

## 5. Manual de usuario

Este pequeño manual del GenModen (Generador de modelos navegacionales) se ha creado para guiar a los usuarios que decidan utilizar esta aplicación. El manual está dividido en dos partes. La primera sirve para que el usuario se familiarice con el entorno que trabajará. La segunda explica las funciones que tiene la aplicación.

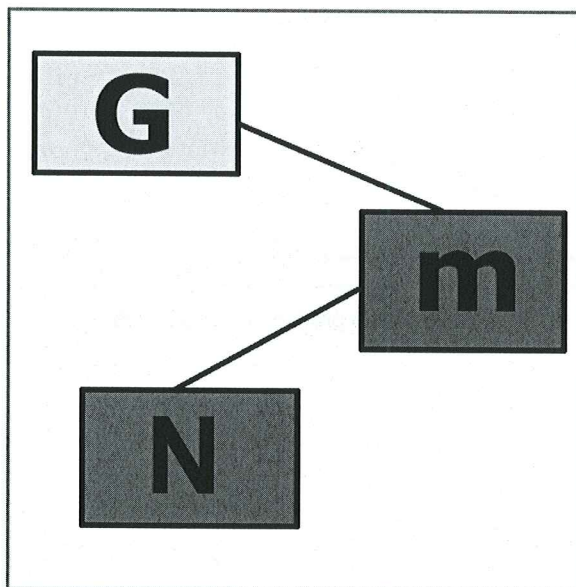


Fig. 5.1 Logotipo de la aplicación

## 5.1 Introducción al área de trabajo

---

La ventana principal de la aplicación esta dividida en tres zonas.

1. Menú
2. Pantalla principal
3. Barra de estado

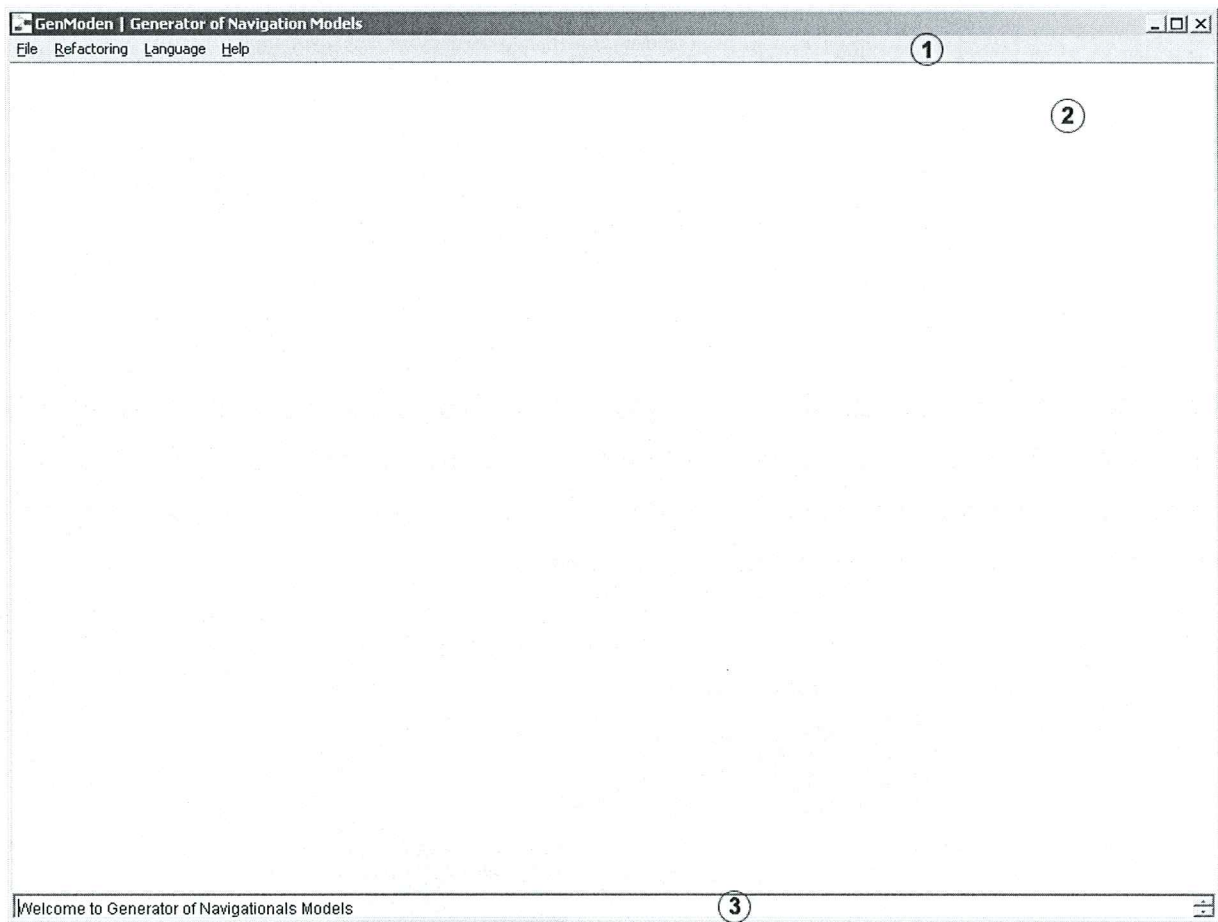


Fig. 5.2 Entorno gráfico de la aplicación

## 5.1.1 Menú

El menú es el elemento encargado de guiar al usuario por todas las opciones de la aplicación. Esta dividido en cuatro submenús.

**Archivo:** Aquí es donde se encuentran las opciones de abrir o cerrar modelos, guardarlos o salir de la aplicación.

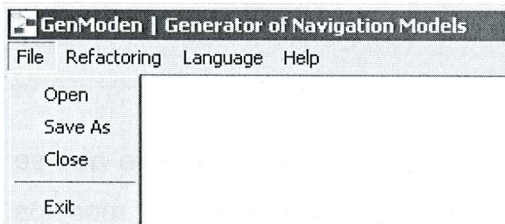


Fig. 5.3 Menú archivo

**Refactoring:** En este menú se pueden encontrar todas las opciones referentes a las refactorizaciones que el usuario puede realizar en el modelo.

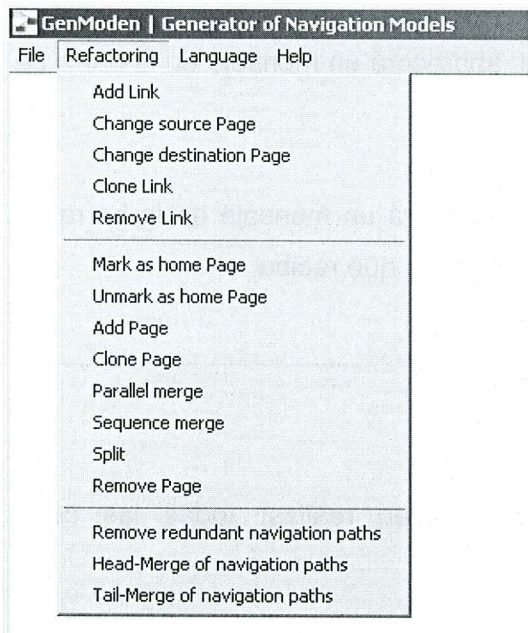


Fig. 5.4 Menú refactoring

**Language:** Se puede cambiar de idioma.

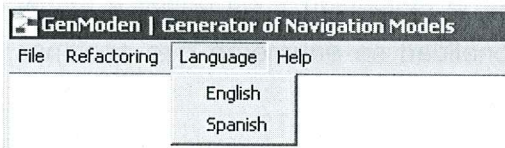


Fig. 5.5 Menú idioma

**Ayuda:** Menu ayuda, donde el usuario podrá consultar el manual y la información de la aplicación.

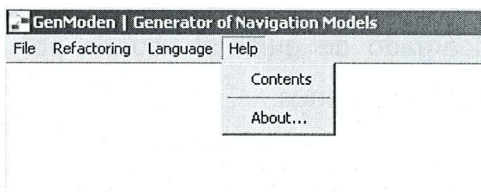


Fig. 5.6 Menú Ayuda

## 5.1.2 Pantalla principal

---

Lo primero que ve el usuario es la pantalla principal vacía (figura 5.1). Lo primero que se debe hacer es escoger un modelo para poder realizar la generación de modelos navegacionales.

## 5.1.3 Barra de estado

---

Cuando se haya generado el modelo de navegacional, aparecerá un mensaje en la barra de estado indicando que se ha realizado el modelo.

Siempre que se haga una acción en la aplicación se mostrará un mensaje en la barra de estado para indicar la respuesta de la aplicación a las acciones que reciba.



Fig. 5.9 Barra de estado

Una vez ya se tiene el modelo navegacional se pueden realizar todas las otras funcionalidades del programa.

## 5.2 Funcionalidades

---

En este apartado se explican que funcionalidades tiene la aplicación y los pasos a seguir para que se realicen correctamente. Para cada funcionalidad se enumeran que opciones puede haber y cuales son los resultados que se obtendrán.



### 5.2.1 Generar modelo navegacional

Para generar el modelo navegacional, el usuario tiene que ir al menú File y seleccionar la opción Open. Se abrirá una opción como se muestra en la siguiente figura para que acepte el archivo XML que contiene la información del modelo de datos.

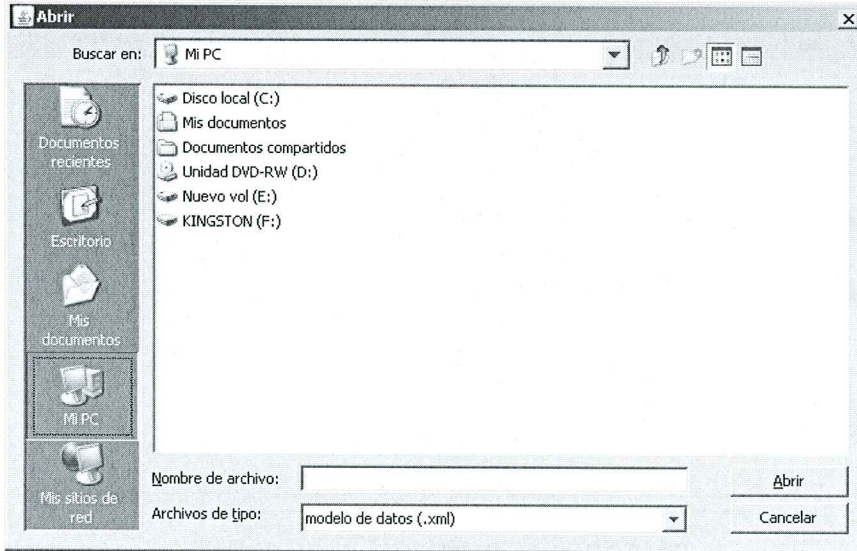


Fig. 5.7 Selección del modelo de datos.

Una vez escogido el modelo de datos, se generará el modelo navegacional y se cargará en la pantalla principal.

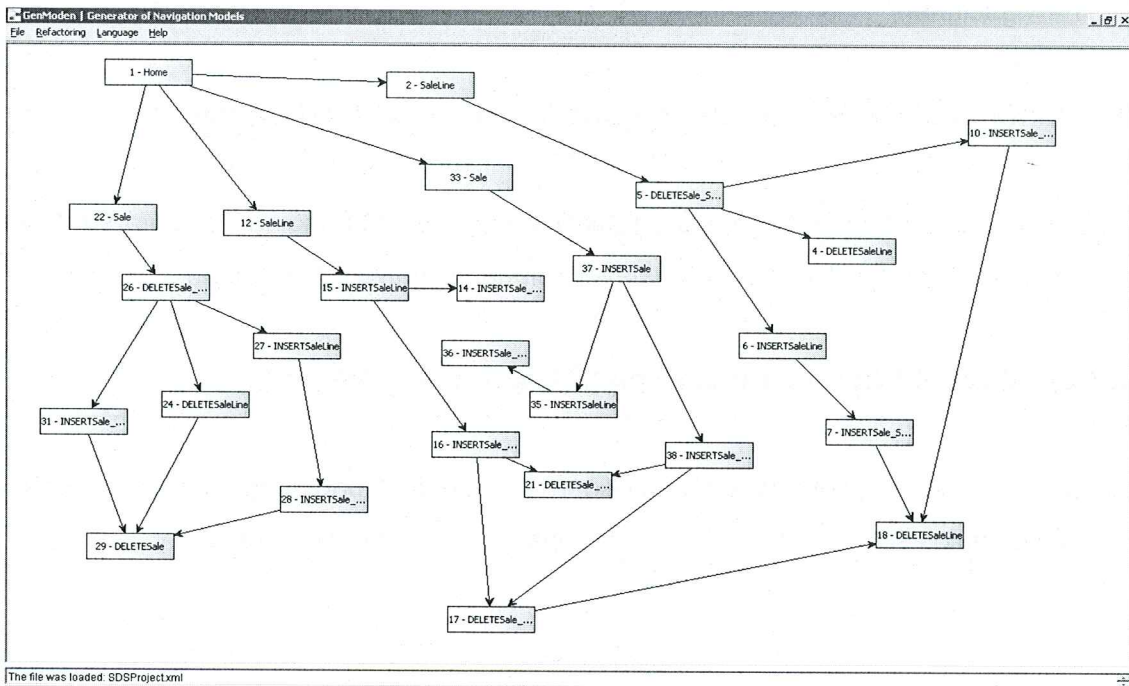


Fig. 5.8 Modelo navegacional.

## 5.2.2 Refactorings

---

Una vez tenemos el modelo navegacional, el siguiente paso es aplicar los refactorings que se deseen si se desea aplicar alguno.

Los diferentes refactorings los podemos encontrar en los submenús del menú Refactoring.

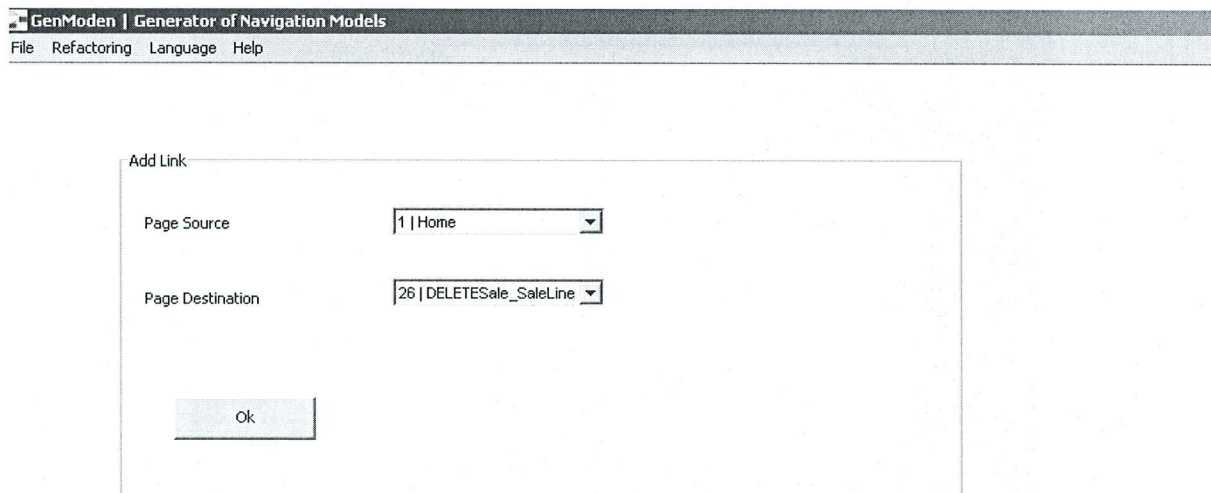


Fig. 5.9 Pantalla de aplicación de refactorings

### 5.2.2.1 Añadir Link (Add Link)

---

Para añadir un link se escoge la página de origen y la de destino. Se añade el nuevo link al modelo navegacional.

### 5.2.2.2 Cambiar el destino de una página (Change destination page)

---

Se indica el link que se quiere cambiar su destino y se indica la página que será su destino. Se realizan los cambios sobre el modelo navegacional y vuelve a mostrarse.

### 5.2.2.3 Cambiar el origen de una página (Change source page)

---

Se indica el link que se quiere cambiar su origen y se indica la página que será su origen. Se realizan los cambios sobre el modelo navegacional y vuelve a mostrarse.

#### **5.2.2.4 Clonar Link (Clone Link)**

---

Se indica que link es el que se desea clonar. Se realizan los cambios en el modelo navegacional y se muestran los cambios en la pantalla.

#### **5.2.2.5 Eliminar Link (Remove Link)**

---

Se indica que link se desea eliminar. Para que el link se elimine la página a la que hace enlace, su página de destino, ha de tener algún otro link que llegue a ella. Si se cumplen estas condiciones, se aplica el cambio y se muestra el modelo navegacional cambiado.

#### **5.2.2.6 Marcar como Home (Mark as Home page)**

---

Se indica que página se desea que sea también Home. Se realizan los cambios y se muestra por pantalla el modelo navegacional cambiado.

#### **5.2.2.7 Desmarcar como Home (Unmark as Home page)**

---

Se indica que página home se desea quitar como esta opción. Si hay alguna otra página Home se realiza el cambio y se muestra el modelo navegacional cambiado.

#### **5.2.2.8 Añadir página (Add page)**

---

Se escoge una página para añadir teniendo en cuenta que tiene que seguir siendo coherente con el modelo de datos introducido, seguir siendo correcto y completo. Como la página no puede quedar suelta, se tiene que añadir un link para que se pueda llegar a esa página. Si es así, se aplica el cambio y se muestra el modelo navegacional.

#### **5.2.2.9 Clonar página (Clone page)**

---

Se indica que página se quiere clonar. La aplicación genera una página con sus links de entrada y salida idénticos. Se realizan los cambios en el modelo navegacional y se muestra.



#### **5.2.2.10 Fusión paralela (Parallel merge)**

---

Se indican que dos páginas se desean fusionar. Las dos páginas a fusionar han de ser paralelas. Cuando se realiza la fusión, la página resultante tendrá de links de entrada los de las dos páginas fusionadas, y de salida, los de las dos páginas. Una vez realizado los cambios se muestra el modelo.

#### **5.2.2.11 Fusión de una secuencia (Sequence merge)**

---

Se indican que dos páginas se desean fusionar. Han de estar una seguida de otra. Una vez fusionadas, se muestran los cambios del modelo navegacional.

#### **5.2.2.12 División (Split)**

---

Se selecciona que página se desea dividir en dos. Se generan a continuación dos páginas unidas por un link. Todos los links de entrada, serán links de entrada de la primera página de la nueva secuencia y los de salida son los links de salida de la segunda página de la secuencia. Una vez realizados los cambios se muestra el modelo de datos.

#### **5.2.2.13 Eliminar página (Remove page)**

---

Se selecciona la página que se desee eliminar. Se elimina esa página y los links que tiene de entrada y salida. Si uno de los links de salida, si tiene, apunta a una página que quedaría suelta, no se podrá realizar el refactoring. En caso contrario se muestra el modelo navegacional sin la página seleccionada.

#### **5.2.2.14 Refactorings referentes a los paths**

---

Al seleccionar estas opciones, los refactorings son automáticos, no se interacciona con el usuario, así que una vez realizado el refactoring, se muestra al usuario el modelo navegacional cambiado.

## 5.2.3 Guardar modelo navegacional

Cuando se crea oportuno no aplicar más refactorings, se puede guardar el modelo navegacional en formato WebML.

## 5.2.4 Idiomas

El idioma por defecto de la aplicación es el inglés pero puede cambiarse al español. No obstante también es posible cambiarlo a otros idiomas si se hace a través de los ficheros de configuración.

En el directorio Language de la aplicación hay un archivo llamado default. En ese archivo está guardado el nombre del fichero que contiene los literales del idioma de la aplicación. Por ejemplo, si el idioma es english, en default estará grabada la palabra english y existirá un archivo english que contiene todos los literales de la aplicación.



Fig. 5.10 Ficheros de configuración de los idiomas

Los literales son la palabra en mayúscula antes del pipe | y lo que sigue es el texto que aparecerá en la aplicación. En la siguiente figura se pueden ver unos ejemplos.

```
english
1 TITLE|Generator of Navigation Models
2 FILE|File
3 OPEN|Open
4 SAVEAS|Save As
5 CLOSE|Close
6 OPTIONS|Options
7 EXIT|Exit
8 EXIT_MESSAGE|Do you want to exit?
9 EXIT_MESSAGE_TITLE|Exit?
10 WELCOME|Welcome to Generator of Navigational Models
```

Fig. 5.11 Estructura de los literales del fichero de idiomas

Si se deseara poner por ejemplo el catalán, bastaría con poner un fichero llamado catalán, editarlo y añadir todos los literales. Por último sería necesario en el archivo default sustituir el nombre de fichero que figurara por catalan.



## 5.2.4 Ayuda

---

En caso que el usuario tenga alguna duda sobre el uso de la aplicación, podrá encontrar ayuda

### 5.2.4.1 Contenidos

---

Al seleccionar esta opción se abrirá una ventana con la ayuda en formato html para que el usuario pueda consultar todas las opciones de este manual de ayuda de manera detallada.

### 5.2.4.2 Acerca de

---

Aquí se puede consultar la información referente a la aplicación.

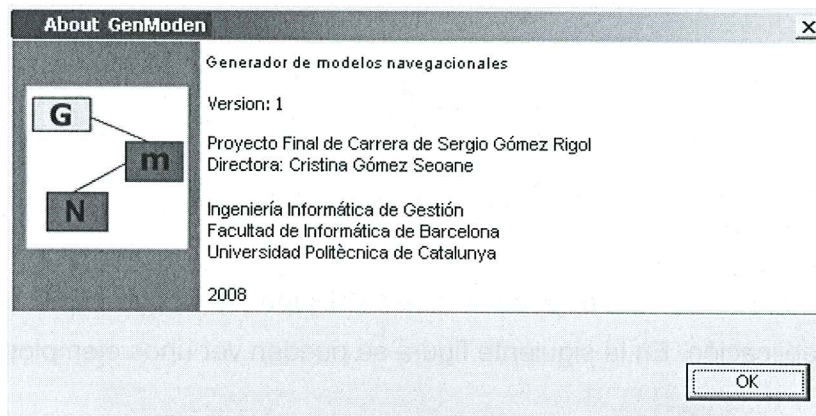


Fig. 5.12 Información sobre la aplicación

## 6. Planificación y análisis económico

En este apartado se puede ver una planificación estimada del tiempo requerido para la consecución del proyecto que elabora la aplicación de generación de modelos navegacionales.

El proyecto comienza con el estudio de una serie de herramientas con las que poder integrar la aplicación como son WebRatio y "Generación automática de código con WebRatio".

Teniendo en cuenta este punto se estudia el documento en el que se basa el proyecto que es "On the Quality of Navigation Models with Content-Modification Operations" y el documento "Refactorings for Navigation Models" que es el documento que engloba los diferentes tipos de refactorización.

Se muestran dos diagramas, el primero muestra la planificación prevista para la consecución del proyecto y el segundo es el tiempo real que se ha necesitado.

La planificación, una vez los estudios previos, continua con la especificación. Es importante fijarse que hay un salto temporal debido a que durante el verano no se ha realizado el proyecto. Se ha empleado el último mes primaveral para el estudio inicial.

La especificación engloba los casos de uso, el modelo conceptual y el modelo de comportamiento.

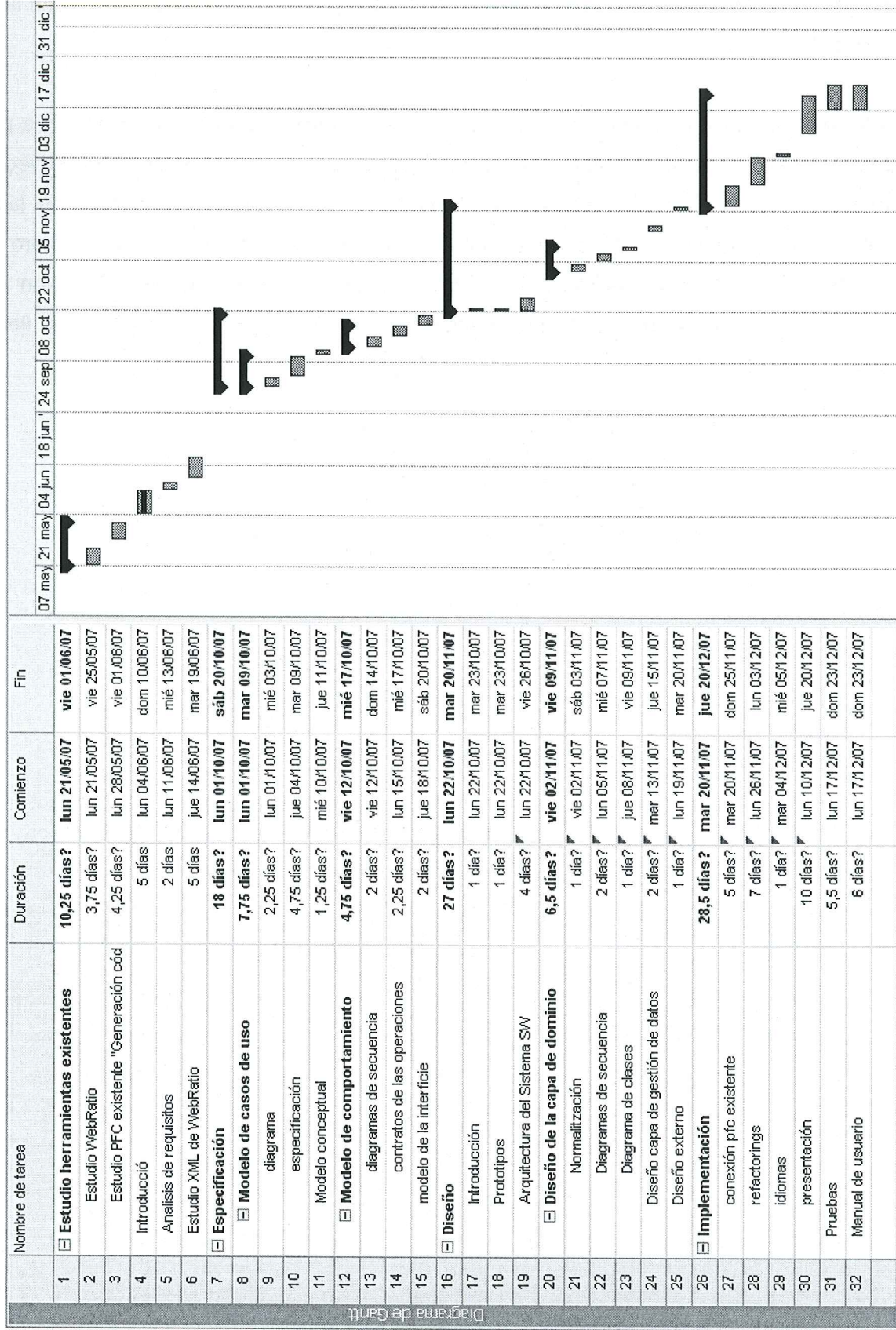
El siguiente paso sería el diseño, donde se trabaja con las tres capas que forman la arquitectura elegida y una vez finalizado el proceso de documentación se pasa a la implementación de la aplicación.

La implementación engloba la conexión con la herramienta del proyecto final de carrera existente, la generación de los refactorings, el tratamiento de los idiomas y las interfaces gráficas.

Se finaliza con las pruebas y el manual de usuario.

Es importante remarcar que los trámites a realizar una vez finalizado el proyecto no se indican.

## 6.1 Planificación estimada





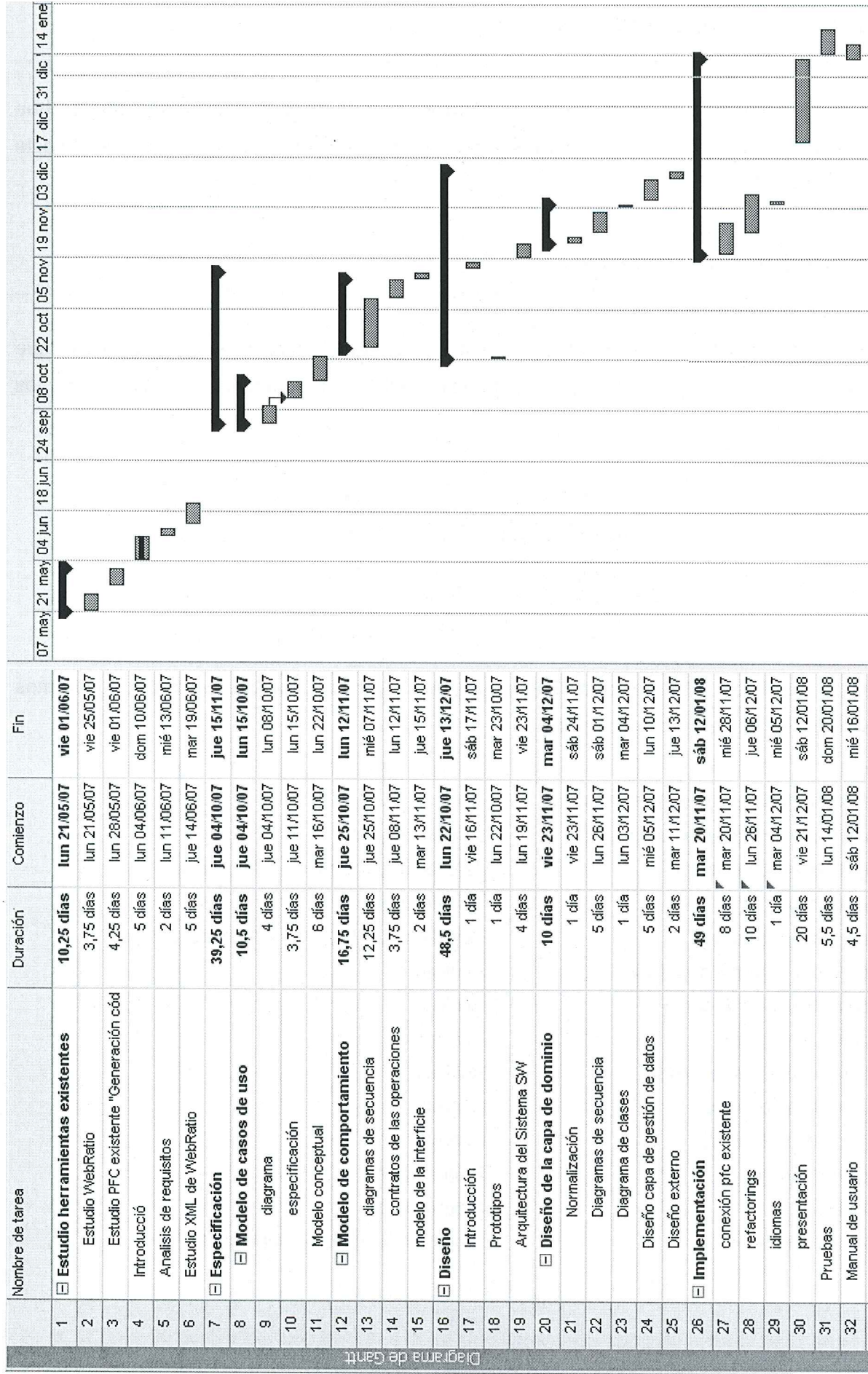
## 6.2 Planificación real

---

Durante toda la consecución del proyecto ha habido desviaciones debidas a correcciones de alguno de los puntos como los diagramas y durante el proceso de documentación en la fase de diseño.

En lo relacionado a la implementación los principales problemas han sido causados por el tiempo previsto en la recolección de información de la herramienta existente del proyecto de Generación automática de código con WebRatio. Pero el principal problema ha sido la creación gráfica del modelo navegacional que ha requerido mucho tiempo. Primero para encontrar la librería más óptima para la creación de los gráficos y a continuación en la manera de mostrar la entrada de información de los datos necesarios para poder llevar a cabo los refactorings.





## 6.3 Análisis económico

Este apartado está dedicado a estimar el coste económico del proyecto. Los costes se han desglosado en dos grupos: los referentes al personal involucrado y los referentes al coste del material utilizado.

### 6.3.1 Recursos humanos

Tal como se puede prever, los sueldos de los empleados variarán en función del rol que desarrollen. A continuación se muestra el sueldo (€/h.) correspondiente a cada uno de los roles:

Rol	€/ hora
Jefe de proyecto	60
Analista	45
Programador	30

Necesitamos saber las horas que trabaja cada uno y sabiendo su sueldo por hora podremos saber el coste de cada uno de los roles.

Desglosando las tareas principales:

<b>Estudio de herramientas existentes</b>	Jefe de proyecto
<b>Estudio del pfc existente</b>	Jefe de proyecto (80%) – Analista (20%)
<b>Introducción</b>	Jefe de proyecto
<b>Análisis de requisitos</b>	Jefe de proyecto (70%) – Analista (30%)
<b>Estudio del XML de WebRatio</b>	Analista
<b>Modelo de casos de uso</b>	Analista
<b>Modelo conceptual</b>	Analista
<b>Modelo de comportamiento</b>	Analista
<b>Modelo de la interficie</b>	Analista
<b>Diseño de las 3 capas</b>	Analista
<b>Implementación</b>	Programador
<b>Pruebas</b>	Programador
<b>Manual de usuario</b>	Programador (90%) – Analista (10%)

En la planificación figura el tiempo en días de las tareas. Los días de jefe de proyecto y analista son de 4 horas y los del programador de 7.

Rol	Horas	Coste
Jefe de proyecto	54	60
Analista	381	45
Programador	409	30

De la tabla anterior se puede comprobar que el coste total de los recursos humanos es de **32655 €**

### 6.3.2 Recursos materiales

---

Para realizar este proyecto, a parte del personal que ha trabajado, se necesita materiales. A continuación se muestran los costes.

Recurso	Precio
Conexión a internet	39 € x 7 meses de proyecto
Impresión de documentación	35 € x 5 copias

No se ha requerido ningún material suplementario ya que se ha optado por no cobrar material como el equipo informático ni gastos de luz.

El total de los recursos materiales es de **448 €**

### 6.3.3 Coste total

---

El coste total sumando los costes de los recursos humanos y los materiales es de **33103 €**



## 7. Trabajo futuro

La aplicación está realizada en Java y sobre una interficie gráfica. Aunque la interficie intenta mostrar el modelo de la manera más amigable posible, una característica importante es la usabilidad. Posiblemente la usabilidad sería mejorable a la hora de interactuar con el modelo creado. Para el futuro sería conveniente tratar estos puntos haciendo posible interacciones directas con el modelo, por ejemplo para eliminar links o páginas o fusionar o clonar a través de eventos del teclado o ratón.

A parte de los aspectos referentes a la usabilidad, hay otros que tienen que ver más con la parte funcional de la herramienta.

Una posible ampliación importante de la herramienta sería dar la posibilidad al usuario de ampliar la oferta de refactorizaciones hay en la aplicación.

Para ello se podría dar la documentación necesaria común entre todos los refactorings y que debería tener la nueva refactorización y a través del polimorfismo poder sobrescribir la funcionalidad a ejecutar para modificar el modelo.

En nuestro caso a causa del tiempo no se ha podido llevar a cabo esta funcionalidad. A nivel de la capa de dominio hubiera sido factible pero implementar la funcionalidad en la capa de presentación no era trivial ya que cada refactorización tiene su propia interfaz. No obstante si se hiciera la ampliación de usabilidad expuesta al principio de este capítulo posiblemente se reduciría la complejidad.

El último aspecto destacable es el coste temporal de las funcionalidades de la aplicación. Se ha intentado tener en cuenta pero en algunas ocasiones no se ha podido llevar a cabo como se hubiera esperado. El ejemplo más claro lo encontramos cada vez que aplicamos una refactorización. En algunas refactorizaciones el caso peor llega a ser cuadrático y posiblemente se podría mejorar este coste.

Además cada vez que se realiza el refactoring, se vuelve a generar el grafo con la librería JGraph como la primera vez, pero es posible que se pudiera de alguna manera no generar todo el grafo cada vez, sino simplemente generar o modificar la parte que el refactoring trata, manteniendo las otras partes del grafo intactas y sin necesidad de rehacerlo entero.



## 8. Conclusiones

A continuación se expondrán las principales conclusiones derivadas de la realización de este proyecto. Primero hablaremos de la utilidad, beneficios y futuro de la aplicación y, seguidamente valoraremos de manera personal el hecho de haber realizado este proyecto.

### 8.1 De la aplicación

---

La aplicación realizada puede ser de gran ayuda a todos los diseñadores de software que trabajan con los 3 tipos de modelos. El modelo navegacional es muy utilizado para aplicaciones con una cierta complejidad y esta herramienta puede ser de gran ayuda tanto para ahorrar tiempo como para la detección de errores.

Como se explicaba en el principio del trabajo, no hay aplicaciones que hagan la generación automática del modelo navegacional a partir del de datos. Aunque nos ayudamos del proyecto final de carrera existente, aportamos una interfaz gráfica diferente que es más práctica para la aplicación de los refactorings.

Ofrecer la posibilidad de aplicar una serie de modificaciones al modelo navegacional creado aporta además un plus de funcionalidad a una aplicación ya de por sí práctica.

La aplicación además puede ser ampliada. Siguiendo este documento y teniendo en cuenta el principio de reusabilidad que se ha tenido siempre presente, no resultará muy complicado poder hacerse con la aplicación y poder ampliarla tanto como se quiera. La aplicación, como se ha explicado, se ha programado siguiendo las 3 capas (presentación, dominio y gestión de datos) de esta manera si se desea, se puede cambiar la capa de presentación, por ejemplo sin tener que tocar nada de la funcionalidad del programa.

Es difícil ser objetivo para valorar la aplicación y más cuando la ha realizado uno mismo, y la manera de ser lo más objetivo posible es teniendo en cuenta las pruebas realizadas y comprando el resultado final con lo que se esperaba. Teniendo en cuenta estas premisas, considero que la aplicación cumple con los requisitos expuestos desde un principio y cumple con la labor para la que se ha creado.

## 8.2 Valoración personal

---

El proyecto final de carrera como su nombre indica es la culminación de la carrera con un proyecto que reúna los conocimientos adquirido durante los años en los que se ha estado estudiando la Ingeniería.

En este proyecto, se engloba la especificación, diseño, desarrollo de una aplicación. Además se puede incluir tareas de planificación, organización... que son aspectos muy tratados durante la ingeniería de Gestión, que es mi caso. Pero hay otro trabajo que no queda reflejado pero que también se realiza y que forman parte de la madurez que se adquiere cada vez que se realiza un nuevo proyecto.

Cuando tuve que elegir proyecto, lo primero que hice fue fijarme en proyectos relacionados con temas de web y poder conseguir la oferta de este proyecto fue para mi muy importante puesto que era llevar a cabo una aplicación física de unos artículos de investigación sobre temas relacionados con modelos navegacionales, importantes en la realización del análisis.

Es importante la ayuda que he recibido para poder llevar a cabo esta aplicación, tanto de personal cualificado como no cualificado. Agradezco a la directora de este proyecto, Cristina Gómez, todos los consejos y el haberme guiado durante la elaboración de este proyecto.

La conclusión más importante es todo lo que he aprendido realizando este proyecto y que ya, a día de hoy, pongo en práctica en los proyectos web que realizado en el mundo laboral.

## 9. Glosario

**Hipertexto:** Sistema que permite que un texto contenga enlaces con otras secciones del documento o con otros documentos

**Link:** Apuntadores de Hipertexto que sirven para saltar de una información a otra, o de un servidor a otro, cuando se navega por internet.

**Página:** Documento situado en una red informática, al que se accede mediante enlaces de hipertexto

**Modelo de datos:** Descripción de forma abstracta de cómo se representan unos datos.

**Modelo Navegacional:** Un modelo de navegación (también conocido como modelo de hipertexto) especifica la organización de las interfaces de una aplicación web.

**Modelo de presentación:** Definición de la forma en la cual los objetos navegacionales pueden aparecer, cómo los objetos de interfaz activarán la navegación y el resto de la funcionalidad de la aplicación.

**Refactornig:** Modificaciones aplicadas sobre un objeto (modelo) sin modificar sus características.

**WebML:** Notación visual para el diseño de aplicaciones Web complejas que usan datos intensivamente

**XML:** Extended Markup Language, es un metalenguaje extensible de etiquetas. Permite definir la gramática de lenguajes específicos.

## 10. Bibliografía

### 10.1 Referencias

---

**Jordi Cabot, Jordi Ceballo, Cristina Gomez**

On the Quality of Navigation Models with Content-Modification Operations

**Jordi Cabot, Cristina Gomez**

Refactorings for Navigation Models

**Javier Martín Rubio**

Proyecto Final de Carrera: Generación automática de código con WebRatio

**Ariadna Bartra Cisa**

Proyecto Final de Carrera: Sistema per a la definició i validació de restriccions de cardinalitat d'associacions ternàries



## 10.2 Enlaces electrónicos

---

**WebRatio:**

<http://www.webratio.com>

**JDOM:**

<http://www.jdom.org>

**JGraph:**

<http://www.jgraph.com>

**API Java Standard Edition**

<http://java.sun.com/j2se/1.4.2/docs/api/>

**Visual data modeling in Rational Application Developer 6.0: new data model features:**

<http://www.ibm.com/developerworks/rational/library/05/r-3312/>

**Otros:**

[http://www.freedownloadmanager.org/es/downloads/JGraph\\_34374\\_p/](http://www.freedownloadmanager.org/es/downloads/JGraph_34374_p/)

<http://www.wikipedia.org>

