'ha decidit explicar-la per parts. Així doncs, es comença per la primera part:

```

226     Public Sub ActualizarSimulacion()
227         'Definición variables necesarias:
228         Dim GuardadoFila As New List(Of Integer)
229         Dim GuardadoFila2 As New List(Of Integer)
230         Dim Convergencia As New List(Of Boolean)
231         Dim ConvergenciaString As String = ""
232         Dim ConvergenciaAdecuada As Boolean
233         Dim ListaEtapaActivaAnterior As New List(Of Integer)
234         Dim ListaEtapasAnteriores As New List(Of String)
235         Dim ContadorConvergencia = 0
    
```

**Figura 3.7.13.2.9.1.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

En primer lloc es defineixen una sèrie de variables. Les dues primeres son llistes de nombres enters que guarden els índexs de les files de les taules d'enllaços que interessin. La tercera és una llista de booleà que permet detectar quan hi ha una convergència. La quarta i la quinta serveixen per tractar les convergències. Les dues següents són llistes que emmagatzemen les etapes actives anteriors i les etapes anteriors i per últim s'inicialitza un comptador de les convergències a 0. La segona part és la següent:

```

236
237     For ind = 0 To MenuPrincipalForm.listaLabsTransTimerSimulacion.LongCount - 1 'Recorre la lista de transiciones de
timers
238         If ContadorTimerAcabado(ind) = 1 Then 'Si un timer ha terminado
239             MenuPrincipalForm.listaLabsTransTimerSimulacion(NumeroTimer(ind)).BackColor = Color.Turquoise 'El color de
ese timer es turquesa
240             MenuPrincipalForm.listaLabsTransTimerNegadoSimulacion(NumeroTimer(ind)).BackColor = Color.Transparent 'El
timer negado correspondiente se vuelve blanco
241             ContadorTimerAcabado(ind) = 0
242             ContadoresTimer(ind) = 0
243         End If
244     Next
245
246     For ind = 0 To MenuPrincipalForm.listaLabsTransTimerAsociadoSimulacion.LongCount - 1 'Lo mismo para las transiciones
de timers asociados
247         If ContadorTimerAsociadoAcabado(ind) = 1 Then
248             MenuPrincipalForm.listaLabsTransTimerAsociadoSimulacion(NumeroTimerAsociado(ind)).BackColor = Color.Turquoise
249             MenuPrincipalForm.listaLabsTransTimerAsociadoNegadoSimulacion(NumeroTimerAsociado(ind)).BackColor =
Color.Transparent
250             ContadorTimerAsociadoAcabado(ind) = 0
251             ContadoresTimerAsociado(ind) = 0
252         End If
253     Next
    
```

**Figura 3.7.13.2.9.2.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

En aquest punt es recorre la llista de transicions dels temporitzadors, en cas que es detecti un comptador a 1 voldrà dir que ha acabat i per tant, caldrà modificar les transicions dels temporitzadors. Després es reinicien els comptadors a 0 per a què puguin tornar a poder crear-se nous temporitzadors. El mateix succeeix amb la llista de transicions dels temporitzadors associats. Continuant amb l'ordre es passa al següent fragment de codi:

```

255     For ind = 0 To MenuPrincipalForm.listaLabsSimulacion.LongCount - 1 'Se recorren todas las transiciones
256     If MenuPrincipalForm.listaLabsSimulacion(ind).BackColor = Color.Turquoise Then 'Si el elemento es de color
turquesa:
257     For Each Fila As DataGridViewRow In EnlacesForm.DataGridView1.Rows 'Se recorre toda la lista de enlaces:
258     Dim EnlacesAnteriores = Fila.Cells(7).Value.ToString 'Se guardan los enlaces anteriores
259     ObtenerSubstring(EnlacesAnteriores, " ")
260     Dim ReconstruirEnlacesAnteriores() As String
261     Dim Convergencias = Fila.Cells(7).Value.ToString
262     Dim ConvergenciaEntera = Fila.Cells(7).Value.ToString
263     Dim Espacio = InStr(Convergencias, " ")
264     Try
265     Convergencias = Convergencias.Substring(0, Espacio - 1)
266     Catch
267     End Try
268     If Convergencias = "Convergencia" Then 'Si el enlace anterior tiene el tag de convergencia
269     ObtenerSubstring(ConvergenciaEntera, " ")
270     Convergencias = Convergencias.Replace(" ", "")
271     If ConvergenciaEntera = MenuPrincipalForm.listaLabsSimulacion(ind).Text Then 'Si coincide el elemento
de color turquesa con el enlace de convergencia:
272     GuardadoFila.Add(Fila.Cells(0).RowIndex) 'Se guarda el indice de la fila coincidente
273     Convergencia.Add(True) 'Es una convergencia
274     End If
275     Else 'Si no lo tiene:
276     ReconstruirEnlacesAnteriores = EnlacesAnteriores.Split(",") 'Separa todos los enlaces cuando hay
múltiples
277     For ind4 = 0 To ReconstruirEnlacesAnteriores.LongCount - 1 'Se recorren los enlaces
278     ReconstruirEnlacesAnteriores(ind4) = ReconstruirEnlacesAnteriores(ind4).Replace(" ", "") 'Se
eliminan espacios
279     If ReconstruirEnlacesAnteriores(ind4) = MenuPrincipalForm.listaLabsSimulacion(ind).Text Then
'Si coincide el elemento de color turquesa con el enlace:
280     GuardadoFila.Add(Fila.Cells(0).RowIndex) 'Se guarda el indice de la fila coincidente
281     ListaEtapasAnteriores.Add(Fila.Cells(8).Value.ToString) 'Se guardan las etapas o etapa
coincidente
282     Convergencia.Add(False) 'No es una convergencia
283     End If
284     Next ind4
285     End If
286     Next Fila
287     End If
288     Next ind
289
'Se recorren todas las listas de timers:
290     RecorridoTimer(MenuPrincipalForm.listaLabsTransTimerSimulacion, GuardadoFila, ListaEtapasAnteriores, Convergencia)
291     RecorridoTimer(MenuPrincipalForm.listaLabsTransTimerAsociadoSimulacion, GuardadoFila, ListaEtapasAnteriores,
Convergencia)
292     RecorridoTimer(MenuPrincipalForm.listaLabsTransTimerNegadoSimulacion, GuardadoFila, ListaEtapasAnteriores,
Convergencia)
293     RecorridoTimer(MenuPrincipalForm.listaLabsTransTimerAsociadoNegadoSimulacion, GuardadoFila, ListaEtapasAnteriores,
Convergencia)

```

**Figura 3.7.13.2.9.3.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

En aquest instant es fa un bucle amb les etiquetes de la simulació i si alguna està activa es recorre la llista d'enllaços per cercar les coincidències. Les convergències es tracten a part ja que si n'hi ha una convergència activa, caldrà que totes les etapes anteriors estiguin actives per poder activar la següent etapa. Quan es troba la coincidència es guarda l'índex on hi ha la coincidència i el valor de l'última columna on es troben les etapes anteriors coincidents. Més endavant es truca 4 cops a la funció RecorridoTimer descrita anteriorment. Si es continua amb el codi s'observa el següent:

```

296     For ind = 0 To MenuPrincipalForm.listaEtapasSimulacion.LongCount - 1 'Se recorren las etapas de la simulacion
297     If MenuPrincipalForm.listaEtapasSimulacion(ind).Tag = True Then 'Si alguna tiene está activa:
298     ListaEtapaActivaAnterior.Add(ind) 'Se añade a la lista de etapas activas
299     End If
300     Next ind
301
302     For ind = 0 To ListaEtapaActivaAnterior.LongCount - 1 'Se recorre la lista de etapas activas
303     ConvergenciaString += ListaEtapaActivaAnterior(ind).ToString + ", " 'Se añaden a una string
304     Next ind
305     Try
306     ConvergenciaString = ConvergenciaString.Substring(0, Len(ConvergenciaString) - 2) 'Se intenta borrar el último
espacio y coma.
307     Catch
308     End Try

```

**Figura 3.7.13.2.9.4.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

En aquest punt s'omple la llista d'etapes actives valorant l'atribut tag de la llista d'etapes de simulació. Després amb aquesta mateixa llista s'actualitza la cadena que podria tenir la convergència i s'elimina l'últim espai i coma de la cadena. Si es continua executant el codi de dalt abaix s'observa el següent fragment de codi:



```

310     For ind = 0 To ListaEtapasActivaAnterior.LongCount - 1 'Se recorre la lista de etapas activas
311     For Each Fila As DataGridViewRow In EnlacesForm.DataGridView1.Rows 'Se recorre la tabla de enlaces
312     Dim EtapasAnteriores = Fila.Cells(8).Value.ToString 'Se guardan las etapas anteriores
313     ObtenerSubstring(EtapasAnteriores, " ")
314     Dim ReconstruirEtapasAnteriores() As String
315     ReconstruirEtapasAnteriores = EtapasAnteriores.Split(",") 'Se separan las etapas anteriores en caso de
múltiples
316     For ind4 = 0 To ReconstruirEtapasAnteriores.LongCount - 1 'Se recorren las etapas anteriores
317     ReconstruirEtapasAnteriores(ind4) = ReconstruirEtapasAnteriores(ind4).Replace(" ", "") 'Se eliminan los
espacios
318     If ReconstruirEtapasAnteriores(ind4) = ListaEtapasActivaAnterior(ind).ToString Then 'Si coincide la etapa
anterior con la etapa activa
319     GuardadoFila2.Add(Fila.Cells(0).RowIndex) 'Se guarda el indice de la fila coincidente
320     End If
321     Next ind4
322     Next Fila
323     For ind1 = 0 To Convergencia.LongCount - 1 'Se recorre la lista de convergencias
324     If Convergencia(ind1) = True Then 'Si existe convergencia:
325     Dim EtapasAnteriores = EnlacesForm.DataGridView1.Rows(GuardadoFila(ind1)).Cells(8).Value.ToString 'Se
obtienen las etapas anteriores de la convergencia
326     ObtenerSubstring(EtapasAnteriores, " ")
327     Dim ReconstruirEtapasAnteriores() As String
328     Dim listaNumero As New List(Of Integer)
329     Dim ConvergenciaCadena As String = ""
330     ReconstruirEtapasAnteriores = EtapasAnteriores.Split(",") 'Se separan las etapas
331     For ind4 = 0 To ReconstruirEtapasAnteriores.LongCount - 1
332     ReconstruirEtapasAnteriores(ind4) = ReconstruirEtapasAnteriores(ind4).Replace(" ", "") 'Se eliminan
los espacios
333     listaNumero.Add(ReconstruirEtapasAnteriores(ind4)) 'Se guardan los numeros de las etapas en una lista
334     Next ind4
335     listaNumero.Sort() 'Se ordenan los números de las etapas
336     For ind5 = 0 To listaNumero.LongCount - 1
337     ConvergenciaCadena += listaNumero(ind5).ToString + ", "
338     Next ind5
339     ConvergenciaCadena = ConvergenciaCadena.Substring(0, Len(ConvergenciaCadena) - 2) 'Se elimina la coma y el
espacio final
340     If ConvergenciaCadena = ConvergenciaString Then 'Si coinciden las etapas activas anteriores con las etapas
anteriores a la convergencia
341     ConvergenciaAdecuada = True 'La convergencia es adecuada
342     End If
343     End If
344     Next ind1
345     Next ind

```

Figura 3.7.13.2.9.5. Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

En aquest instant es recorre la llista d'etapes actives juntament amb la taula d'enllaços (línies 310 i 311) i es cerquen les files de les taules on hi ha coincidència amb les etapes actives anteriors. Per exemple si només es té una etapa activa anterior que és la número 8, es cerquen a la taula d'enllaços a la última columna de cada fila si n'hi ha coincidència amb aquest número. Quan es troba coincidència es guarda la fila en altre variable. Dintre del bucle generat a la primera línia de codi de la figura anterior, es recorre la llista booleana de convergències la qual cercava si n'hi havia coincidència de convergència en algun dels enllaços. Si existeix alguna convergència (línia 324) s'obtenen les etapes anteriors on hi va haver coincidència amb la llista d'enllaços. S'ordenen aquests números i s'arreglen de la mateixa manera que s'ha fet al fragment anterior amb la variable ConvergenciaString. Finalment, en cas que hi hagi coincidència amb aquesta cadena voldrà dir que la convergència és adequada ja que totes les etapes anteriors de la convergència estan actives i la pròpia convergència també. El codi següent permet la continuació:

```

347     For ind = 0 To GuardadoFila.LongCount - 1 'Se recorren las listas de guardado de filas coincidentes
348     For ind2 = 0 To GuardadoFila2.LongCount - 1
349     If Convergencia(ind) = False Then 'Si no hay convergencia
350     If GuardadoFila(ind) = GuardadoFila2(ind2) Then 'Si la fila o filas donde está la transicion en turquesa
coincide con la fila donde se encuentran las etapas activas anteriores
351     MenuPrincipalForm.listaEtapasSimulacion(GuardadoFila2(ind2)).Tag = True 'Se activa la etapa
correspondiente
352     AuxiliarListaEtapasSimulacion(GuardadoFila2(ind2)) =
MenuPrincipalForm.listaEtapasSimulacion(GuardadoFila2(ind2)) 'Se guarda en la auxiliar
353     End If
354     Else
355     If ConvergenciaAdecuada = True Then 'Si hay convergencia y es adecuada
356     MenuPrincipalForm.listaEtapasSimulacion(GuardadoFila(ind)).Tag = True 'Se activa la etapa anterior a
la convergencia
357     AuxiliarListaEtapasSimulacion(GuardadoFila(ind)) =
MenuPrincipalForm.listaEtapasSimulacion(GuardadoFila(ind)) 'Se guarda en la auxiliar
358     Dim EtapasAnterioresConvergencia() As String
359     EtapasAnterioresConvergencia = ConvergenciaString.Split(",")
360     For ind3 = 0 To EtapasAnterioresConvergencia.LongCount - 1
361     MenuPrincipalForm.listaEtapasSimulacion(CInt(EtapasAnterioresConvergencia(ind3))).Tag = False 'Se
desactivan las etapas activas anteriores
362     AuxiliarListaEtapasSimulacion(CInt(EtapasAnterioresConvergencia(ind3))) =
MenuPrincipalForm.listaEtapasSimulacion(CInt(EtapasAnterioresConvergencia(ind3))) 'Se guarda en la auxiliar
363     Next ind3
364     End If
365     End If
366     Next ind2
367     Next ind

```

Figura 3.7.13.2.9.6. Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

A la figura anterior es mostra com es fan dos bucles amb les llistes que guardaven les files amb coincidències. Si no hi ha convergència (línia 349) es mira si la fila on la transició és turquesa coincideix amb la fila de l'etapa activa anterior (línia 350). En cas afirmatiu voldrà dir que s'ha d'activar aquella etapa i acte seguit es guarda l'auxiliar. En el cas que hi hagi convergència activa i sigui adequada, en primer lloc s'activa l'etapa següent a la convergència i després fent un recorregut amb les etapes anteriors a la convergència es desactiven aquelles etapes. En totes les opcions es guarda a l'auxiliar la informació renovada. Abans de finalitzar cal desactivar alguna de les etapes i això s'observa al codi següent:

```

369     For ind = 0 To ListaEtapasAnteriores.LongCount - 1 'Se recorren las etapas anteriores en las que se ha encontrado
coincidencia
370     Dim Espacio = InStr(ListaEtapasAnteriores(ind), " ")
371     Dim ListaEtapasAnterioresAux As New List(Of String)
372     Dim ReconstruirEtapasAnteriores() As String
373     ListaEtapasAnterioresAux = ListaEtapasAnteriores
374     ListaEtapasAnterioresAux(ind) = ListaEtapasAnterioresAux(ind).Substring(Espacio,
Len(ListaEtapasAnterioresAux(ind)) - Espacio)
375     ReconstruirEtapasAnteriores = ListaEtapasAnterioresAux(ind).Split(",")
376     For ind1 = 0 To ReconstruirEtapasAnteriores.LongCount - 1 'Se recorren las etapas anteriores
377     ReconstruirEtapasAnteriores(ind1) = ReconstruirEtapasAnteriores(ind1).Replace(" ", "") 'Se eliminan espacios
378     Dim NumeroEtapasAnteriores As Integer 'Se define un entero que guarda el número de la etapa
379     Try
380     NumeroEtapasAnteriores = CInt(ReconstruirEtapasAnteriores(ind1)) 'Se intenta guardar el número en la
variable
381     For ind2 = 0 To ListaEtapaActivaAnterior.LongCount - 1 'Se recorre la lista de etapas activas anteriores
382     If ListaEtapaActivaAnterior(ind2) = NumeroEtapasAnteriores Then 'Si coincide el número de etapa con la
etapa activa anterior
383     MenuPrincipalForm.ListaEtapasSimulacion(NumeroEtapasAnteriores).Tag = False 'Se desactiva la etapa
384     AuxiliarListaEtapasSimulacion(NumeroEtapasAnteriores) =
MenuPrincipalForm.ListaEtapasSimulacion(NumeroEtapasAnteriores) 'Se guarda en la auxiliar
385     End If
386     Next ind2
387     Catch
388     End Try
389     Next ind1
390     Next ind

```

**Figura 3.7.13.2.9.7.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

Es recorre la llista d'etapes anteriors on hi ha hagut coincidència, s'ordenen de forma correcta i es recorren totes elles obtenint els números d'etapes anteriors. Es recorre de nou la llista d'etapes actives anteriors i si coincideix amb el número es desactiva aquella etapa i es guarda a l'auxiliar. L'última part d'aquest procediment s'observa a continuació:

```

391
392     'Se borran todas las variables:
393     ListaEtapasAnteriores.Clear()
394     GuardadoFila.Clear()
395     ListaEtapaActivaAnterior.Clear()
396     GuardadoFila2.Clear()
397     ActualizarImágenes()
398     End Sub

```

**Figura 3.7.13.2.9.8.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

Abans de finalitzar es borren totes les llistes utilitzades i es restableixen les variables. Finalment es truca a la funció ActualizarImágenes la qual en funció de l'atribut tag actualitzarà les imatges de cada etapa, acció, acció associada, temporitzador i temporitzador associat.

### 3.7.13.2.10 Procediment Aïllar Número

En primer lloc cal dir que aquest procediment es truca amb la fi d'esborrar tot allò de l'etiqueta que no es necessari i que molesta amb la fi de obtenir una cadena amb únicament el valor desitjat. En aquest

cas, el valor que es vol aïllar és el número del temporitzador per poder crear més endavant un de real amb aquest interval de temps. Així doncs la funció és la següent:

```

400      'Función que elimina las letras y los espacios:
401      Private Sub AislarNumero(ByRef cadena As String)
402          cadena = cadena.Replace(" ", "")
403          cadena = cadena.Replace("s", "")
404          cadena = cadena.Replace("m", "")
405          cadena = cadena.Replace("i", "")
406          cadena = cadena.Replace("n", "")
407          cadena = cadena.Replace("h", "")
408          cadena = cadena.Replace("(", "")
409          cadena = cadena.Replace(")", "")
410          cadena = cadena.Replace("/", "")
411      End Sub
    
```

**Figura 3.7.13.2.10.1.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

El mètode es molt simple ja que li arriba per referència un paràmetre i es manipula per eliminar els espais, les lletres s, m, i, n, h, els parèntesis i les barres. En finalitzar el procediment el paràmetre ha eliminat tots aquests caràcters.

### 3.7.13.2.11 Procediment Obtenir Substring

La funció que es mostra a continuació permet obtenir una subcadena amb la informació rellevant. El codi és el següent:

```

413      'Función que elimina la cadena desde el caracter que se busca hasta el final de la cadena
414      Private Sub ObtenerSubstring(ByRef cadena As String, caracter As String)
415          Dim Espacio = InStr(cadena, caracter)
416          cadena = cadena.Substring(Espacio, Len(cadena) - Espacio)
417      End Sub
    
```

**Figura 3.7.13.2.11.1.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

Per referència rep una cadena i per valor un caracter. Es cerca a la cadena la posició del caracter i finalment s'obté la subcadena desde la posició on s'ha trobat l'espai fins el final de la cadena. En finalitzar el procediment la cadena s'ha modificat ja que el paràmetre es passa per referència. Tant aquesta com la funció anterior es fan per reduir línies de codi ja que en certs punts es necessita fer-ho repetits cops.

### 3.7.13.2.12 Procediment Activar Etapes Accions i Temporitzadors

```

419 'Función que activa las etapas, acciones y temporizadores:
420 Private Sub ActivarEtapasAccionesYTimers(ind As Integer, ruta As String)
421     MenuPrincipalForm.listaEtapasSimulacion(ind).Image = Image.FromFile(ruta) 'Cambia la imagen de la etapa a seleccionado
422     AuxiliarListaEtapasSimulacion(ind) = MenuPrincipalForm.listaEtapasSimulacion(ind) 'Guardo el cambio en el auxiliar
423     Dim AccionEtapas0 = EnlacesForm.DataGridView1.Rows(ind).Cells(2).Value.ToString 'Obtiene la accion de la tabla de
enlaces
424     ObtenerSubstring(AccionEtapas0, " ") 'Obtiene la acción en sí
425     Dim AccionAsociadaEtapas0 = EnlacesForm.DataGridView1.Rows(ind).Cells(3).Value.ToString 'Lo mismo para acciones
asociadas y temporizadores
426     ObtenerSubstring(AccionAsociadaEtapas0, " ")
427     Dim TimerEtapas0 = EnlacesForm.DataGridView1.Rows(ind).Cells(4).Value.ToString
428     ObtenerSubstring(TimerEtapas0, " ")
429     Dim TimerAsociadoEtapas0 = EnlacesForm.DataGridView1.Rows(ind).Cells(5).Value.ToString
430     ObtenerSubstring(TimerAsociadoEtapas0, " ")
431     For ind1 = 0 To MenuPrincipalForm.listaLabsAccionSimulacion.LongCount - 1
432         If MenuPrincipalForm.listaLabsAccionSimulacion(ind1).Text.ToString = AccionEtapas0 And
MenuPrincipalForm.listaLabsAccionSimulacion(ind1).Top.ToString = MenuPrincipalForm.listaEtapasSimulacion(ind).Top + 7 Then
'Si la accion coincide con la etapa activa y se encuentra a su lado, se activa:
433             MenuPrincipalForm.listaAccionesSimulacion(ind1).Image = Image.FromFile("Imagenes\AccionSel.png")
434             AuxiliarListaAccionesSimulacion(ind1) = MenuPrincipalForm.listaAccionesSimulacion(ind1)
435         End If
436     Next
437     For ind1 = 0 To MenuPrincipalForm.listaLabsAccionAsociadaSimulacion.LongCount - 1
438         If MenuPrincipalForm.listaLabsAccionAsociadaSimulacion(ind1).Text.ToString = AccionAsociadaEtapas0 And
MenuPrincipalForm.listaLabsAccionAsociadaSimulacion(ind1).Top.ToString = MenuPrincipalForm.listaEtapasSimulacion(ind).Top + 7
Then 'Lo mismo para la accion asociada
439             MenuPrincipalForm.listaAccionesAsociadasSimulacion(ind1).Image = Image.FromFile("Imagenes\AccionAsoSel.jpg")
440             AuxiliarListaAccionesAsociadasSimulacion(ind1) = MenuPrincipalForm.listaAccionesAsociadasSimulacion(ind1)
441         End If
442     Next
443     For ind1 = 0 To MenuPrincipalForm.listaLabsTimerSimulacion.LongCount - 1
444         If MenuPrincipalForm.listaLabsTimerSimulacion(ind1).Text.ToString = TimerEtapas0 Then 'Si activa un temporizador
445             ContadoresTimer(ind1) += 1 'El contador suma 1
446             If ContadoresTimer(ind1) <= 1 Then 'Únicamente se debe crear un temporizador, el bucle podría generar más de
la cuenta y por eso está la limitación
447                 MenuPrincipalForm.listaTimerSimulacion(ind1).Image = Image.FromFile("Imagenes\AccionSel.png") 'Se marca
como activo el temporizador
448                 AuxiliarListaTimerSimulacion(ind1) = MenuPrincipalForm.listaTimerSimulacion(ind1) 'Se guarda el cambio
en el auxiliar
449                 Dim TransicionTimer = MenuPrincipalForm.listaLabsTransTimerSimulacion(ind1).Text 'Se crea una variable con
el nombre de la transicion
450                 MenuPrincipalForm.listaLabsTransTimerNegadoSimulacion(ind1).BackColor = Color.Turquoise 'El color de fondo
del timer negado respectivo se fija a color turquesa, se activa
451                 MenuPrincipalForm.listaLabsTransTimerSimulacion(ind1).BackColor = Color.Transparent 'El color de fondo del
timer en cuestion se fija a blanco, se desactiva
452                 AuxiliarListaLabsTransTimerNegadoSimulacion(ind1) =
MenuPrincipalForm.listaLabsTransTimerNegadoSimulacion(ind1) 'Se guardan cambios en los auxiliares
453                 AuxiliarListaLabsTransTimerSimulacion(ind1) = MenuPrincipalForm.listaLabsTransTimerSimulacion(ind1)
454                 ObtenerSubstring(TransicionTimer, "=") 'Se separa la cadena de la transicion del timer
455                 AislarNumero(TransicionTimer) 'Se obtiene el numero
456                 CrearTimer(CInt(TransicionTimer) * 1000, False, ind1) 'Se crea el timer con el número (en segundos a
milisegundos -> *1000), False ya que no es timer asociado y se guarda el numero del timer de ind1.
457             End If
458         End If
459     Next

```

Figura 3.7.13.2.12.1. Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

Així com es pot veure a la figura anterior aquesta funció consisteix en activar les etapes, accions i temporitzadors que calguin, o sigui, canviar la imatge mostrada d'objecte inactiu a actiu. També es generen els temporitzadors. En primer lloc es canvia l'imatge de l'etapa que s'activa i es guarda l'auxiliar (línia 422). Després s'obtenen les accions, accions associades, temporitzadors i temporitzadors associats de la taula d'enllaços que es guarden en diferents variables. En aquestes funcions és on es fan les crides dels procediments dels dos subapartats anteriors. A continuació es recorre la llista que conté les etiquetes de les accions i si n'hi ha alguna coincidència amb les variables anteriors i l'acció es situa al costat d'una etapa activa, aquesta acció s'activa, es canvia la seva imatge i es guarda l'auxiliar. Això es repeteix per les llistes d'accions associades, temporitzadors i temporitzadors associats. En el cas dels temporitzadors, un cop s'activi, cal sumar una unitat al comptador indicat (línia 445) amb la fi de limitar a un únic temporitzador actiu per etapa, ja que es podrien activar de més per error. Després s'activa l'acció de temporitzador que s'està tractant, i la transició negada també (utilitzant el fons de color turquesa). Totes les accions es guarden a les variables auxiliars i acte seguit es tracta d'aïllar el número de la transició del temporitzador actiu. Un cop s'obté amb ajuda de les funcions anteriors, es crea el temporitzador amb la funció corresponent CrearTimer(interval de temps, tipus de temporitzador, número del temporitzador).

### 3.7.13.2.13 Procediment Desactivar Etapes Accions i Temporitzadors

```

479 'Funció que desactiva las etapas, acciones y temporizadores (mismo funcionamiento que la función anterior pero cambiando
480 las imágenes a desactivado):
481 Private Sub DesactivarEtapasAccionesYTimers(ind As Integer, ruta As String)
482     MenuPrincipalForm.listaEtapasSimulacion(ind).Image = Image.FromFile(ruta)
483     AuxiliarListaEtapasSimulacion(ind) = MenuPrincipalForm.listaEtapasSimulacion(ind)
484     Dim AccionEtapa0 = EnlacesForm.DataGridView1.Rows(ind).Cells(2).Value.ToString
485     Dim AccionAsociadaEtapa0 = EnlacesForm.DataGridView1.Rows(ind).Cells(3).Value.ToString
486     Dim TimerEtapa0 = EnlacesForm.DataGridView1.Rows(ind).Cells(4).Value.ToString
487     Dim TimerAsociadoEtapa0 = EnlacesForm.DataGridView1.Rows(ind).Cells(5).Value.ToString
488     Dim AccionEtapa0 = EnlacesForm.DataGridView1.Rows(ind).Cells(2).Value.ToString
489     Dim TimerAsociadoEtapa0 = EnlacesForm.DataGridView1.Rows(ind).Cells(5).Value.ToString
490     Dim AccionEtapa0 = EnlacesForm.DataGridView1.Rows(ind).Cells(2).Value.ToString
491     Dim TimerAsociadoEtapa0 = EnlacesForm.DataGridView1.Rows(ind).Cells(5).Value.ToString
492     For ind1 = 0 To MenuPrincipalForm.listaLabsAccionSimulacion.LongCount - 1
493         If MenuPrincipalForm.listaLabsAccionSimulacion(ind1).Text.ToString = AccionEtapa0 And
494             MenuPrincipalForm.listaAccionesSimulacion(ind1).Image = Image.FromFile("Imagenes\Accion.png")
495             AuxiliarListaAccionesSimulacion(ind1) = MenuPrincipalForm.listaAccionesSimulacion(ind1)
496         End If
497     Next
498     For ind1 = 0 To MenuPrincipalForm.listaLabsAccionAsociadaSimulacion.LongCount - 1
499         If MenuPrincipalForm.listaLabsAccionAsociadaSimulacion(ind1).Text.ToString = AccionAsociadaEtapa0 And
500             MenuPrincipalForm.listaAccionesAsociadasSimulacion(ind1).Image = Image.FromFile("Imagenes\AccionAso.png")
501             AuxiliarListaAccionesAsociadasSimulacion(ind1) = MenuPrincipalForm.listaAccionesAsociadasSimulacion(ind1)
502         End If
503     Next
504     For ind1 = 0 To MenuPrincipalForm.listaLabsTimerSimulacion.LongCount - 1
505         If MenuPrincipalForm.listaLabsTimerSimulacion(ind1).Text.ToString = TimerEtapa0 Then
506             MenuPrincipalForm.listaTimerSimulacion(ind1).Image = Image.FromFile("Imagenes\Accion.png")
507             AuxiliarListaTimerSimulacion(ind1) = MenuPrincipalForm.listaTimerSimulacion(ind1)
508         End If
509     Next
510     For ind1 = 0 To MenuPrincipalForm.listaLabsTimerAsociadoSimulacion.LongCount - 1
511         If MenuPrincipalForm.listaLabsTimerAsociadoSimulacion(ind1).Text.ToString = TimerAsociadoEtapa0 Then
512             MenuPrincipalForm.listaTimerAsociadoSimulacion(ind1).Image = Image.FromFile("Imagenes\AccionAso.png")
513             AuxiliarListaTimerAsociadoSimulacion(ind1) = MenuPrincipalForm.listaTimerAsociadoSimulacion(ind1)
514         End If
515     Next
516 End Sub

```

Figura 3.7.13.2.13.1. Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

Tal i com s'observa a la figura anterior aquesta funció consisteix en desactivar les etapes, accions i temporitzadors que calguin, o sigui, canviar la imatge mostrada d'objecte actiu a objecte inactiu. Això s'aconsegueix de la mateixa manera que a la funció explicada al subapartat anterior. En primer lloc es canvia l'imatge de l'etapa que es desactiva i es guarda l'auxiliar (línia 482). Després s'obtenen les accions, accions associades, temporitzadors i temporitzadors associats de la taula d'enllaços que es guarden en diferents variables. A continuació es recorre la llista que conté les etiquetes de les accions i si n'hi ha alguna coincidència amb les variables anteriors i l'acció es situa al costat d'una etapa no activa, aquesta acció es converteix en no activa, es canvia la seva imatge i es guarda l'auxiliar. Això es fa per totes les llistes d'accions associades, temporitzadors i temporitzadors associats. En aquest cas no cal fer res amb els temporitzadors.

### 3.7.13.2.14 Procediment Actualitzar Imatges

```

517 'Actualiza las imagenes de la simulación:
518 Private Sub ActualizarImagenes()
519     For ind = 0 To MenuPrincipalForm.listaEtapasSimulacion.LongCount - 1 'Recorre la lista de etapas de la simulación
520         If MenuPrincipalForm.listaEtapasSimulacion(ind).Tag = True Then 'Si la etapa está activa:
521             If ind = 0 Then 'Etapas
522                 ActivarEtapasAccionesYTimers(ind, "Imagenes\Etapa0Sel.jpg") 'Para activar la etapa0, acciones y
temporizadores asociados a ella
523             Else 'Resto de etapas
524                 ActivarEtapasAccionesYTimers(ind, "Imagenes\EtapaSel.jpg") 'Para activar las etapas, acciones y
temporizadores asociados a ellas
525             End If
526         Else 'Si la etapa no está activa:
527             If ind = 0 Then 'Etapas
528                 DesactivarEtapasAccionesYTimers(ind, "Imagenes\Etapa0.jpg") 'Para desactivar la etapa0, acciones y
temporizadores asociados a ella
529             Else 'Resto de etapas
530                 DesactivarEtapasAccionesYTimers(ind, "Imagenes\Etapa.png") 'Para desactivar las etapas, acciones y
temporizadores asociados a ellas
531             End If
532         End If
533     Next
534 End Sub

```

**Figura 3.7.13.2.14.1.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

Després de actualitzar la simulació és necessari actualitzar les etapes que estiguin actives, juntament amb les accions i els temporitzadors que s'han de posar en marxa. Per fer-ho, es recorre la llista d'etapes de la simulació tal i com s'observa a la figura anterior (línia 519) i en cas que l'atribut tag sigui vertader voldrà dir que l'etapa està activa. Per les etapes actives es trucarà a la funció ja explicada `ActivarEtapasAccionesYTimers` i per les que no estan actives al mètode `DesactivarEtapasAccionesYTimers` amb els paràmetres adients per cada etapa.

### 3.7.13.2.15 Procediment Crear Temporitzador

Quan és necessari crear un temporitzador, es truca a aquesta funció. Aquí, es declara un objecte del tipus `Temporizador` (classe creada per l'usuari), amb les característiques que l'arriben per paràmetres de la pròpia funció. Un cop es crea el temporitzador s'inicialitza i s'afegeix un esdeveniment que s'executa en el moment en que el temporitzador arriba al temps establert. Tota aquesta explicació es relaciona amb el codi que es mostra a continuació:

```

536 'Función que crea el timer:
537 Public Sub CrearTimer(tiempo As Double, Asociado As Boolean, ind1 As Integer)
538     Dim Temporizador1 As New Temporizador(tiempo, ind1, Asociado) With {.Enabled = True, .PropNumero = ind1}
'Definición del timer
539     Temporizador1.Start() 'Inicializa el timer
540     AddHandler Temporizador1.Elapsed, AddressOf ActualizarTimers 'En cuanto el timer llega al tiempo establecido por la
transición llama a la función de actualizar timers
541 End Sub

```

**Figura 3.7.13.2.15.1.** Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

### 3.7.13.2.16 Procediment Actualitzar Temporitzador

En el moment en que un temporitzador arriba al seu temps establert, es truca a aquest procediment. Segons els atributs definits a la creació del temporitzador, es manipulen els comptadors i les variables necessàries que s'utilitzen per actualitzar la simulació. A més, es para el temporitzador, es deshabilita i es truca a recuperar les dades i a actualitzar la simulació. El que s'ha explicat en aquest paràgraf es veu reflectit en la figura següent:

```

543 'Función al acabar un timer:
544 Private Sub ActualizarTimers(ByVal sender As Object, ByVal e As EventArgs)
545     If sender.PropTipo = False Then 'Si el timer no es asociado:
546         ContadorTimerAcabado(sender.PropNumero) = 1 'En el número del timer de la lista de contadores acabados el timer ha
           acabado y el valor es 1
547         NumeroTimer(sender.PropNumero) = sender.PropNumero 'En la lista de numero de timer se guarda el número de timer
548     Else 'Si el timer es asociado lo mismo para las variables asociadas.
549         ContadorTimerAsociadoAcabado(sender.PropNumero) = 1
550         NumeroTimerAsociado(sender.PropNumero) = sender.PropNumero
551     End If
552     sender.Stop() 'Se para el timer
553     sender.Enabled = False 'Se deshabilita el timer
554     RecuperarDatos() 'Se recuperan los datos
555     ActualizarSimulacion() 'Se actualiza la simulacion de nuevo
556 End Sub

```

Figura 3.7.13.2.16.1. Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

### 3.7.13.2.17 Procediment Recuperar Dades

Aquesta funció es truca cada cop que finalitza un temporitzador. Per alguna raó desconeguda quan això succeeix es perden certes variables de la classe menú principal i, com cal utilitzar-les dins de la classe simulació és necessari recuperar-les. Així doncs s'igualen les variables de l'altra classe a les variables auxiliars definides a la classe simulació en un principi tal i com es mostra al codi següent:

```

558 'Función de recuperación de datos, al terminar un temporizador se borran, por tanto hay que recuperarlas.
559 Private Sub RecuperarDatos()
560     MenuPrincipalForm.listaLabsSimulacion = AuxiliarListaLabsSimulacion
561     MenuPrincipalForm.listaLabsAccionSimulacion = AuxiliarListaLabsAccionSimulacion
562     MenuPrincipalForm.listaLabsAccionAsociadaSimulacion = AuxiliarListaLabsAccionAsociadaSimulacion
563     MenuPrincipalForm.listaLabsTimerSimulacion = AuxiliarListaLabsTimerSimulacion
564     MenuPrincipalForm.listaLabsTimerAsociadoSimulacion = AuxiliarListaLabsTimerAsociadoSimulacion
565     MenuPrincipalForm.listaLabsTransTimerSimulacion = AuxiliarListaLabsTransTimerSimulacion
566     MenuPrincipalForm.listaLabsTransTimerAsociadoSimulacion = AuxiliarListaLabsTransTimerAsociadoSimulacion
567     MenuPrincipalForm.listaLabsTransTimerNegadoSimulacion = AuxiliarListaLabsTransTimerNegadoSimulacion
568     MenuPrincipalForm.listaLabsTransTimerAsociadoNegadoSimulacion = AuxiliarListaLabsTransTimerAsociadoNegadoSimulacion
569     MenuPrincipalForm.listaEtapasSimulacion = AuxiliarListaEtapasSimulacion
570     MenuPrincipalForm.listaAccionesSimulacion = AuxiliarListaAccionesSimulacion
571     MenuPrincipalForm.listaAccionesAsociadasSimulacion = AuxiliarListaAccionesAsociadasSimulacion
572     MenuPrincipalForm.listaTimerSimulacion = AuxiliarListaTimerSimulacion
573     MenuPrincipalForm.listaTimerAsociadoSimulacion = AuxiliarListaTimerAsociadoSimulacion
574     EnlacesForm.DataGridView1 = AuxiliarDataGridView
575 End Sub

```

Figura 3.7.13.2.17.1. Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

### 3.7.13.2.18 Procediment Tick Temporitzador 2

Aquest esdeveniment s'executa cada 0,3 segons de simulació ja que el temporitzador 2 té aquest temps establert com a interval a les seves propietats. Així doncs cada cop que succeeix aquest interval de temps es prova de trucar a la funció ActualizarSimulacion, en cas que s'hagi produït un error dins d'aquest mètode es mostra un missatge d'error. En teoria no s'hauria de mostrar mai però en cas d'error salva a l'usuari de que hagi de reiniciar l'aplicació i perdi el progrés. A continuació es mostra el codi corresponent a aquest subapartat:

```

577 'Temporizador cada 0,3 segundos actualiza simulación, si hay un error salta un mensaje.
578 Private Sub Timer2_Tick(sender As Object, e As EventArgs) Handles Timer2.Tick
579     Try
580         ActualizarSimulacion()
581     Catch
582         MessageBox.Show("Se ha producido un error en la simulación")
583     End Try
584 End Sub

```

Figura 3.7.13.2.18.1. Captura d'imatge part del codi annex A14. (Font: Pablo Mañas)

### 3.7.14. Temporitzador Classe

Aquesta classe es crea amb la fi de guardar certes característiques dels objectes de tipus Timer que aquesta no permet. El codi d'aquesta classe és el que s'observa a continuació:

```

1  Public Class Temporizador
2
3      Inherits System.Timers.Timer 'Herencia de la clase Timers
4
5      Private Numero As Integer 'Definición de número para definir el número del temporizador
6      Private Tipo As Boolean 'Definición del tipo (temporizador asociado o no)
7
8      Public Sub New(interval As Double, Numero As Integer, Tipo As Boolean)
9          MyBase.New(interval) 'Define intervalo
10         Me.Numero = Numero 'Si recibe un "Numero" en la definicion lo guarda en su propiedad "Numero"
11         Me.Tipo = Tipo 'Si recibe el "Tipo" en la definicion lo guarda en su propiedad "Tipo"
12     End Sub
13
14     Public Property PropNumero() As Integer
15     Get
16         Return Numero 'Si se llama a su propiedad numero devuelve la propiedad "Numero"
17     End Get
18     Set(value As Integer) 'Si se hace set a la propiedad "Numero" con un valor
19         Numero = value 'La propiedad "Numero" se convierte en ese valor
20     End Set
21 End Property
22
23     Public Property PropTipo() As Boolean
24     Get
25         Return Tipo 'Si se llama a su propiedad tipo devuelve la propiedad "Tipo"
26     End Get
27     Set(value As Boolean) 'Si se hace set a la propiedad "Tipo" con un valor
28         Tipo = value 'La propiedad "Tipo" se convierte en ese valor
29     End Set
30 End Property
31
32 End Class

```

**Figura 3.7.14.1.** Captura d'imatge codi annex A15. (Font: Pablo Mañas)

En primera instància s'hereta tot de la classe Timers i es defineixen els dos atributs que s'afegiran a la classe nova. Aquesta es declara a la línia 8 amb la característica interval heretada de la classe timer i amb les noves propietats anteriors. Per permetre que es puguin modificar els atributs nous s'utilitzen les instruccions Get i Set. Amb la primera funció retorna l'atribut al que es fa referència quan es fa la crida (per exemple si Temporitzador.PropNumero es guarda en una variable, aquesta obtindrà el número del temporitzador). En el cas del Set s'aconsegueix que si s'accedeix a l'atribut i es canvia, quan es torni a cridar proporcionarà l'atribut canviat (per exemple si Temporitzador.PropNumero = 2 i després es fa  $x = \text{Temporitzador.PropNumero}$ ,  $x$  valdrà 2). Funciona de la mateixa manera amb la propietat Tipo.

### 3.7.15. Temporitzador Form

En trucar a aquesta classe es mostra un quadre com el de la figura 3.2.5. En primer lloc s'observa un TextBox, 3 RadioButton, un CheckBox i dos botons d'acceptar i cancel·lar.

#### 3.7.15.1. Variables

Es declaren 3 etiquetes, 1 PictureBox per fer la imatge del temporitzador i 6 llistes que permeten guardar les transicions, i els temporitzadors associats i normals.



### 3.7.15.2. Procediments i funcions

A continuació es mostra l'explicació i el codi de cadascun dels mètodes.

#### 3.7.15.2.1 Procediment limitació tecles del teclat

Així com s'ha vist a la figura 3.7.8.1. en aquesta classe també es prohibeix escriure altres tecles que no siguin números. En aquest cas es fa una petita modificació en la funció que tampoc permet escriure el caràcter “.” ja que no interessa.

#### 3.7.15.2.2 Procediment Botó Acceptar

```

35 'Botón aceptar:
36 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
37 If TextBox1.Text <> "" Then 'Si el usuario escribe algo
38 If RadioButton1.Checked Then 'Si se selecciona la primera opción:
39 ConstruirTimer() 'Se construye el timer
40 ConstruirLabelTimer() 'Se construye la etiqueta del timer
41 ConstruirTransicionVLabelTimer() 'Se construye la transicion y su etiqueta
42 LabelTransTimer.Text += " h)" 'Se añade la medida de tiempo
43 If CheckBox1.Checked = True Then 'Si está marcada la caja
44 LabelTransTimerNegado.Text += " h)" 'Se añade la medida de tiempo
45 End If
46 MenuPrincipalForm.contadorTimer += 1 'Suma 1 el contador de timers
47 Me.Close() 'Se cierra la clase
48 ElseIf RadioButton2.Checked Then 'Si se selecciona la primera opción:
49 ConstruirTimer() 'Se construye el timer
50 ConstruirLabelTimer() 'Se construye la etiqueta del timer
51 ConstruirTransicionVLabelTimer() 'Se construye la transicion y su etiqueta
52 LabelTransTimer.Text += " min)" 'Se añade la medida de tiempo
53 If CheckBox1.Checked = True Then 'Si está marcada la caja
54 LabelTransTimerNegado.Text += " min)" 'Se añade la medida de tiempo
55 End If
56 MenuPrincipalForm.contadorTimer += 1 'Suma 1 el contador de timers
57 Me.Close() 'Se cierra la clase
58 ElseIf RadioButton3.Checked Then 'Si se selecciona la tercera opción:
59 ConstruirTimer() 'Se construye el timer
60 ConstruirLabelTimer() 'Se construye la etiqueta del timer
61 ConstruirTransicionVLabelTimer() 'Se construye la transicion y su etiqueta
62 LabelTransTimer.Text += " s)" 'Se añade la medida de tiempo
63 If CheckBox1.Checked = True Then 'Si está marcada la caja
64 LabelTransTimerNegado.Text += " s)" 'Se añade la medida de tiempo
65 End If
66 MenuPrincipalForm.contadorTimer += 1 'Suma 1 el contador de timers
67 Me.Close() 'Se cierra la clase
68 Else
69 MessageBox.Show("Selecciona una medida de tiempo") 'Si no se selecciona medida tiempo se avisa con un mensaje
70 End If
71 Else
72 MessageBox.Show("Escribe el tiempo de duración") 'Si no se escribe nada se pide escribir al usuario algún número
73 End If
74 End Sub

```

Figura 3.7.15.2.2.1. Captura d'imatge part del codi annex A16. (Font: Pablo Mañas)

En pulsar acceptar primer de tot l'usuari ha d'haver escrit alguna cosa al textbox1 (línia 37) si no es així s'avisava amb un missatge (línia 72). Després en funció al que s'escull als RadioButtons i si el CheckBox està o no marcat, es farà una acció o altra. Bàsicament modifica el text afegint la cadena “ h)”, “ min)” o “ s)” en funció de l'elecció a la variable adequada. En cas que no es seleccioni cap RadioButton també s'avisava (línia 69).

#### 3.7.15.2.3 Procediment Construir Temporitzador

```

76 'Función que construye la acción de tipo timer:
77 Public Sub ConstruirTimer()
78     Timer = New PictureBox With {.Left = 200, .Top = 100, .Cursor = Cursors.SizeAll, .Tag = False} 'Se define
79     características del timer
80     If MenuPrincipalForm.TimerAsociado = False Then 'Si no es un timer asociado
81         Timer.Image = Image.FromFile("Imágenes\Timer.png") 'Imagen del timer
82         Timer.SizeMode = PictureBoxSizeMode.StretchImage 'Ajuste de la imagen
83         Timer.Width = 90 'Ajuste tamaños
84         Timer.Height = 28
85     Else 'Si es un timer asociado
86         Timer.Image = Image.FromFile("Imágenes\TimerAso.png") 'Imagen del timer asociado
87         Timer.SizeMode = PictureBoxSizeMode.StretchImage 'Ajuste de la imagen
88         Timer.Width = 65 'Ajuste tamaños
89         Timer.Height = 28
90     End If
91     MenuPrincipalForm.Panel1.Controls.Add(Timer) 'Se añade el timer al panel
92     If MenuPrincipalForm.TimerAsociado = False Then 'Si no es un timer asociado
93         ListaTimer.Add(Timer) 'Se añade el timer a la lista correspondiente
94     Else 'Si es un timer asociado
95         MenuPrincipalForm.ListaCoordsTimer.Add(Timer.Location) 'Se añaden las coordenadas a la lista
96         MenuPrincipalForm.ListaCoordsTimerAsociado.Add(Timer.Location) 'Se añaden las coordenadas a la lista
97     End If
98 End Sub

```

**Figura 3.7.15.2.3.1.** Captura d'imatge part del codi annex A16. (Font: Pablo Mañas)

Tal i com s'observa a la imatge anterior quan es crida a la funció ConstruirTimer, es genera un PictureBox amb les característiques adients. Després en funció de la variable TimerAsociado del menú principal, es modificarà la imatge ja que es diferent un temporitzador normal que un d'associat. Aquesta variable es modifica en fer clic sobre el dibuix del temporitzador normal o de l'associat al menú principal. Un cop s'han ajustat les mides del PictureBox creat s'afegeix al panell i de nou en funció de la variable mencionada anteriorment, es decideix a quines llistes s'afegeix l'objecte i les coordenades.

#### 3.7.15.2.4 Procediment Construir Etiqueta Temporitzador

```

100 'Función que construye la etiqueta de la acción de tipo timer:
101 Public Sub ConstruirLabelTimer()
102     LabelTimer = New Label With {.AutoSize = True, .Text = "TIM " + MenuPrincipalForm.contadorTimer.ToString, .Left =
103     Timer.Left + (Timer.Width() / 3.6), .Top = Timer.Top + (Timer.Height() / 4), .Height = 13, .Width = 60, .Parent = Timer,
104     .BackColor = Color.Transparent} 'Se definen características de la etiqueta
105     MenuPrincipalForm.Panel1.Controls.Add(LabelTimer) 'Se añade la etiqueta al panel
106     LabelTimer.BringToFront() 'Se envía al frente
107     LabelTimer.ResetBackColor() 'Se resetea el color de fondo
108     If MenuPrincipalForm.TimerAsociado = False Then 'Si no es un timer asociado
109         MenuPrincipalForm.ListaLabsTimer.Add(LabelTimer) 'Se añade la etiqueta a la lista correspondiente
110     Else 'Si es un timer asociado
111         MenuPrincipalForm.ListaLabsTimerAsociado.Add(LabelTimer) 'Se añade la etiqueta a la lista correspondiente
112     End If
113 End Sub

```

**Figura 3.7.15.2.4.1.** Captura d'imatge part del codi annex A16. (Font: Pablo Mañas)

A continuació de la funció anterior es troba la construcció de l'etiqueta associada a l'acció de temporitzador. Funciona de la mateixa manera que totes les etiquetes i no cal remarcar res. Únicament que al text se li afegeix "TIM " al principi per indicar el tipus que és. El número rere l'etiqueta anterior depèn de les vegades que s'hagi fet clic a qualsevol botó de temporitzador. En funció de la mateixa variable que abans s'escull a quina llista s'afegeix.

#### 3.7.15.2.5 Procediment Construir Transició i etiqueta del temporitzador

L'última funció d'aquesta classe consta de dos parts. La primera és la següent:

```

114     Public Sub ConstruirTransicionYLabelTimer()
115         TransicionesForm.CrearTransicion() 'Se crea la transición llamando a la clase
116         If MenuPrincipalForm.TimerAsociado = False Then 'Si el timer no es asociado
117             listaTransTimer.Add(TransicionesForm.transicion) 'Se añade la transición a la lista correspondiente
118             TransicionesForm.transicion.Left = 200 'Se posiciona en el lugar
119             TransicionesForm.transicion.Top = 150
120             MenuPrincipalForm.listaCoordsTransTimer.Add(TransicionesForm.transicion.Location) 'Se añaden las coordenadas a
la lista correspondiente
121         Else
122             listaTransTimerAsociado.Add(TransicionesForm.transicion) 'Se añade la transición a la lista correspondiente
123             TransicionesForm.transicion.Left = 200 'Se posiciona en el lugar
124             TransicionesForm.transicion.Top = 150
125             MenuPrincipalForm.listaCoordsTransTimerAsociado.Add(TransicionesForm.transicion.Location) 'Se añaden las
coordenadas a la lista correspondiente
126         End If
127         LabelTransTimer = New Label With {.AutoSize = True, .Text = "(t" + MenuPrincipalForm.contadorTimer.ToString + " = " +
TextBox1.Text, .Left = TransicionesForm.transicion.Left + TransicionesForm.transicion.Width(), .Top =
TransicionesForm.transicion.Top + (TransicionesForm.transicion.Height() / 3), .BackColor = Color.Transparent, .Height = 13,
.Cursor = Cursors.SizeAll} 'Se definen las características de la etiqueta
128         MenuPrincipalForm.Panell.Controls.Add(LabelTransTimer) 'Se añade la etiqueta al panel
129         LabelTransTimer.BringToFront() 'Se envía al frente
130         If MenuPrincipalForm.TimerAsociado = False Then 'En función del tipo de timer se añade a la lista que corresponde:
131             MenuPrincipalForm.listaLabsTransTimer.Add(LabelTransTimer)
132         Else
133             MenuPrincipalForm.listaLabsTransTimerAsociado.Add(LabelTransTimer)
134         End If

```

**Figura 3.7.15.2.5.1.** Captura d'imatge part del codi annex A16. (Font: Pablo Mañas)

En aquesta part es crea la transició del temporitzador de la mateixa manera que es creen les transicions normals juntament amb l'etiqueta associada a aquesta nova transició. En funció de la variable booleana es situarà l'objecte en un lloc o altre i a la llista corresponent. El mateix passa per l'etiqueta associada.

En aquest punt cal mirar si s'ha marcat el CheckBox del quadre de diàleg. Així doncs, la segona part respon a aquesta qüestió:

```

135     If CheckBox1.Checked = True Then 'Si se ha seleccionado la caja (se crea un timer negado de la misma manera que se
ha hecho en esta función anteriormente)
136         TransicionesForm.CrearTransicion()
137         TransicionesForm.transicion.Left = 300
138         TransicionesForm.transicion.Top = 150
139         If MenuPrincipalForm.TimerAsociado = False Then
140             listaTransTimerNegado.Add(TransicionesForm.transicion)
141             MenuPrincipalForm.listaCoordsTransTimerNegado.Add(TransicionesForm.transicion.Location)
142         Else
143             listaTransTimerAsociadoNegado.Add(TransicionesForm.transicion)
144             MenuPrincipalForm.listaCoordsTransTimerAsociadoNegado.Add(TransicionesForm.transicion.Location)
145         End If
146         LabelTransTimerNegado = New Label With {.AutoSize = True, .Text = "/(t" + MenuPrincipalForm.contadorTimer.ToString
+ " = " + TextBox1.Text, .Left = TransicionesForm.transicion.Left + TransicionesForm.transicion.Width(), .Top =
TransicionesForm.transicion.Top + (TransicionesForm.transicion.Height() / 3), .BackColor = Color.Transparent, .Height = 13,
.Cursor = Cursors.SizeAll}
147         MenuPrincipalForm.Panell.Controls.Add(LabelTransTimerNegado)
148         LabelTransTimerNegado.BringToFront()
149         If MenuPrincipalForm.TimerAsociado = False Then
150             MenuPrincipalForm.listaLabsTransTimerNegado.Add(LabelTransTimerNegado)
151         Else
152             MenuPrincipalForm.listaLabsTransTimerAsociadoNegado.Add(LabelTransTimerNegado)
153         End If
154     Else 'Si no se ha seleccionado la caja (se crea un timer negado virtual para que haya siempre 3 timers (2
transiciones negadas y 1 accion) para que al borrar se borren todos a la vez y no se pierda la referencia de cada uno la
construcción es la misma.
155         TransicionesForm.CrearTransicion()
156         TransicionesForm.transicion.Image = Nothing
157         TransicionesForm.transicion.Left = 9999 'Se envía fuera del panel
158         TransicionesForm.transicion.Top = 9999
159         If MenuPrincipalForm.TimerAsociado = False Then
160             listaTransTimerNegado.Add(TransicionesForm.transicion)
161             MenuPrincipalForm.listaCoordsTransTimerNegado.Add(TransicionesForm.transicion.Location)
162         Else
163             listaTransTimerAsociadoNegado.Add(TransicionesForm.transicion)
164             MenuPrincipalForm.listaCoordsTransTimerAsociadoNegado.Add(TransicionesForm.transicion.Location)
165         End If
166         LabelTransTimerNegado = New Label With {.AutoSize = True, .Text = ""}
167         MenuPrincipalForm.Panell.Controls.Add(LabelTransTimerNegado)
168         If MenuPrincipalForm.TimerAsociado = False Then
169             MenuPrincipalForm.listaLabsTransTimerNegado.Add(LabelTransTimerNegado)
170         Else
171             MenuPrincipalForm.listaLabsTransTimerAsociadoNegado.Add(LabelTransTimerNegado)
172         End If
173     End If
174 End Sub

```

**Figura 3.7.15.2.5.2.** Captura d'imatge part del codi annex A16. (Font: Pablo Mañas)

La continuació primerament mira si s'ha marcat la caixa. En cas afirmatiu es genera altre transició amb el text canviat ja que haurà de ser del tipus negat. El funcionament és igual que a la primera part ja que també es depèn de la variable TimerAsociado del menú principal.

En el cas que l'usuari no hagi marcat la casella (línia 154), és necessari crear una transició fictícia negada amb la fi que sempre hi hagi un temporitzador amb una transició i una transició negada. D'aquesta manera sempre tindran el mateix índex associat a la seva llista pròpia i serà més fàcil borrar-los al mateix temps, copiar-los i evitarà problemes a la simulació.

Això són totes les explicacions pel que fa a les funcions dels temporitzadors. Algunes coses es podrien fer de diferent manera però així ja funciona.

### 3.7.16. Tipus d'enllaç

Aquesta classe està relacionada amb la creació dels enllaços manuals. En cridar-se es mostra la finestra que es pot veure a la figura 3.2.8. Únicament conté dos botons d'acceptar i cancel·lar i 2 RadioButtons.

#### 3.7.16.1. Variables

Es manegen dues variables públiques de tipus booleà que s'inicialitzen com a fals. Segons el que es decideixi al quadre podran passar a ser vertaderes.

#### 3.7.16.2. Procediments i funcions

Els procediments corresponents són el del botó d'acceptar i el de cancel·lar. El segon únicament tanca la finestra de la classe, per tant, només cal observar el primer.

```

7      'Botón aceptar:
8      Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
9          If RadioButton1.Checked Then 'Si la primera opción se selecciona:
10             RespuestaAnterior = True 'La variable se convierte en verdad
11             Me.Close() 'Se cierra la clase
12         ElseIf RadioButton2.Checked Then 'Si la segunda opción se selecciona:
13             RespuestaSiguiente = True 'La variable se convierte en verdad
14             Me.Close() 'Se cierra la clase
15         Else
16             MessageBox.Show("Selecciona un tipo de enlace") 'Si no se selecciona nada se muestra un mensaje
17         End If
18     End Sub

```

**Figura 3.7.18.2.1.** Captura d'imatge codi annex A17. (Font: Pablo Mañas)

Segons el botó que s'hagi seleccionat en fer clic a acceptar es variarà el valor de la variable corresponent a true. Si no s'ha seleccionat res es mostra un missatge.

Això es tot pel que fa a la part de programació del treball, en cas que es vulgui mirar amb més detall, tot el codi es troba a l'annex A. Cal dir que el diagrama de classes de l'annex A0 no està de forma detallada ja que faltaria definir tots els procediments, però serveix de guia per entendre el funcionament del programa.

### 3.8. Millores al programa

Amb la fi de donar una visió crítica del programa i donar marge a millores d'aquest en un futur, s'ha decidit crear aquest subapartat on es destaquen els punts fluixos del funcionament de l'editor i es presenten algunes característiques que farien que l'usuari es sentís més còmode a l'hora d'utilitzar el programa.

Una de les característiques que es van plantejar inicialment a l'hora de realitzar el programa era la implementació d'una funció de zoom. Això permetria apropar i allunyar els objectes del GRAFCET i poder així realitzar els enllaços de forma més còmoda. Aquesta, però, no va ser possible incorporar-la perquè donava problemes amb la creació del GRAFCET i amb la composició de la taula d'enllaços i es va descartar pel fet que no s'aconseguien arreglar aquestes dificultats. Tot i així, plantejant alguns assumptes de programació de forma diferent es podria haver implementat i no s'exclou que no es pugui fer en un futur.

Altre aspecte millorable pot ser el mètode d'allargar i escurçar els enllaços, és a dir, les línies verticals i horitzontals. El fet d'haver de seleccionar primerament l'element i utilitzar el teclat per dur aquesta funció a terme és un mètode bastant molest i dificulta la creació dels diagrames. Una possible solució seria poder fer més llargues i curtes les línies clicant als extrems d'aquests elements amb el ratolí i modificant la llargada en funció del moviment realitzat.

Relacionat amb la característica anterior, també es podria fer que els objectes poguessin canviar de dimensió a través del moviment del ratolí, tot i que això podria fer fallar l'algoritme dels enllaços perquè modificaria la posició, llargada i ample dels diferents elements que formen el GRAFCET.

El fet d'haver d'obrir la simulació ja sigui de forma directa o indirecta per fer l'exportació en format Excel pot arribar a ser molest per l'usuari i es podria implementar d'una manera diferent la qual permetria fer-ho sense obrir cap finestra addicional. Implicaria modificar alguna de les funcions programades, més en concret la de l'exportació i potser la de simulació.

Moltes vegades les línies verticals i horitzontals es mouen massa i és difícil trobar els punts on l'enllaç es fa de forma correcta. Per millorar aquest fet, caldria ser més lax en les funcions de cerca d'enllaços programades. Relacionat amb aquest tema, encara que ja s'ha mencionat amb anterioritat, molts cops al moure una línia que es troba entre d'altres pel fet de l'ajuda al posicionament aquesta es perd. En aquest cas l'usuari hauria de desactivar l'ajuda o moure les línies amb el teclat, però tot això pot ser una incomoditat i s'hauria de poder detectar d'alguna manera i corregir automàticament l'error. A més, quan un grup seleccionat es mou si es deixa anar el clic prop d'un element i el posiciona, la resta del grup hauria de posicionar-se de forma correcta i no quedar-se de forma incorrecta respecte l'element posicionat. La solució possible podria ser que en el moment en que hi ha varis elements

movent-se, l'ajuda de posicionament quedi inhabilitada. Altre possible idea podria ser que un cop s'enllaci un element a un altre aquest quedi fix i no es pugui deslligar; tot i que podria arribar a ser molest en cas que l'usuari s'equivoqués d'enllaç ja que hauria de tornar a construir-lo i això ho faria de manera més lenta.

Altre característica a millorar és el fet que no es puguin exportar en pdf els fulls en format DIN A3. En un inici estava plantejat que el programa ho permetés però, després de molts intents i hores cercant la manera de fer-ho, no es va trobar res factible i es va decidir descartar aquesta funció.

Respecte a la pestanya d'edició, més concretament la funció de desfer y refer ("Deshacer" y "Rehacer"), el programa actual només permet desfer una acció i refer aquesta acció. En molts programes del mateix estil, aquestes funcions són capaces de dur-se a terme durant diferents accions, és a dir, poden arribar a desfer 15 o més accions que hagi fet l'usuari anteriorment. Per problemes a l'hora de programació es va eliminar que això es pogués fer en el programa actual, tot i que seguint treballant el codi es podria aconseguir implementar.

Encara que dintre de l'apartat 2 d'aquesta memòria s'ha parlat de les macroetapes, aquestes no apareixen en el programa. Això és degut a que no es va considerar inicialment la possibilitat de crear macroetapes i quan es van tenir en compte era molt difícil la seva implementació.

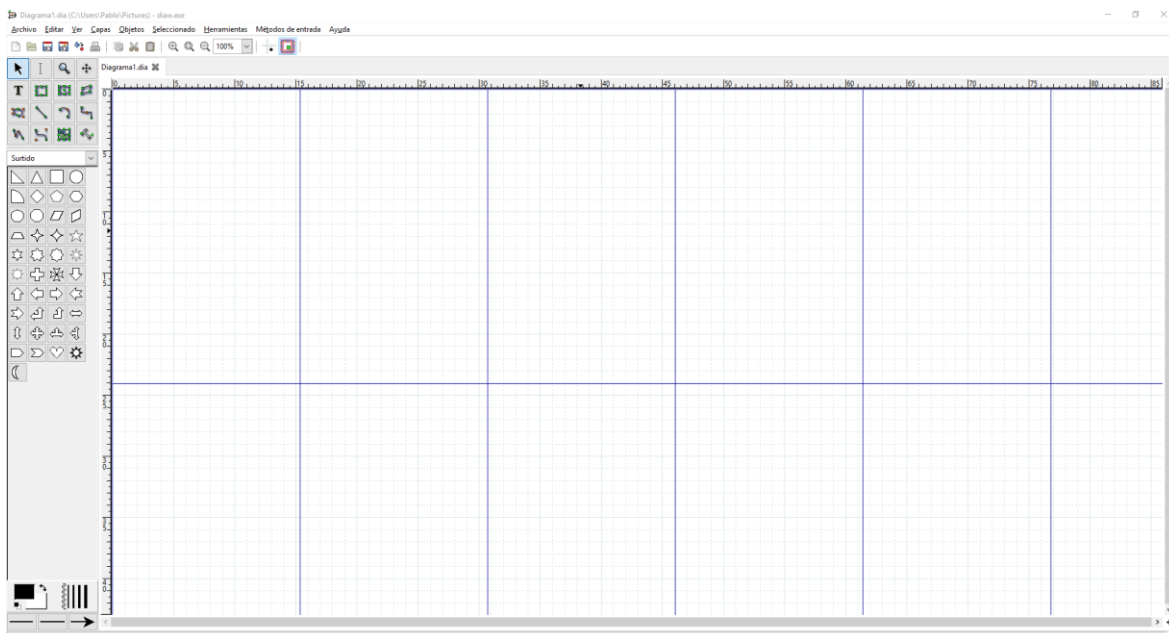
Per últim, es podria haver afegit, a més de les accions i els temporitzadors, els comptadors. Aquest tipus d'accions poden ser útils ja que en automatització poden aparèixer certs problemes que s'han de resoldre amb l'ús de comptadors. Aquests consisteixen en sumar cert número cada cop que s'activa la seva etapa i quan arriben al valor corresponent activen la seva transició de forma similar als temporitzadors els quals sí han sigut incorporats en el programa.

## 4. Estat de l'art

Prèviament a l'elaboració del programa "Editor de GRAFCETs – Pablo Mañas" es va fer una cerca d'alguna sèrie d'editors ja existents en el mercat per veure les seves característiques i tractar d'implementar-les i, en alguns casos, millorar-les. Per això, en aquest apartat s'expliquen diferents exemples d'editors i, finalment, es mostra una taula comparativa amb les diferents característiques dels programes juntament amb les de l'editor creat per l'autor d'aquest document.

### 4.1. Dia 0.97.2

El primer exemple d'editor sotmès a estudi és l'editor que s'anomena Dia i, en concret, la versió 0.97.2 del programa. A l'iniciar l'aplicació la finestra que es mostra és la següent:



**Figura 4.1.1.** Captura d'imatge de la finestra d'inici de Dia 0.97.2 (Font: Pablo Mañas)

A primera vista, s'observa que el programa no només està dissenyat per crear diagrames de tipus GRAFCET sinó que també disposa d'altres elements per elaborar diagrames de flux, llenguatge de modelització unificada (UML), circuits elèctrics, portes lògiques, etc. Aquesta característica mostra que es tracta d'un programa robust basat en l'edició de diagrames en diferents àmbits sense centrar-se precisament en l'edició de GRAFCETs.

A continuació es desglossen les diferents funcionalitats del programa respecte les pestanyes que apareixen a la part superior:

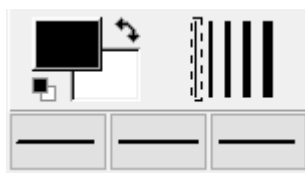
- **Pestanya “Archivo”:** Així com tots els programes d’edició, conté les característiques bàsiques de crear nou diagrama, obrir, guardar, exportar i imprimir. També permet crear diferents diagrames al mateix temps i editar-los.
- **Pestanya “Editar”:** Aquesta pestanya permet les funcions de desfer (fins a 15 accions), refer, copiar, tallar, duplicar, eliminar, cercar i altres eines de modificació de text.
- **Pestanya “Ver”:** Utilitzant les funcions d’aquesta pestanya l’usuari pot ampliar la imatge del diagrama per veure de més a prop la seva elaboració (a poc a poc o mitjançant un percentatge) i també reduir-la. També té un ajust òptim del diagrama que es fa tenint en compte els objectes apareguts al diagrama creat, un mode de finestra en pantalla completa i una sèrie de funcions que permeten ocultar i mostrar algunes barres d’elements o la quadrícula del diagrama. Per últim, a més es permet fer una vista nova del diagrama que s’està editant o, fins i tot, clonar-la.
- **Pestanya “Capas”:** Com si fos una eina de dibuix, el programa permet crear capes que es poden superposar una sobre l’altra i nomenar-les d’alguna manera. Així, també es poden moure els objectes a les diferents capes del diagrama i veure els objectes que apareixen a cada una de les capes creades.
- **Pestanya “Objetos”:** En el cas que en un diagrama s’hi trobin objectes superposats, en aquest subapartat es troben funcions que permeten que un objecte o varis es col·loquin al fons, a la part superior o un sobre de l’altre. A més es pot seleccionar un grup d’objectes i agrupar-lo per a què quan es seleccioni es moguin tots a l’uníson i, en cas que es vulguin tornar a separar, també hi ha un botó per fer-ho. En la mateixa pestanya es troben opcions per a diagrames de classes o UML ja que apareixen funcions de polimorfisme com són les classes pares i les heretades, com no forma part del que interessa en aquest document, no és necessari estendre aquests conceptes. Per últim, es permet alinear els diferents objectes seleccionats i modificar les seves propietats les quals varien segons l’objecte seleccionat.
- **Pestanya “Seleccionado”:** Fent clic en la pestanya corresponent, s’obren les diferents funcions que permeten seleccionar tots els objectes del diagrama, cap o invertir la selecció feta actualment. També es pot seleccionar l’objecte o objectes connectats al que està seleccionat en aquell moment o en transició. Clicant el botó adequat es permet seleccionar tots els objectes d’un mateix tipus. Després es troben una sèrie de caselles que en funció de la que es trobi activada variarà la selecció que es faci sobre el diagrama.
- **Pestanya “Herramientas”:** Dintre d’aquest subapartat es troben les funcions que apareixen de forma gràfica a la figura 4.1.2. Aquestes permeten modificar el diagrama actual, editar algun text que hagi sigut creat prèviament, ampliar una zona del diagrama amb ajuda del ratolí, desplaçar el diagrama, introduir text, crear objectes amb formes definides com poden ser: el·lipses, caixes, polígons, línies normals, amb corbes, múltiples i en zig-zag, arcs, lletres amb fons i imatges. A més, aquests objectes juntament amb els textos, es poden modificar



mitjançant la barra d'elements que es troba a la part inferior esquerra del programa tal i com es pot observar a la figura 4.1.1, més concretament a la figura següent 4.1.3. En resum, amb aquestes eines es pot variar el color de l'element, el fons, el seu gruix de línia, i en cas que sigui una línia es poden modificar les puntes perquè acabin i comencin d'una manera definida, per exemple, en punta de fletxa i l'estil de línia, com pot ser en línia de punts.

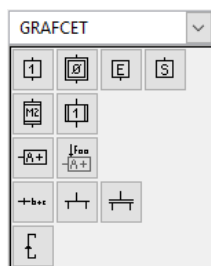


**Figura 4.1.2.** Captura d'imatge de les eines de Dia 0.97.2 (Font: Pablo Mañas)



**Figura 4.1.3.** Captura d'imatge de les propietats línies de Dia 0.97.2 (Font: Pablo Mañas)

- **Pestanya “Métodos de entrada”:** En aquesta pestanya apareixen els mètodes d'entrada corresponents als diferents tipus de teclats que es poden utilitzar per comunicar-se amb el programa.
- **Pestanya “Ayuda”:** En aquest subapartat es troba l'ajuda del programa on en fer clic s'obre el manual en anglès d'aquest on estan documentades totes les opcions i funcionalitats del programa que es resumeixen en aquesta memòria molt per sobre. També hi ha un botó que parla en resum sobre el programa on es pot veure la llicència i els autors del programa amb el seu correu de contacte.
- **Pestanya de selecció d' elements:** Aquesta pestanya es troba sota la barra d'eines mostrada a la figura 4.1.2 i en desplegar-se es poden seleccionar els tipus d'elements que es poden dibuixar sobre el diagrama. A la imatge següent (figura 4.1.4) es mostra la selecció del tipus GRAFCET i els elements que es mostren a l'instant de fer clic.



**Figura 4.1.4.** Captura d'imatge dels elements GRAFCET de Dia 0.97.2 (Font: Pablo Mañas)

- **Esdeveniments de teclat i ratolí:** A més de les pestanyes de la barra superior també es important mencionar com interactua el teclat i el ratolí amb el programa, per tant, ara s'expliquen aquests esdeveniments.

Per crear un objecte al diagrama cal fer clic esquerre sobre aquest en el panell d'elements que es genera quan es selecciona un tipus a la pestanya de selecció d'elements i, després clicar de nou al diagrama al lloc on es vulgui posicionar. Acte seguit apareixerà l'objecte creat seleccionat el qual es podrà moure polsant el ratolí sobre l'element i arronsant-lo fins la posició que es desitgi. En el cas que es vulguin modificar les seves propietats caldrà fer doble clic esquerra sobre l'objecte o un cop estigui seleccionat clicar a "Objetos" i "Propiedades" o clic dret i "Propiedades...". En fer aquesta combinació s'obrirà una finestra amb els detalls de l'objecte en qüestió i les propietats modificables que poden variar segons el tipus d'element, tot i que usualment són colors i mides del text que contenen. Si no hi ha cap element seleccionat; clicant i arronsant sobre el diagrama es crea un quadrat que selecciona tots els elements que estiguin al seu interior en el cas que el tipus de selecció sigui el correcte.

Amb la roda del ratolí es pot moure el diagrama amunt i avall i si prèviament es polsa la tecla Ctrl es fa Zoom, és a dir, s'amplia o s'allunya el diagrama.

Respecte el Zoom, aquest també es pot fer utilitzant les tecles + i – del teclat. A més d'aquests esdeveniments de teclat existeixen les dreteres que permeten accedir a les funcions corresponents amb la combinació de tecles polsades adequada.

Ara que ja s'ha explicat com funciona de manera resumida el programa en qüestió, es pot parlar d'algunes característiques que l'usuari pot trobar a faltar en un editor de GRAFCETs més concretament.

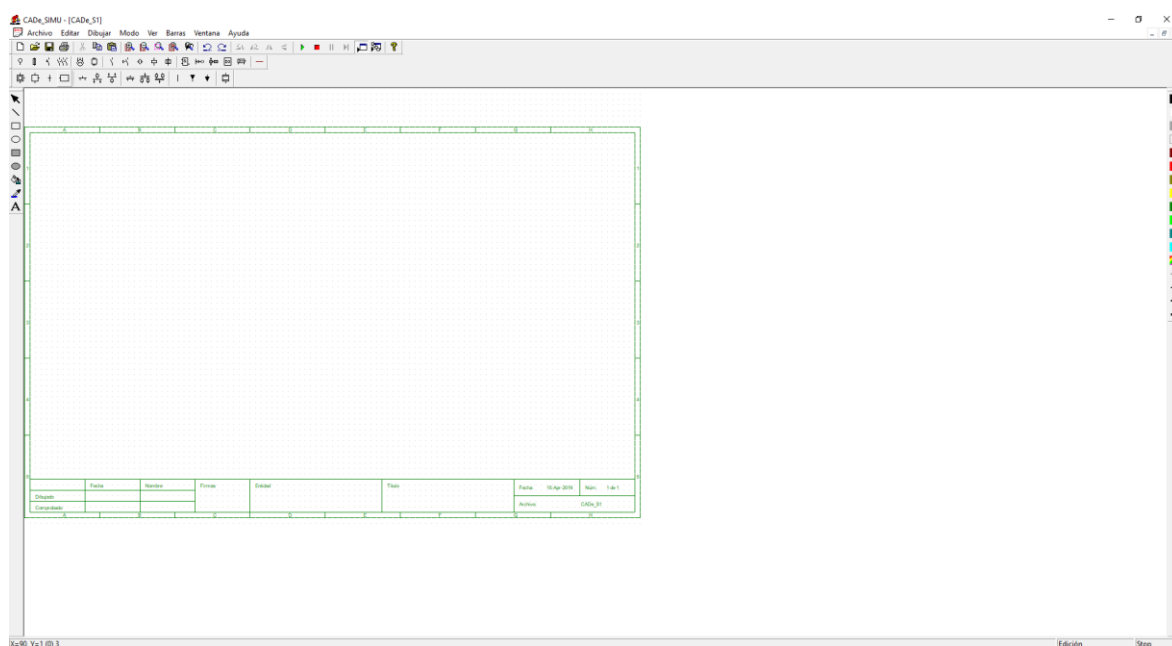
Primerament, encara que el programa es pot utilitzar per crear qualsevol GRAFCET ja que disposa de molts dels elements que apareixen en un diagrama d'aquest estil, no es permet fer cap simulació del diagrama creat. Això pot ser un inconvenient en cas que l'usuari vulgui provar si el seus sistema o diagrama funciona correctament. Pel que fa als elements que es troben en la pestanya GRAFCET es troba a faltar les accions associades múltiples a una etapa, és a dir, quan hi ha més d'una acció associada a una etapa. A més, tampoc hi ha cap menció a temporitzadors ni comptadors, tot i que aquests es podrien definir amb transicions i accions. Fixant-se en les transicions, no es pot definir si la

transició s'activa per flancs, encara que es podria dibuixar posteriorment a la creació de les transicions. Tampoc es poden crear taules de manera molt còmode i exportar el seu contingut a altres programes com pot ser Excel, tot i que en el cas de l'exportació, aquesta es pot fer en nombrosos formats. Aquest fet provoca que l'usuari hagi de fer taules a part per acabar de completar el GRAFCET.

Tot i que el programa Dia 0.97.2 pot tenir aquests petits inconvenients mencionats anteriorment, es tracta d'una eina molt útil i pràctic en la creació de GRAFCETS ja que és fàcil d'utilitzar, sobre tot en la creació d'enllaços entre els elements, i molt intuïtiva.

## 4.2. CADeSIMU v3

Així com s'ha fet amb el programa anterior, en aquest subapartat es resumiran les diferents funcions i característiques que conté el programa CADeSIMU, més concretament, la versió 3 d'aquest. L'aspecte inicial en obrir l'aplicació en qüestió és la següent:



**Figura 4.2.1.** Captura d'imatge de la finestra d'inici de CADeSIMU v3 (Font: Pablo Mañas)

A diferència del programa anterior, el CADeSIMU està més centrat en circuits elèctrics, electrònics, portes lògiques i biestables, actuadors pneumàtics i en diagrames com el ladder i el GRAFCET. A continuació s'expliquen les funcions del programa juntament amb altres característiques importants:

- **Pestanya "Archivo":** Conté les característiques bàsiques de crear nou diagrama, obrir, tancar, guardar, importar imatges, exportar imatge, imprimir, presentació preliminar a la impressió, configuració de la impressió i altres configuracions del diagrama que s'edita en qüestió. Una

característica bàsica que es presenta a l'inici es que el programa també permet treballar amb diversos diagrames al mateix temps i editar-los.

- **Pestanya “Editar”:** Aquesta pestanya conté les funcions de desfer (elimina l'últim element creat), refer (funció inversa a l'anterior), tallar, copiar, enganxar i seleccionar tot. També conté una casella que permet modificar el contingut del caixetí que s'observa al vèrtex inferior dret del nou document creat, tal i com es pot observar a la figura 4.2.2. Cal dir que aquest caixetí no es pot esborrar i, per tant, sempre apareixerà al diagrama.

	Fecha	Nombre	Firmas	Entidad	Título	Fecha: 15-Apr-2019	Núm: 1 de 1
Dibujado						Archivo: CADe_S4	
Comprobado							

**Figura 4.2.2.** Captura d'imatge del caixetí de CADeSIMU v3 (Font: Pablo Mañas)

- **Pestanya “Dibujar”:** Dintre d'aquest apartat es troben elements de dibuix que permeten traçar línies, dibuixar rectangles, el·lipses, rectangles i el·lipses amb l'àrea interior acolorida, omplir de color un àrea definida, obtenir el color d'algun element o escriure un text. Totes aquestes funcions es troben de manera més intuïtiva en forma d'ícones a una barra vertical a l'esquerra del programa. Aquesta sol treballar amb altra barra vertical que es veu a l'altre extrem de l'aplicació i que consisteix en definir els colors i el gruix de les línies o elements traçats. A la figura següent es presenten captures d'aquestes dues barres mencionades anteriorment:



**Figura 4.2.3.** Barres de dibuix de CADeSIMU v3 (Font: Pablo Mañas)

- **Pestanya “Modo”:** Aquesta pestanya es relaciona amb la simulació, ja que aquest programa, en oposició a l'anterior, permet fer simulacions. Aquestes es poden fer per comprovar el funcionament dels diferents diagrames que es vulguin crear. Si es decideix simular únicament el GRAFCET, el programa no ho permetrà ja que necessita activar les transicions d'alguna manera i, per tant, caldrà fer-ho amb ajuda d'un diagrama del tipus “ladder”, de contactes, amb pulsadors o altres elements elèctrics, electrònics o pneumàtics.

Així doncs, dins de la pestanya de mode es troba la casella d'edició que torna a permetre l'edició del diagrama en cas que s'hagi pulsat prèviament la simulació; el botó de simulació posa el document actiu en simulació en cas que estigui aquesta casella marcada o tots els documents en cas que sigui l'altra casella, mentre estigui el diagrama en aquest estat no es pot editar, només activar els pulsadors i interruptors corresponents o tots aquells elements

manuals que s’hagin utilitzat. Durant la simulació s’habilita la pausa en cas que es vulgui parar el diagrama (fer pausa als temporitzadors) però continuar amb la simulació i el pas per pas que faria les accions a cada cop que li pressionés al botó. L’última casella d’aquesta pestanya es centra amb les entrades i sortides del PLC (“Programmable Logic Controller”) però no cal explicar-la. Totes aquestes funcions a més es troben a la barra del programa següent:



**Figura 4.2.4.** Barra de mode (simulació) de CADeSIMU v3 (Font: Pablo Mañas)

- **Pestanya “Ver”:** Aquesta pestanya conté totes les eines de Zoom del programa i permet ampliar i allunyar el diagrama de diferents formes, inclús guarda la vista anterior i, per tant, es pot recuperar el Zoom anterior. També té una casella anomenada “Rejilla” la qual mostra o oculta els punts que es veuen al document a la figura 4.2.1 d’aquest mateix apartat i que serveixen per ajudar a situar els diferents objectes.
- **Pestanya “Barras”:** Dintre d’aquesta pestanya es troben tots els noms de les barres auxiliars que componen el programa i per tant, es poden seleccionar o deseleccionar amb la fi de que es mostrin en l’aplicació o no. Per defecte, totes estan seleccionades ja que l’usuari pot estar interessat en utilitzar-les totes. Si en canvi, ja s’està familiaritzat amb el programa, algunes d’elles es poden obviar i, per tant, deixar de mostrar-les amb ajuda dels botons que conformen aquesta pestanya.
- **Pestanya “Ventana”:** Aquí s’observen tots els documents oberts numerats per ordre. El document actiu es marca amb un senyal a l’esquerra de la casella corresponent ja que només es pot activar 1 al mateix temps. Dintre de la pestanya també es poden crear noves finestres, és a dir, nous documents de diagrames, i organitzar-los a la pantalla en modes de cascada o mosaic.
- **Pestanya “Ayuda”:** L’última pestanya mostra l’ajuda del programa amb informació de l’autor, la versió del programa i botons que obren ajuda online mitjançant un parell de pàgines web.
- **Barra d’elements:** A més de les barres d’elements ja mostrades i explicades a les figures 4.2.3 i 4.2.4, el programa mostra d’altres que serveixen per crear els objectes del diagrama o per editar-los.

Quan es selecciona un objecte aquest pot ser rotat 90° cap a dreta o esquerra o amirallat horitzontalment o verticalment. Per fer-ho es disposa de la barra següent:



**Figura 4.2.5.** Barra de rotació d’objectes de CADeSIMU v3 (Font: Pablo Mañas)

A més, es troba una segona barra on es poden seleccionar els tipus d'objectes que es volen dibuixar al diagrama. A la figura 4.2.6 es mostra aquesta segona barra de tipus d'elements, juntament amb una tercera barra on es troben els propis objectes que es podran incorporar al document actiu.



**Figura 4.2.6.** Barres de tipus d'element i elements de CADeSIMU v3 (Font: Pablo Mañas)

- **Esdeveniments de teclat i ratolí:** Així com s'han explicat per sobre a l'exemple anterior, en aquest cas també es considera necessari per acabar de completar les característiques del programa. En aquesta ocasió els esdeveniments de teclat són molt simples, ja que només s'utilitza la tecla Supr per esborrar els objectes seleccionats, l'Esc per deixar de seleccionar-los i les dreceres corresponents per activar cada una de les funcions individuals ja explicades en els anterior punts. Pel que fa al ratolí, aquest permet mitjançant el clic esquerre crear els objectes, moure els elements seleccionats, traçar les línies i els elements de dibuix i traçar les connexions mitjançant els cables. El doble clic esquerra obre finestres de propietats dels diferents elements seleccionats on es poden escriure els noms, la funció a la que estan sotmesos i en alguns casos descriure els comportaments.

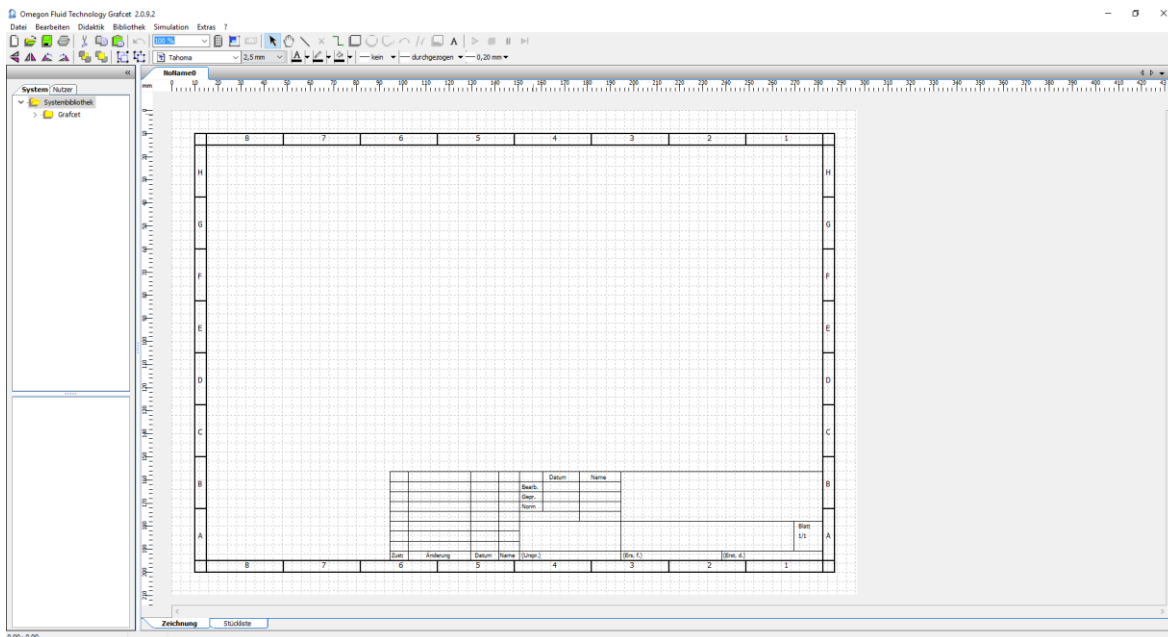
Un cop s'ha explicat sense profunditzar el funcionament del programa, es poden mencionar algunes característiques del programa que es poden considerar dificultoses per l'usuari o que li poden causar problemes a l'hora d'elaborar els diagrames com ells volen.

Tot i que el programa és bastant complet i permet la simulació, seria convenient que el caixetí dels diagrames creats es pogués eliminar ja que en molts casos no és interessant i ocupa espai necessari. Es coneix que les connexions entre diferents elements s'ha de fer amb els cables corresponents i no amb la connexió directa entre ells, cosa que pot arribar a ser molest ja que s'ha de tornar en moltes ocasions a canviar de llibreria a la barra d'elements per fer les annexions necessàries. En el cas de les transicions, per mostrar que es tracta de flanc de pujada o de baixada la seva validació, el programa utilitza les nomenclatures [FP] (flanc positiu) i [FN] (flanc negatiu) en comptes d'utilitzar un símbol representatiu més clar, encara que aquest es podria obtenir amb les eines de dibuix proporcionades per l'aplicació. Per últim, l'aplicació tampoc permet exportar taules en algun format com el que utilitza Microsoft Excel.

Aquestes característiques anteriors serien les úniques qüestionables al programa ja que realment aquest funciona de forma molt adequada encara que sigui més complex que l'anterior aplicació mostrada en el subapartat 4.1.

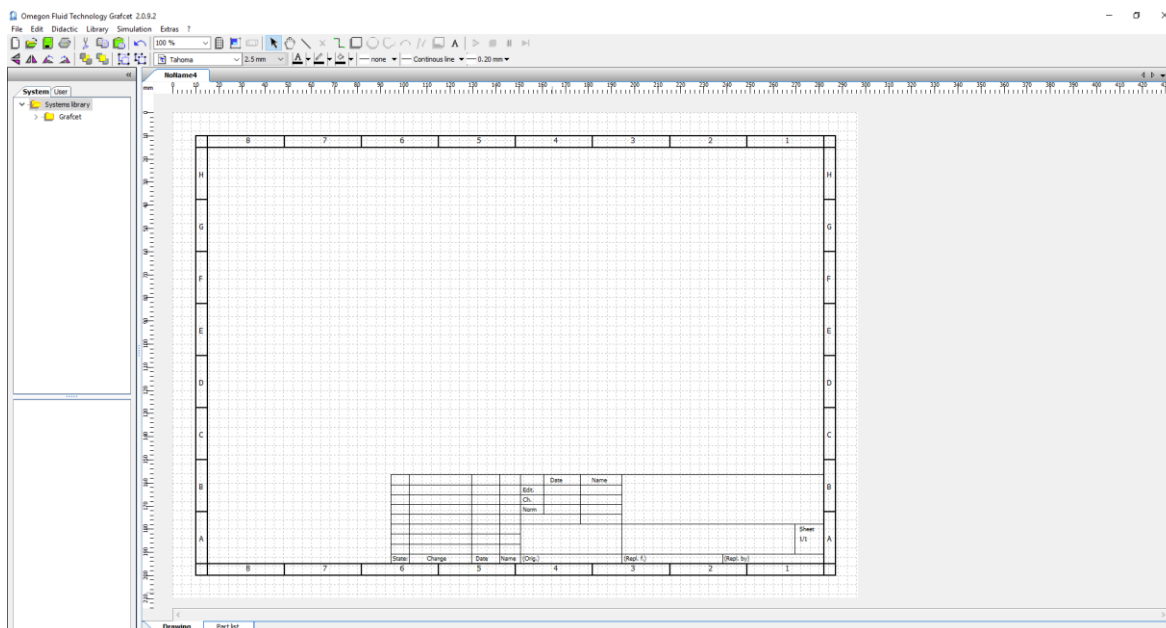
### 4.3. Omegon Fluid Technology Grafcet 2.0.9.2

L'últim exemple a analitzar és un mòdul del programa Omegon Fluid Technology de distribució gratuïta centrat en l'edició de diagrames GRAFCET. La versió completa a més del tema d'edició dels GRAFCETs conté llibreries amb elements pneumàtics i hidràulics i habilita la simulació. En aquest cas, només es parla de la versió reduïda, en concret del programa Omegon Fluid Technology Grafcet 2.0.9.2. Un cop s'obre l'aplicació es troba la següent finestra:



**Figura 4.3.1.** Captura d'imatge de la finestra d'inici de Omegon Fluid Technology Grafcet 2.0.9.2  
(Font: Pablo Mañas)

Encara que en un primer inici el programa es trobi en alemany, l'última casella dintre de la pestanya "Extras" permet seleccionar l'idioma entre l'anglès, l'alemany o l'holandès. Un cop es fa el canvi de llenguatge a l'anglès, es pot observar la següent finestra d'inici:



**Figura 4.3.2.** Captura d'imatge de la finestra d'inici de Omega Fluid Technology Grafcet 2.0.9.2  
(Font: Pablo Mañas)

Això permet que els usuaris que no coneguin la llengua alemanya i sí l'holandesa o l'anglesa puguin fer servir el programa de forma eficaç sense ajuda d'un diccionari. A continuació, es mostren les característiques punt per punt de la versió del programa en qüestió:

- **Pestanya "File":** Es troben totes les funcions corresponents a crear un nou projecte, obrir un projecte, obrir exemples de diagrames creats pels propis dissenyadors, dirigir-se al diagrama que ha sigut utilitzat en última instància, obtenir còpia de seguretat, guardar el document, exportar (només en la versió completa), tancar, tancar tots els projectes, imprimir, configurar el programa i sortir de l'aplicació. Moltes d'aquestes funcions es troben repartides a la pantalla principal en icones en barres d'elements.
- **Pestanya "Edit":** Aquí es troben les eines d'edició tals com desfer (acció anterior), tallar, copiar, enganxar, esborrar, seleccionar tots els elements, agrupar diversos objectes i dividir-los, bloquejar objecte seleccionat amb la fi que no es pugui moure i desbloquejar-lo, rotar 90° cap a dreta i esquerra, amirallar horitzontal i verticalment l'objecte, alinear grup d'objectes i, per últim, inserta imatge, tot i que no es troba disponible en la versió que s'està explicant. Així com en el cas anterior, es troben símbols corresponents a aquestes funcions en diferents barres d'elements.
- **Pestanya "Didactic":** Dintre d'aquesta pestanya apareix un diccionari d'anglès-alemany i alemany-anglès molt útil per aquella persona que no coneix l'idioma i pugui aprendre les paraules en la llengua desconeguda. A més, si està seleccionada la llengua en alemany es troba un manual en aquest idioma que explica els GRAFCETs de manera introductòria.



- **Pestanya “Library”:** Al programa complet en aquesta pestanya es troba la llibreria amb totes els elements disponibles. Com en aquest cas es treballa amb la versió de lliure distribució aquestes llibreries es troben inhabilitades i, per tant, no és necessari afegir res més dintre d’aquest subapartat.
- **Pestanya “Simulation”:** Així com en la pestanya anterior, aquesta tampoc es pot utilitzar dintre d’aquesta versió, ja que no es disposen de les llibreries adequades per poder fer les simulacions, per tant, no es podran simular els diagrames construïts.
- **Pestanya “Extras”:** En aquesta pestanya es troben diverses funcions que completen el programa. La primera casella és una eina que serveix com a bloc de notes per a que l’usuari pugui escriure el que vulgui i es quedi guardat. La segona obre una finestra de configuració que permet a l’usuari modificar la mida del paper en que està treballant. La tercera serveix per a modificar les dades del caixetí que s’observa al nou diagrama generat, el qual no es podrà eliminar. Per últim es troba la casella de selecció d’idioma ja mencionada amb anterioritat.
- **Pestanya “?”:** Dintre d’aquest subapartat es troba tota l’ajuda del programa. La primera funció obre el manual sencer del programa en anglès que explica totes les característiques del programa de manera més extensa. El següent botó mostra una taula comparativa entre les diferents versions del programa i altres afers sobre llicències i compres. Els dos següents botons obren informació també sobre la llicència del programa i l’últim mostra la versió del programa, la direcció de l’empresa i el nom del director de l’aplicació.
- **Altres característiques i barres d’elements:** A més de les funcions mostrades anteriorment, és necessari explicar algunes característiques del programa que apareixen en pantalla i les quals no es mencionen dins de les pestanyes. Per exemple, a la part superior del programa es pot observar una barra d’elements com la següent:



**Figura 4.3.3.** Barra d’elements de Omega Fluid Technology Grafset 2.0.9.2

(Font: Pablo Mañas)

El primer requadre de la figura anterior representa el Zoom del diagrama en que s’està treballant i es pot modificar el número per tal d’ampliar o reduir la imatge mostrada. Continuant amb l’ordre d’esquerra a dreta es troba un botó que permet activar o desactivar la graella del diagrama i al costat un botó que bloqueja l’objecte a la graella A l’altra banda es troba una icona d’un punter que en seleccionar-se farà que el cursor treballi com a seleccionador, una icona d’una mà que permetrà a l’usuari arronsar el diagrama per tal de moure’s a través d’ell, un símbol d’una línia recta en diagonal que serveix per dibuixar línies rectes, un botó amb un dibuix d’una línia en zig-zag que permet fer unions de connexions de

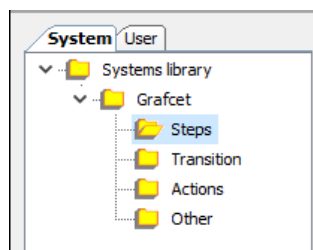
forma manual, una icona d'un quadrat que permet dibuixar rectangles i quadrats al diagrama i, per últim una lletra "A" que permet introduir un text qualsevol. Aquest text podrà tenir el tipus de lletra, la mida i el color que es seleccioni als requadres que es troben a l'extrem inferior esquerre de la figura 4.3.3. A la barra inferior també es podrà escollir el tipus de línia dibuixada tant de l'interior com dels extrems, el gruix i el color. Pel que fa als objectes amb àrees a dintre, també es podrà pintar aquest interior d'un color determinat.

En el cas que es solapin objectes, es podran apilar de la manera que es vulguin amb ajuda d'aquests dos botons que es mostren en la imatge següent:



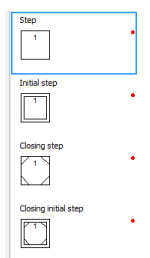
**Figura 4.3.4.** Enviar objecte al fons i portar objecte al front de Omegon Fluid Technology Grafcet 2.0.9.2 (Font: Pablo Mañas)

Dintre d'aquest subapartat cal parlar de la llibreria que utilitza aquesta versió del programa. Així doncs, s'observa al marge esquerre de la finestra principal el següent quadre:



**Figura 4.3.5.** Quadre llibreria d'objectes de Omegon Fluid Technology Grafcet 2.0.9.2 (Font: Pablo Mañas)

A la figura anterior es pot veure com l'usuari pot fer clic als diferents elements que componen un GRAFCET. En fer clic a qualsevol carpeta apareixeran a sota els diferents elements disponibles dintre d'aquest apartat. Per exemple si s'ha seleccionat la carpeta "Steps" la qual correspon a les etapes, es mostra un requadre al vèrtex inferior esquerre com el següent:



**Figura 4.3.6.** Quadre objectes de la carpeta "Steps" de Omegon Fluid Technology Grafcet 2.0.9.2 (Font: Pablo Mañas)

Amb els objectes que apareixen l'usuari podrà crear el GRAFCET que desitgi arronsant-los al diagrama.

Un cop s'hagi creat el diagrama o mentre es duu a terme el procés l'usuari pot observar la llista d'objectes que es troben dintre del GRAFCET fent clic a la part inferior del programa a "Part list" així com es pot observar a la figura que es mostra a continuació:

Position	Quantity	Unit	Description in...	Legend	Remark	ID-No.
1	1	Piece		Initial step		
2	1	Piece		Transition		
3	1	Piece		Synchronisation		
4	1	Piece		Step		
5	1	Piece		Continuous action		
6	1	Piece		Transition		
7	1	Piece		Step		
8	1	Piece		Continuous action		
9	1	Piece		Transition		
10	1	Piece		Step		
11	1	Piece		Continuous action		
12	1	Piece		Transition		
13	1	Piece		Step		
14	1	Piece		Step		
15	1	Piece		Continuous action		
16	1	Piece		Transition		
17	1	Piece		Step		
18	1	Piece		Continuous action		
19	1	Piece		Step		
20	1	Piece		Transition		
21	1	Piece		Step		
22	1	Piece		Continuous action		
23	1	Piece		Transition		
24	1	Piece		Synchronisation		
25	1	Piece		Transition		
26	1	Piece		Synchronisation		
27	1	Piece		Step		
28	1	Piece		Transition		
29	1	Piece		Step		
30	1	Piece		Continuous action		
31	1	Piece		Step		
32	1	Piece		Continuous action		
33	1	Piece		Synchronisation		
34	1	Piece		Transition		
35	1	Piece		Step		
36	1	Piece		Continuous action		
37	1	Piece		Transition		

**Figura 4.3.7.** Exemple llista d'elements d'un GRAFCET de Omega Fluid Technology Grafcet 2.0.9.2 (Font: Pablo Mañas)

Aquesta llista pot ser útil per a l'usuari, encara que també pot obtenir una més general amb la quantitat d'objectes de cada tipus.

- **Esdeveniments de teclat i ratolí:** Per fer funcionar un programa és important saber com interacciona el ratolí i el teclat amb ell. Així com la resta de programes que ja s'han explicat amb el clic esquerre del ratolí es pot seleccionar objectes, moure'ls, traçar línies i altres formes de dibuix. Amb doble clic s'obren les propietats de cada objecte les quals es poden modificar i varien entre ells. Amb el clic dret sobre del diagrama apareix una pestanya amb les opcions d'edició que ja s'han explicat amb anterioritat. La roda del ratolí permet fer les accions de Zoom sobre el diagrama actiu.

Pel que fa al teclat, obviant les dreceres que s'activen mitjançant macros, únicament es fan servir les fletxes per moure els objectes seleccionats i la tecla Supr per esborrar els elements.

Encara que no s'ha pogut provar la versió completa del programa, amb aquestes funcions es poden crear múltiples GRAFCETS i per això ha sigut inclòs en aquest apartat de la memòria. Així com s'ha fet

amb els dos exemples anteriors explicats als subapartats previs, a continuació es parlarà d'algunes característiques millorables o que no apareixen en la versió d'aquest programa.

El fet que inicialment el programa s'iniciï en alemany pot arribar a ser confús per als usuaris ja que majoritàriament entendran millor l'anglès que no pas l'altre idioma i en cas que ho necessitin canviar hauran de mirar prèviament el manual ja que probablement no trobaran fàcilment la manera i, per tant, perdran temps.

Centrant-se ara en aspectes de l'editor en sí, de la mateixa manera que succeïa al programa anterior, es creu que el caixetí hauria de ser opcional, és a dir, que es pogués treure en cas que l'usuari així ho desitgés. També es troba a faltar alguna manera de crear taules amb les transicions i les accions corresponents que després podran ser útils per definir les entrades i sortides d'un PLC, per exemple. En aquesta ocasió tampoc es pot exportar cap taula amb dades del GRAFCET ja que la versió no ho permet, tot i que en la versió completa es podria exportar la taula que s'ha mostrat a la figura 4.3.7.

Sota l'opinió de l'autor aquestes serien les úniques característiques que es podrien tenir en compte en un futur per a possibles millores. Tot i així, la versió del programa compleix la funció d'editor de GRAFCET i, per tant, es considera adequat a les necessitats.

#### 4.4. Taula comparativa

Amb la fi de destacar els punts forts i dèbils de cadascun dels programes descrits anteriorment se ha decidit crear una taula amb uns criteris que permeten ressaltar les característiques més importants de cada una de les aplicacions.

Per tant, a continuació es mostra una taula amb les especificacions i la importància de cada criteri explicat a mode de resum:

	<b>Excel·lent (5-4)</b>	<b>Suficient (3-2)</b>	<b>Insuficient (1-0)</b>
<b>1. Correcció</b>	El programa satisfà totes les necessitats de l'usuari	El programa compleix gran part de les necessitats de l'usuari	El programa compleix únicament algunes necessitats de l'usuari
<b>2. Usabilitat</b>	El programa és molt senzill d'aprendre a utilitzar	El programa és senzill d'utilitzar excepte algunes funcions	El programa és complex d'utilitzar i es necessita d'ajuda prèvia

<b>3. Extensibilitat</b>	El programa permet afegir més característiques de forma fàcil	El programa permet afegir característiques de manera moderada en quant a dificultat	El programa permet afegir més característiques però de forma difícil
<b>4. Eficiència</b>	El programa consumeix de forma eficaç la memòria	El programa té alguns problemes de consum de memòria	El programa presenta molts problemes de consum de memòria
<b>5. Fiabilitat</b>	El programa funciona perfectament en execució	El programa pot tenir algun error durant l'execució	El programa falla durant l'execució
<b>6. Portabilitat</b>	Permet dur a terme totes les funcions en qualsevol plataforma	Permet dur a terme totes les funcions en algunes plataformes	Permet dur totes les funcions en una única plataforma
<b>(Característiques centrades en l'editor de grafcet)</b>			
<b>7. Exportació</b>	El programa permet exportar els documents creats en diferents formats	El programa permet exportar els documents en pocs formats	El programa permet exportar els documents en un únic format
<b>8. Funció de simulació</b>	El programa permet simular els GRAFCETS creats	El programa permet simular els GRAFCETS a partir d'altres diagrames	El programa no permet simular GRAFCETS
<b>9. Taula d'entrades/sortides</b>	El programa permet crear taules de dades amb les entrades i sortides al sistema	El programa permet crear taules de dades però difícilment exportables	El programa no permet de cap manera crear taules
<b>10. Creació d'enllaços</b>	El programa ajuda a crear els enllaços entre els objectes de manera còmoda	El programa ajuda a crear els enllaços entre els objectes però no fàcilment	El programa no ajuda a crear els enllaços entre els objectes

<b>11. Control d'enllaços</b>	El programa mostra una taula amb les connexions dels diferents elements i els enllaços	El programa mostra els diferents elements del grafcet però no els enllaços	El programa no mostra cap informació dels enllaços entre elements
<b>12. Funcions de vista (zoom)</b>	El programa té múltiples funcions per ampliar i reduir la pantalla	El programa té alguna funció per ampliar i reduir la pantalla	El programa no té cap funció per ampliar i reduir la pantalla
<b>13. Existència de temporitzadors</b>	El programa permet afegir temporitzadors a la simulació	El programa permet afegir temporitzadors a la simulació però no es poden pausar	El programa no permet afegir temporitzadors a la simulació
<b>14. Múltiples finestres</b>	L'usuari pot crear i editar diferents documents al mateix temps amb múltiples finestres	-	L'usuari no pot crear i editar diferents documents al mateix temps amb múltiples finestres
<b>15. Altres funcionalitats</b>	El programa disposa de moltes altres funcions a més de la creació i la edició de GRAFCETs	El programa disposa d'algunes funcions a més de la creació i la edició de GRAFCETs	El programa disposa únicament de funcions sobre la creació i la edició de GRAFCETs

**Figura 4.4.1.** Taula d'avaluació de criteris (Font: Pablo Mañas)

Per tal de mantenir un ordre en la correcta avaluació dels criteris descrits a la figura anterior es numeren els programes de la manera següent:

1. Editor de GRAFCETs – Pablo Mañas
2. Dia 0.97.2
3. CADeSIMU v3
4. Omegon Fluid Technology Grafcet 2.0.9.2

Ara que ja s’han decidit els criteris d’avaluació dels programes descrits anteriorment i els programes sotmesos a avaluació, la taula següent mostra la valoració personal de cada una de les aplicacions incloent la elaborada per l’autor d’aquesta memòria.

Criteris \ Programes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Media
1	3	4	2	3	3	-	4	4	4	4	5	0	3	0	3	3
2	3	5	-	5	5	4	4	0	1	5	1	5	0	5	4	3,4
3	4	5	-	5	5	5	3	3	3	5	1	4	5	5	5	4,1
4	4	4	-	4	4	3	3	0	0	4	3	5	0	5	3	3

**Figura 4.4.1.** Taula de valors comparativa (Font: Pablo Mañas)

Cal dir que la valoració es subjectiva i algunes dades podrien variar en funció de la experiència de l’usuari amb cadascun dels programes. Tambés es necessari afegir que en el cas de l’últim programa al tractarse d’una versió no completa és normal que tingui la puntuació més baixa ja que no s’ha pogut provar el mode complet que segurament faria que es valorés de forma més positiva en molts dels aspectes.

A més dels programes descrits en aquest subapartat n’hi ha alguns exemples més d’editors de grafsets, però tots acaben consistint en el mateix. Solen ser programes de pneumàtica i hidràulica com per exemple FluidSIM que també permet construir els GRAFCETS que depenen dels elements hidràulics i pneumàtics. Pràcticament en tots el casos de programes d’aquest estil i que tenen una funció per simular, no es centren en els GRAFCETS en sí, sinó que van més enllà amb la creació i simulació d’enllaços pneumàtics, hidràulics, elèctrics o electrònics.

Per últim, és necessari mencionar que per la complexitat que podria resultar imitar algun dels programes anteriors, el creador del primer programa va decidir centrar-se únicament en l’edició, creació, simulació i exportació de GRAFCETS sense entrar en altres aspectes com la implementació d’elements hidràulics, electrònics, elèctrics, etc.

## 5. Metodologia de treball

Abans de concloure la memòria del treball de fi de grau cal parlar de quin ha sigut el mètode portat a terme durant la realització d'aquest. Això engloba la metodologia emprada en l'elaboració de l'editor de GRAFCETs desenvolupat. Per tant, en aquest apartat es tracta de donar a conèixer aquests procediments.

Com ja s'ha citat amb anterioritat, el programa realitzat ha sigut programat amb l'entorn de desenvolupament de Microsoft Visual Studio 2017 amb el llenguatge de Visual Basic. Tot i que es tracta d'un llenguatge dirigit per esdeveniments guarda relació amb la programació orientada a objectes (POO) i, per això, inicialment es va crear un diagrama de classes adient que va anar evolucionant durant la realització de l'aplicació el qual es troba a l'annex A0. Aquesta progressió es deu en certa part a haver seguit una metodologia del tipus espiral, ja que provoca que cada cert temps es torni a revisar tot el programa per veure les millores o errors possibles i, per tant, es produeixi una modificació dels diferents elements.

Aquest mètode de treball consisteix en anar recorrent una espiral cada cert temps al voltant de 4 activitats bàsiques les quals son: determinació d'objectius, anàlisi de riscos, desenvolupament i prova, i planificació. La figura 5.1 mostra aquestes tasques de forma gràfica. La primera tasca consisteix en fixar els requisits del programa i les restriccions, la segona d'analitzar els possibles riscos (en general errors d'execució) i idear alternatives per evitar-los, la tercera en desenvolupar i avaluar els resultats mitjançant proves i la quarta en planificar de nou en funció dels problemes surgits. D'aquesta manera cada cop que s'afegia una funció nova al programa calia recórrer una volta d'espiral per tal de verificar el bon funcionament. Per tant, en cas que es volguessin aplicar millores caldria tornar a fer les quatre tasques descrites anteriorment. Tot i així sempre poden sorgir problemes no esperats que en el cas que apareguin caldrà mirar amb detall.



Figura 5.1. Metodologia en espiral (Font: Pablo Mañas)



Poden existir variacions en el mètode amb més o menys ventatges però des de l'inici del projecte es va decidir realitzar el treball d'aquesta forma i, per tant, no es va canviar. Per últim cal dir que encara que és un mètode llarg de desenvolupar ja que gasta molt de temps haver de preveure els errors que poden succeir també els minimitza perquè ja es pensen prèviament maneres d'arreglar-lo.







## Conclusions

Amb l'evolució del treball s'ha pogut exposar a grans trets les característiques bàsiques d'un GRAFCET juntament amb els seus elements. A partir d'aquestes dades s'ha elaborat un programa que permet l'edició d'aquests diagrames, la seva exportació en diversos formats i la simulació, objectius principals que es tractaven abordar des d'un inici. Finalment, s'ha comparat el resultat del treball de programació amb altres aplicacions actuals amb la fi de valorar-lo dintre del món dels editors de GRAFCETS.

Pel que fa al nucli del treball, que és l'aplicació realitzada, cal dir que com ja s'ha indicat hi ha marge de millora, o sigui, es poden seguir programant funcions que vagin completant el programa i millorant-lo. Tot i així, en l'estat en que es troba el programa elaborat es compleixen les expectatives que es tenien inicialment, és a dir, presta els requisits que es consideren necessaris per ser un editor de GRAFCETS.

En relació amb el que s'ha descrit anteriorment, en un futur no es descarta la idea de continuar amb el projecte, ja sigui de forma individual o amb ajuda externa amb la finalitat d'obtenir un programa robust amb el màxim de funcions possibles que permeti a l'usuari editar les seves creacions de la forma més còmoda i simular-les de manera adient. També es podrien millorar aspectes com l'interfície del programa, el disseny o la correcció d'algunes funcions per optimitzar recursos.

Per concloure finalment, cal afegir que el treball de fi de grau s'ha centrat més en explicar el programa realitzat per tal que l'usuari pugui entendre les funcions i la manera en que pot fer ús de l'editor i no tant en la part de programació efectuada ja que en un inici es va plantejar d'aquesta manera i, per tant, es desvinculava dels objectius principals incipients. Tot i així, s'ha vist necessari aprofundir en aquest últim tema per deixar a qualsevol lector satisfet.



## Pressupost i/o Anàlisi Econòmica

En aquest capítol es detallen, de forma orientativa, els costos totals del desenvolupament de l'aplicació, dividits entre costos de disseny, costos dels materials utilitzats i, finalment, un pressupost amb el cost total del projecte.

### Costos de disseny

Dintre d'aquest subapartat es tracten els costos generats per les hores de treball de l'enginyer al desenvolupar el projecte. Aquests es poden desglossar en diverses activitats que han intervingut en el projecte:

- Anàlisi: Conté els costos referents al temps dedicat a l'aprenentatge de l'entorn de programació, l'estudi de les diferents funcionalitats del programa i a la recerca d'informació adient per al desenvolupament del treball.
- Disseny: Inclou els costos referents al temps destinat al plantejament de solucions als problemes sorgits.
- Informe: Es basa en les hores emprades en l'elaboració de l'informe del projecte.
- Programació: Es basa en les hores dedicades a la codificació del programa.

La taula següent resumeix tot el conjunt de costos referents a les activitats anteriors:

Concepte	Hores dedicades	Preu per hora (€/h)	Cost (€)
Anàlisi	120	12	1440
Disseny	250	15	3750
Informe	50	12	600
Programació	180	12	2160
COST TOTAL			7950

**Figura 1.** Taula de costos de disseny.

## Costos materials

A més dels costos de disseny, cal aplicar els costos referents als materials emprats durant el desenvolupament del projecte. Aquests són el cost de l'ordinador necessari, la llum consumida per les hores de treball realitzades (considerant el preu mig de la llum a 0,135 €/kWh i una potencia de consum mitjana de 230 W/h), l'internet (considerant una mitja de 46 € al mes) i altres despeses com fulls de paper i tinta d'impressora. La taula següent mostra aquestes dades:

Concepte	Hores/Mesos	Preu	Cost (€)
Ordinador	-	-	800
Llum	600 hores	0,03105 (€/h)	18,63
Internet	8 mesos	46 (€/mes)	368
Altres	-	-	60
<b>COST TOTAL</b>			<b>1246,63</b>

**Figura 2.** Taula de costos materials.

## Pressupost global

Finalment, el pressupost total del projecte queda resumit en la següent taula:

Pressupost	Preu total (€)
Costos de disseny	7950
Costos materials	1246,63
Total (sense benefici)	9196,63
Total (5% de benefici)	9656,46
Total (aplicant IVA 21%)	11684,32

**Figura 3.** Taula de pressupost global.



## Bibliografia

En aquest apartat es mostra la bibliografia consultada a l'hora de realitzar el treball. Per tal de mantenir un ordre, l'autor ha decidit desglossar-la en diferents subapartats perquè sigui més fàcil de comprendre.

### Recursos utilitzats per la programació:

- [1] Col·laboradors de Windows. Docs.microsoft.com (guia). *Guía de Visual Basic* [online]. Disponible a: <https://docs.microsoft.com/es-es/dotnet/visual-basic/> [Consultat al setembre de 2018]
- [2] Col·laboradors de Windows. Docs.microsoft.com (guia). *Filtrar para crear formularios secundarios MDI*. (2017). [online]. Disponible a: <https://docs.microsoft.com/es-es/dotnet/framework/winforms/advanced/how-to-create-mdi-child-forms> [Consultat al setembre de 2018]
- [3] Edmundo González. YouTube.com (vídeo). *Mi primer Formulario MDI y Modal en Visual Basic*. (2015). [online]. Disponible a: <https://www.youtube.com/watch?v=3ONd9KR0rN0> [Consultat en setembre de 2018]
- [4] Grijalvaromero. YouTube.com (vídeo). *Drag And Drop en Visual Basic .NET*. (2015). [online]. Disponible a: <https://www.youtube.com/watch?v=Csr5wipGNRO> [Consultat al setembre de 2018]
- [5] AndyNakamura. Social.msdn.microsoft.com (fòrum de programació). *Protected Overrides Function ProcessCmdKey return values*. (2016). [online]. Disponible a: <https://social.msdn.microsoft.com/Forums/en-US/a068941a-1c27-4d97-a31c-da01a594dd1d/protected-overrides-function-processcmdkey-retun-values?forum=vbgeneral> [Consultat al setembre de 2018]
- [6] Col·laboradors de Windows. Docs.microsoft.com (guia). *Keys Enum*. [online]. Disponible a: <https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.keys?view=netframework-4.7.2> [Consultat al novembre de 2018]
- [7] Iconfinder (web de iconos). *Add, cancel, delete, edit, guardar, refresh, save icon*. [online]. Disponible a: [https://www.iconfinder.com/icons/27876/add\\_cancel\\_delete\\_edit\\_guardar\\_refresh\\_save\\_icon](https://www.iconfinder.com/icons/27876/add_cancel_delete_edit_guardar_refresh_save_icon) [Consultat al desembre de 2018]
- [8] Enrique M. Montejo. Social.msdn.microsoft.com (fòrum de programació). *Necesito guardar operadores lógicos en variable para usarlos en operación de cálculo*. (2008). [online]. Disponible a: <https://social.msdn.microsoft.com/Forums/es-ES/8cda2e72-2dc1-44ce-9bb4->



[19] ELHALCON1970. YouTube.com (vídeo). *Puente grua con grafcet*. (2017). [online]. Disponible a: <https://www.youtube.com/watch?v=sldemoDSbBE> [Consultat al setembre de 2018]

[20] ELECTROTECNIA. YouTube.com (vídeo). *PLC S7-1200 SIEMENS EN CADE SIMU simulación, programacion y funcionamiento*. (2017). [online]. Disponible a: <https://www.youtube.com/watch?v=IASCpYWZAT8> [Consultat al setembre de 2018]