



# DESENVOLUPAMENT D'APLICACIÓ PEL RECONeixEMENT DE TARGETES DE PAGAMENT

MARCEL PEÑA HIDALGO

**Director/a**

MIGUEL ÀNGEL DOMÍNGUEZ TORRES (INTERNET SECURITY AUDITORS SL )

**Ponent:** SILVIA LLORENTE VIEJO (Departament d'Arquitectura de Computadors)

**Titulació**

Grau en Enginyeria Informàtica (Tecnologies de la informació)

**Memòria del treball de fi de grau**

**Facultat d'Informàtica de Barcelona (FIB)**

**Universitat Politècnica de Catalunya (UPC) - BarcelonaTech**

**28/06/2024**

## **Resum**

El projecte "Desenvolupament d'Aplicació per al Reconeixement de Targetes de Pagament" té com a objectiu dissenyar i crear una eina pròpia per a la identificació de dades de targetes de pagament, en resposta a les necessitats específiques de seguretat de l'empresa ISecAuditors. Actualment, s'utilitza una aplicació externa, CardRecon, que comporta costos elevats i una dependència tecnològica. La nova aplicació pretén millorar l'eficiència operativa, reduir costos i oferir un major control sobre els processos interns. Aquesta eina busca assegurar el compliment de l'estàndard PCI DSS, protegint les dades crítiques dels clients i prevenint el frau, augmentant així la competitivitat de l'empresa en el mercat de la seguretat de la informació.

## **Resumen**

El proyecto "Desarrollo de Aplicación para el Reconocimiento de Tarjetas de Pago" tiene como objetivo diseñar y crear una herramienta propia para la identificación de datos de tarjetas de pago, en respuesta a las necesidades específicas de seguridad de la empresa ISecAuditors. Actualmente, se utiliza una aplicación externa, CardRecon, que conlleva costes elevados y una dependencia tecnológica. La nueva aplicación pretende mejorar la eficiencia operativa, reducir costes y ofrecer un mayor control sobre los procesos internos. Esta herramienta busca asegurar el cumplimiento del estándar PCI DSS, protegiendo los datos sensibles de los clientes y previniendo el fraude, aumentando así la competitividad de la empresa en el mercado de seguridad de la información.

## **Abstract**

The "Development of Application for the Recognition of Payment Cards" project aims to design and create an own tool for the identification of payment card data, in response to the specific security needs of the ISecAuditors company. Currently, an external application, CardRecon, is used, which involves high costs and a technology dependency. The new application aims to improve operational efficiency, reduce costs and offer greater control over internal processes. This tool seeks to ensure compliance with the PCI DSS standard, protecting sensitive customer data and preventing fraud, thus increasing the company's competitiveness in the information security market.

# Índex

1. Introducció.....	7
1.1. Contextualització.....	7
1.2. Identificació del problema.....	8
1.3. Terminologia i definicions.....	9
1.4. Actors implicats.....	11
2. Justificació.....	12
2.1. Solucions i alternatives existents.....	12
2.2. Solució pressa.....	12
3. Abast.....	13
3.1. Objectiu.....	13
3.2. Competències tècniques.....	14
3.3. Possibles obstacles i riscos.....	14
4. Metodologia.....	16
5. Planificació temporal.....	18
5.1. Tasques de Gestió del Projecte.....	18
5.1.1. Gestió del Projecte.....	18
5.1.2. Recerca i Investigació.....	19
5.1.3. Disseny del Projecte.....	20
5.1.4. Desenvolupament del Projecte.....	20
5.2. Recursos.....	21
5.3. Estimació de les tasques.....	22
5.4. Diagrama de Gantt.....	23
5.5. Gestió dels Riscos.....	24
5.6. Actualització Planificació Temporal.....	25
6. Gestió econòmica del projecte.....	27
6.1. Pressupost.....	27
6.1.1. Personal.....	27
6.1.2. Recursos de Maquinari.....	29

6.1.3. Recursos de Programari.....	29
6.1.4. Despeses Generals.....	30
6.1.5. Cost de Contingència.....	31
6.1.6. Cost d'Imprevistos.....	31
6.1.7. Estimació Resum Final.....	32
6.2. Control de Gestió.....	32
6.3. Actualització Gestió Econòmica.....	33
7. Sostenibilitat del Projecte.....	35
7.1. Dimensió Econòmica.....	35
7.2. Dimensió Mediambiental.....	36
7.3. Dimensió Social.....	36
8. Desenvolupament Tècnic.....	38
8.1. Disseny.....	38
8.1.1. Pantalla Login.....	39
8.1.2. Pantalla principal.....	41
8.1.3. Pantalla Secundària.....	42
8.2. Desenvolupament inicial.....	44
8.2.1. Configuració de botons.....	44
8.2.2. Funcions bàsiques.....	49
8.3. Desenvolupament de funcions.....	52
8.4. Base de dades.....	56
9. Integració de coneixements.....	58
10. Identificació de lleis i regulacions.....	59
11. Conclusions.....	61
12. Referències.....	63
Annex.....	65

## Taula de Figures

Figura 1. Anvers i revers d'una targeta de crèdit amb els diferents elements.....	10
Figura 2. Diagrama de la Metodologia Iterativa.....	17
Figura 3. Diagrama de Gantt del Projecte.....	23
Figura 4. Diagrama de flux de l'aplicació.....	39
Figura 5. Pantalla log in.....	40
Figura 6. Pantalla principal.....	42
Figura 7. Pantalla secundària.....	44
Figura 8. Codi d'execució de l'aplicació.....	45
Figura 9. Modificació del codi d'execució de l'aplicació.....	45
Figura 10. Codi de la classe LoginDialog i la seva funció Login.....	46
Figura 11. Inicialització de les diferents pantalles.....	47
Figura 12. Carrega pantalla secundaria.....	47
Figura 13. Declaració de QPushButton i QCheckBox.....	47
Figura 14. Declaració funció on_pushButton_buscar_clicked().....	48
Figura 15. Pestanya emergent de QFileDialog.....	49
Figura 16. Codi de la funció action_open.....	50
Figura 17. Arxiu de configuració.....	50
Figura 18. Codi de la funció Configuracion.....	51
Figura 19. Codi de la funció select_export.....	52
Figura 20. Inici del codi de la funció «buscar_coincidencias».....	53
Figura 21. Codi de la funció buscar_directorios.....	54
Figura 22. Codi ampliat de la funció action_open.....	54
Figura 23. Codi de les funcions validar_numero_tarjeta i validar_algoritmo_luhn.....	56
Figura 24. Funció complerta action_about.....	66
Figura 25. Codi de la funció action_save_as.....	67
Figura 26. Codi de la funció mostrar_error.....	68
Figura 27. Codi de les dos primeres condicions de la funció «buscar_coincidencias»	69
Figura 28. Codi de les dos últimes condicions de la funció «buscar_coincidencias»...	70
Figura 29. Codi de la funció mostrar_tarjetas.....	71
Figura 30. Codi de la funció detectar_encoding.....	72
Figura 31. Part inicial del codi de la funció «buscar_grupos_numeros».....	73
Figura 32. Segona part de la funció buscar_grupos_numeros.....	74

Figura 33. Codi de la funció buscar_tarjetas.....	75
Figura 34. Codi de la funció crea_tablas.....	76
Figura 35. Codi de la funció action_export.....	77
Figura 36. Codi de la funció export_to_pdf.....	78
Figura 37. Codi de la funció export_to_excel.....	78
Figura 38. Codi de la funció action_back.....	79
Figura 39. Codi de la funció is_junction.....	79
Figura 40. Codi del script trans.sh.....	80

## Taula de Taules

Taula 1. Taula resum de Tasques.....	22
Taula 2. Taula sobre els Riscos i la seva mitigació.....	25
Taula 3. Modificació de la Taula resum de Tasques.....	26
Taula 4. Taula de Salari per Rol desenvolupat al Projecte.....	27
Taula 5. Taula dels Costos Humans de l'Estimació de les Tasques.....	28
Taula 6. Taula de Costos i Amortització del Maquinari.....	29
Taula 7. Taula de Costos de Programari.....	29
Taula 8. Taula de Costos i Amortització d'elements generals.....	30
Taula 9. Taula de predicció de Costos d'Imprevistos.....	31
Taula 10. Taula de Resum dels Costos.....	32
Taula 11. Modificació de la Taula dels Costos Humans de l'Estimació de les Tasques.....	34
Taula 12. Taula de Resum dels Costos.....	34

# 1. Introducció

Amb el creixement continu de les transaccions electròniques i el comerç en línia, la seguretat de les targetes de pagament és una preocupació prioritària per als consumidors, les empreses i els proveïdors de serveis financers. Les empreses del sector s'enfronten a reptes constants com ara; protegir-se del furt de les dades dels clients, prevenir el frau en les transaccions amb targetes de pagament, a més d'haver de complir amb una creixent normativa.

En resposta a aquests reptes, hi han empreses especialitzades en la protecció de dades bancàries que ofereixen serveis com a solucions de prevenció de fraus, xifrat de dades, detecció d'intrusions, monitoratge continu de la seguretat i altres serveis relacionats amb la ciberseguretat. Hi han empreses que inclouen proveïdors de serveis de seguretat informàtica, així com empreses especialitzades en serveis de pagament i altres proveïdors de serveis de processament de pagaments que ofereixen opcions de seguretat addicionals als seus clients empresarials.

Al marc de l'empresa on estic fent les pràctiques universitàries vaig copsar que hi havia diverses problemàtiques relacionades amb la seguretat de les dades bancàries i és el que em va motivar a oferir la possibilitat d'una col·laboració amb Internet Security Auditors, SL, d'ara endavant ISecAuditors. Fruit de l'anàlisi de les necessitats de l'empresa va sorgir la idea treballar en una solució que ajudés a evolucionar els seus procediments interns i ajudar altres empreses. L'objectiu és agilitzar la identificació de dades de targetes de pagament, millorar l'eficiència operativa de IsecAuditors i el desig de contribuir a incrementar el valor afegit als serveis que ofereixen als seus clients amb la finalitat de poder redefinir estratègies defensives existents o fins i tot, crear-ne de noves per tal de protegir les dades crítiques al mateix temps que prevenir el frau.

## 1.1. Contextualització

El treball de fi de grau «DESENVOLUPAMENT D'APLICACIÓ PEL RECONeixEMENT DE TARGETES DE PAGAMENT», forma part dels estudis del Grau d'Enginyeria Informàtica dintre de l'especialitat de Tecnologies de la Informació. Aquest grau és impartit per la Facultat d'Informàtica de Barcelona (FIB), a la Universitat Politècnica de Catalunya (UPC). El projecte es realitza dins del marc de l'empresa i es treballa sota les guies del director Miguel Ángel Domínguez Torres, expert en consultoria i auditoria de tecnologies de la informació.

L'empresa ISecAuditors[1] neix a mitjans de l'any 2001 amb l'objectiu de ser el proveïdor de seguretat independent que les empreses poden necessitar, donant servei global des de les seves oficines de Barcelona, Madrid i Bogotà. Compten amb experts en seguretat amb àmplia experiència i ofereixen un ampli ventall de serveis de Consultoria, Auditoria, Ciberintel·ligència, Seguretat Gestionada i Formació en Seguretat en Sistemes d'Informació que ajuden a protegir els actius de les organitzacions que confien en aquesta empresa, entre

les quals es troben les majors empreses espanyoles en els seus respectius sectors i clients en més de 15 països.

D'una banda, ofereixen serveis especialitzats de consultoria i auditoria en Ciberseguretat i, en segon lloc, disposen d'un equip especialitzat en productes i solucions de fabricants per a aquells projectes on es plantegin aquestes necessitats. El resultat que s'obté és mantenir la independència sobre fabricants, coneixent els seus productes, i comptant amb els millors experts especialistes en cadascuna de les solucions disponibles al mercat. Així mateix, la xarxa de col·laboradors és internacional, disposant d'empreses que col·laboren activament amb l'equip de professionals en projectes tant a Europa com a Amèrica Llatina.

Durant el novembre de 2022, com a part del meu programa de formació dual, vaig unir-me a IsecAuditors com a consultor de seguretat informàtica. Aquesta oportunitat va sorgir gràcies a una oferta de feina al marc del programa de "pràctiques per empresa" facilitat per la Facultat d'Informàtica de Barcelona. Les meves responsabilitats i tasques durant aquest període se centren principalment en l'àmbit de la consultoria especialitzada en l'estàndard PCI DSS. En aquest rol, he estat immers en projectes i activitats destinats a garantir el compliment i la implementació efectiva d'aquest estàndard de seguretat de la indústria de les targetes de pagament. Aquestes pràctiques m'han permès adquirir coneixements pràctics i habilitats rellevants en la gestió de la seguretat de la informació i la protecció de les dades delicades dels clients.

Després d'explicar als responsables de l'empresa els diferents projectes elaborats a la carrera i una vegada avaluats el meu rendiment i aptituds, se'm va plantejar l'opció de realitzar un projecte de millora, el disseny i desenvolupament de l'aplicació pel reconeixement de targetes de pagament, aprofitant l'oportunitat de plantejar-ho com un treball de fi de grau.

## **1.2. Identificació del problema**

Les dades de les targetes de pagament són susceptibles a múltiples amenaces de seguretat, que van des de l'emmagatzematge inadequat fins a un ús imprudent. En resposta a aquesta realitat, l'estàndard PCI DSS ofereix un marc de requisits meticulosament dissenyat per assegurar que les empreses que manipulen, emmagatzemen o transmeten aquestes dades ho facin de manera segura i responsable. El seu propòsit principal és garantir que tota la informació associada amb les Dades de Titular de Targeta (CHD) que es desi, es faci utilitzant un xifratge segur.

Una de les principals responsabilitats per assegurar la correcta gestió d'aquestes dades i, per tant, garantir la seguretat del Medi d'Entorn del Client (CDE) sense vulnerabilitats, és la revisió dels arxius generats en transaccions i emmagatzemats dins dels diferents sistemes de l'empresa.

En l'actualitat, aquesta tasca es duu a terme mitjançant l'ús d'una aplicació de mercat coneguda com a CardRecon, la qual facilita l'anàlisi dels sistemes operatius per detectar dades de targeta en text clar. Això implica els següents problemes:

- El fet que sigui una aplicació externa genera una dependència tecnològica que limita el control sobre els processos que realitza deixant poca flexibilitat pels entorns específics i els seus requisits particulars.
- D'altra banda, l'aplicació és gestionada mitjançant subscripcions per usuaris, això suposa uns costos associats molt elevats per l'empresa que repercuteix sobre els clients.
- L'ús de l'aplicació externa suposa una reducció pel que fa a l'eficiència en el desenvolupament de tasques pel fet que es requereixen gestions que tenen un impacte directe en fluxos de treball específics de l'empresa.

### 1.3. Terminologia i definicions

- **PCI DSS:** És l'estàndard de seguretat de dades per l'indústria de targetes de pagament[2]. Aquest estàndard defineix un conjunt de mesures de seguretat dissenyades per protegir les dades de titulars de targeta de pagament durant el processament, emmagatzematge i transmissió de les dades.
- **CardHolder Data:** Fa referència a tota la informació que pugui ser utilitzada per identificar un titular de targeta o la seva targeta de pagament. S'acostuma a utilitzar el seu acrònim «CHD» i està compost per 4 elements principals[3]:
  - **PAN:** És l'acrònim de «Primary Account Number», que fa referència al número que trobem a l'anvers o revers de les targetes de crèdit. Aquest número és únic i identifica cada targeta. En funció de la companyia que proporcioni la targeta aquest número serà d'entre 12 i 19 dígits.
  - **Titular de targeta:** Nom complet que està imprès, juntament amb el PAN i la Data de venciment, que indica el nom del propietari de la targeta.
  - **Data de venciment:** És el conjunt de números que s'acostuma a trobar a continuació del PAN. Aquests números, de dos dígits separats per una barra, indiquen el mes i l'any de caducitat de la targeta.
- **Codi de servei:** És un codi numèric de 4 dígits aleatoris que autoritzen el pagament en una transacció. Aquest codi s'utilitza per verificar la identitat del pagador amb la targeta pel fet que és un codi únic i que el titular acostuma a personalitzar per recordar-lo.

- **Sensitive Authentication Data:** Fa referència a tots els elements de la targeta que contenen la informació completa del compte. S'acostuma a utilitzar el seu acrònim «SAD» i està compost per 3 elements:
  - **Codi d'autorització:** Més conegut com a «CVV2 o Card Verification Value 2», és un número de 3 o 4 dígits en funció de la targeta. Aquest és un codi numèric que permet autoritzar compres que es fan de manera no presencial. Juntament amb el nom CVV2, també rep els noms de CAV2, CVC2 o CID.
  - **Bloc PIN:** És un element situat en l'anvers de la targeta que està format per un petit xip. Aquest xip conté tota la informació necessària per poder realitzar una transacció. Es fa ús mitjançant la introducció de la targeta, per la banda on és el xip dins d'un terminal de pagament habilitat.
  - **Cinta Magnètica:** És un element situat al revers de la targeta que està format per una tira de color negre. Aquesta tira compta tota la informació del compte, com el titular de targeta, PAN, data de venciment i codi de servei. S'empra mitjançant el lliscament de la banda magnètica per un lector magnètic que recull la informació proporcionada per la cinta magnètica.

A continuació, a la figura 1, mostrem una imatge amb les localitzacions dels termes definits anteriorment.

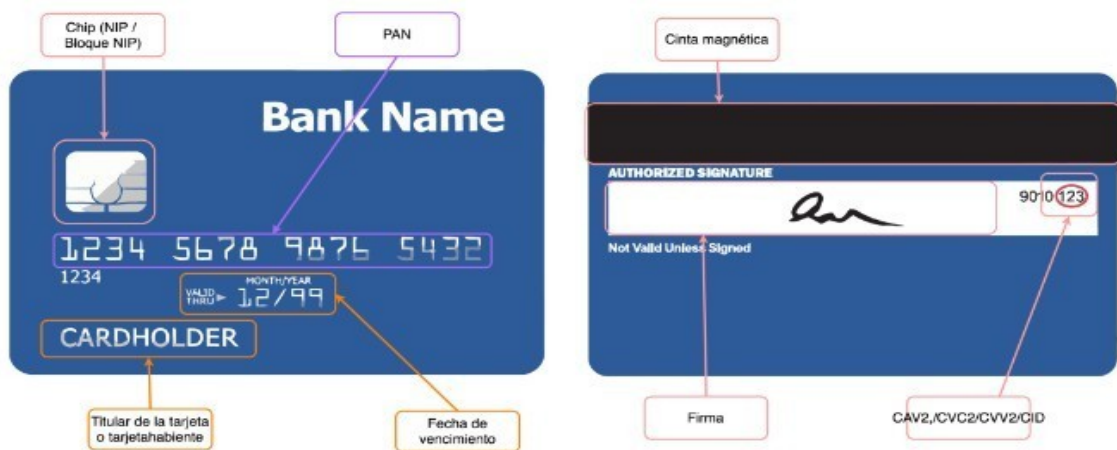


Figura 1. Anvers i revers d'una targeta de crèdit amb els diferents elements

Font. [www.pciresources.com](http://www.pciresources.com)[3]

- **CardHolder Data Enviroment:** Fa referència a l'entorn de dades de titulars de targetes i s'acostuma a fer servir les seves sigles «CDE». El CDE inclou qualsevol sistema o xarxa que emmagatzemi, processi o transmeti dades de targetes de pagament.

## 1.4. Actors implicats

A continuació es llisten els diferents actors implicats en el desenvolupament d'aquest projecte. La llista inclou les persones implicades directament o indirectament en el projecte sigui per interès, benefici, o participació en el desenvolupament.

- **Desenvolupador:** Aquest projecte només té un desenvolupador, en Marcel Peña Hidalgo. Ell serà l'encarregat d'investigar possibles tecnologies que es poden utilitzar, desenvolupar l'aplicació i components que es puguin utilitzar per fer proves o desplegar l'aplicació i tota la documentació necessària per al projecte.
- **Director del Treball de Fi de Grau:** El director Miguel Angel Dominguez Torres de l'empresa IsecAuditors és el responsable de la branca de consultoria. És l'encarregat de supervisar el meu projecte.
- **IsecAuditors:** Empresa per la qual es desenvoluparà el projecte i el principal beneficiari del projecte.
- **Organitzacions:** La publicació de manera gratuïta de l'aplicació, codi obert, permetria a qualsevol empresa en el món de la ciberseguretat que en sigui beneficiari en el moment que en faci ús. Sempre podrà adaptar l'aplicació a les seves necessitats i millorar el desenvolupament de certes tasques en funció de l'evolució de l'aplicació.
- **Població general:** Aquest és un actor indirecte. El desenvolupament i millora de l'aplicació podrien tenir un impacte significatiu en la protecció de les dades de les targetes, i comportaria una millora en la seguretat i, per tant, un benefici per a tota població en general.

## 2. Justificació

### 2.1. Solucions i alternatives existents

Actualment, existeix al mercat una solució coneguda com a CardRecon[4], que serveix com a punt de referència per al nostre projecte. Aquesta eina de programari ha estat desenvolupada per la companyia nord-americana Ground Labs i està dissenyada per a la detecció i anàlisi de dades de targeta de pagament en conformitat amb l'estàndard PCI DSS, en diversos sistemes. Les funcionalitats clau de l'aplicació inclouen la cerca i anàlisi de sistemes, la identificació de dades crítiques, així com la generació d'informes i alertes de seguretat. Per a utilitzar-la, es requereix l'adquisició d'una llicència, amb opcions disponibles com l'estàndard, professional o empresarial, que varien segons les funcionalitats requerides i el nombre d'usuaris necessaris.

### 2.2. Solució pressa

La justificació de la solució presa comença destacant la manca de competència en el mercat per a aquest tipus d'eines comercials que reconeguin PANs, o sigui, els números de compte de les targetes de pagament. Això suggeriria que, malgrat tenir una solució existent, aquesta no cobreix completament les necessitats del projecte, ja que no hi ha alternatives viables que ofereixin una millor relació qualitat-preu o funcionalitats addicionals. A més, es remarca que els elevats costos associats amb l'adquisició d'una llicència per a l'ús de CardRecon fan que l'opció de desenvolupar una solució pròpia sigui més atractiva econòmicament. Aquesta decisió s'emmarca en una visió a llarg termini, on es considera que la inversió inicial en el desenvolupament d'una solució interna pot resultar en un estalvi substancial al llarg del temps. A més, el control intern sobre el desenvolupament i la millora de l'aplicació permet una major flexibilitat i adaptabilitat a les necessitats específiques del projecte i dels usuaris finals. Finalment, es destaca que l'objectiu principal és assegurar que l'aplicació resultant compleixi els estàndards de seguretat requerits i que proporcioni resultats precisos i complets en la detecció i anàlisi de dades de targeta de pagament.

En definitiva, en ser capaços de respondre de manera més ràpida i precisa a les necessitats dels clients, l'empresa pot augmentar la seva competitivitat en el mercat i captar més clients potencials i pot representar una via directa cap a una millora substancial del rendiment econòmic global de l'empresa.

## 3. Abast

### 3.1. Objectiu

L'objectiu general del projecte és dissenyar i desenvolupar una aplicació interna que faciliti la revisió i protecció de sistemes i dades de targetes de pagament, amb funcionalitats similars a les de l'eina CardRecon. La seva finalitat és augmentar el control i la flexibilitat en la gestió de les dades de targetes de pagament mitjançant el desenvolupament d'una solució interna, permetent adaptar-se millor als requisits específics dels sistemes de l'empresa i evitar la dependència tecnològica d'aplicacions externes com CardRecon. Reduir els costos associats amb l'adquisició de llicències per a aplicacions externes com CardRecon, optant per una solució interna que proporcioni una millor relació qualitat-preu i sigui més econòmica a llarg termini. Millorar l'eficiència en el desenvolupament de tasques relacionades amb la gestió de dades de targetes de pagament, reduint els temps i recursos invertits en processos que impacten directament en els fluxos de treball específics de l'empresa, sent així més competitiu en el mercat.

Descripció dels subobjectius tècnics en el desenvolupament de l'aplicació:

#### 1. Desenvolupament d'algorismes de detecció de PANs:

- Implementar algorismes de processament de llenguatge natural per a identificar i validar números de compte de targeta de pagament (PANs) en documents i arxius de text.
- Desenvolupar tècniques de reconeixement de patrons per a detectar PANs amb diferents formats i contextos.

#### 2. Arquitectura distribuïda i independència de sistemes:

- Dissenyar una arquitectura modular que permeti la integració fàcil amb múltiples sistemes sense generar dependències tecnològiques.
- Assegurar que l'aplicació sigui compatible amb diferents sistemes operatius i entorns de desplegament, com ara Windows i Linux.

#### 3. Interfície d'usuari intuïtiva:

- Dissenyar una interfície d'usuari responsive i fàcil d'utilitzar, amb funcionalitats de cerca avançada, filtres interactius i notificacions en temps real.
- Proporcionar una experiència de navegació coherent i consistent.

#### 4. Seguretat i privadesa de dades:

- Implementar mesures de seguretat avançades per a protegir les dades de targetes de pagament, incloent-hi xifratge de dades en repositori i en trànsit.
- Garantir el compliment de les normatives de protecció de dades, com ara el estàndard PCI DSS, per a minimitzar el risc de filtracions de dades.

#### 5. Documentació i Suport Tècnic:

- Generar documentació detallada sobre l'ús, la configuració i el manteniment de l'aplicació per a facilitar la seva implementació i gestió en múltiples sistemes.
- Proporcionar suport tècnic adequat per a resoldre els problemes i les consultes dels usuaris en diferents entorns d'ús.

### Altres requisits

Pel que fa a altres requisits també necessaris seria desenvolupar i programar les funcionalitats de l'aplicació de manera eficient i eficaç, utilitzant les millors pràctiques de desenvolupament de programari. Al mateix temps també garantint una disponibilitat del servei elevada mitjançant l'ús de redundància de servidors, plans de contingència i supervisió proactiva per a identificar i mitigar problemes de forma anticipada.

## 3.2. Competències tècniques

- **CTI1.1:** Demostrar comprensió de l'entorn d'una organització i de les seves necessitats en l'àmbit de les tecnologies de la informació i les comunicacions.
- **CTI1.3:** Seleccionar, desplegar, integrar i gestionar sistemes d'informació que satisfacin les necessitats de l'organització amb els criteris de cost i qualitat identificats.
- **CTI2.2:** Administrar i mantenir aplicacions, sistemes informàtics i xarxes de computadors (els nivells de coneixement i de comprensió són a les competències tècniques comunes).
- **CTI2.3:** Demostrar comprensió, aplicar i gestionar la garantia i la seguretat dels sistemes informàtics (CEIC6).
- **CTI4:** Emprar metodologies centrades en l'usuari i l'organització per al desenvolupament, l'avaluació i la gestió d'aplicacions i sistemes basats en tecnologies de la informació que assegurin l'accessibilitat, l'ergonomia i la usabilitat dels sistemes.

## 3.3. Possibles obstacles i riscos

Durant el procés de creació del projecte poden aparèixer certs riscos i problemes que poden alentir el desenvolupament del projecte. És per això, que saber-los identificar i saber-ne les conseqüències pot ajudar a reduir els seus efectes.

Els principals riscos identificats són:

- **Panificació errònia:** Aquest projecte, en ser un projecte de final de grau, té una data d'entrega fixada i tenint en compte que la quantitat de càrrega de feina que implica un projecte d'aquestes característiques, s'ha de realitzar una planificació molt estricta. Qualsevol imprevist o contratemps que pugui tenir pot influir molt negativament en el desenvolupament del projecte pel fet que la data d'entrega que no es pot modificar.

- **Desconeixement de les tecnologies:** Tot i utilitzar tecnologies ja conegudes, durant el desenvolupament del projecte s'incorporen funcionalitats que no s'havien utilitzat fins al moment. Això requereix dedicar temps a aprendre i comprendre-les per implementar-les de manera òptima. La manca d'experiència o coneixements específics en aquestes àrees podrien provocar retards en el progrés del projecte.
- **Errors (Bugs):** Com a tot projecte que comença des de zero, el desenvolupament de funcions i classes té implícits possibles errors de codi que impedeixen avançar en el projecte. Aquests errors es divideixen en dos segons tipus la tipologia. Per una banda, els errors en el codi com a tal que no suposen una gran dedicació de temps sinó més aviat una tasca de revisió i troballa d'aquest. D'altra banda, els errors d'implementació del codi, llibreries no compatibles o resultats que no produeixen el resultat esperat, poden requerir una reestructuració del codi o de l'estratègia, augmentant així el temps dedicat a buscar una solució alternativa. Tot i això, la seva ocurrència és menor. La resolució d'aquests errors, segons la seva criticitat, pot demanar un major o menor temps d'estudi i dedicació.
- **Desplegament de l'Arquitectura Distribuïda:** Dissenyar i implementar una arquitectura modular que permeti la integració amb múltiples sistemes pot ser desafiant. S'han de considerar les compatibilitats i les interfícies amb altres sistemes existents per garantir una integració fluida i sense conflictes.
- **Problemes de comunicació:** La comunicació deficient entre els membres de l'equip de desenvolupament, els stakeholders i els usuaris finals pot provocar una mala definició de requisits, malentesos, endarreriments i errors en el desenvolupament. El que resulta en un producte final que no compleix amb les expectatives.
- **Acceptació de l'Usuari Final:** És crucial que l'aplicació sigui ben rebuda per l'usuari final. Si l'aplicació no compleix amb les expectatives o no és fàcil d'usar, pot comportar la no utilització vers a recursos ja existents en el mercat. Els empleats poden mostrar resistència al canvi, especialment si estan acostumats a utilitzar eines externes com CardRecon. Caldrà un esforç significatiu per a convèncer-los dels beneficis de la nova solució interna.

La insuficient identificació, avaluació i gestió dels riscos pot causar sorpreses durant el desenvolupament, amb repercussions negatives en el projecte, reforçant la necessitat de reconèixer-los des del principi i implementar estratègies per mitigar-los eficaçment al llarg del cicle de vida del projecte.

## 4. Metodologia

La metodologia escollida per la realització del projecte és el mètode iteratiu. Aquest mètode es basa en la divisió per tasques o seccions del projecte complet que s'aniran desenvolupant de manera independent i integrant-se al projecte un cop finalitzades.

La metodologia iterativa es defineix en 5 etapes[5]:

- 1. Planificació i Requisites:** Durant aquesta etapa, es defineix el pla pel projecte que ha d'estar alineat amb els objectius generals del projecte. També s'especifiquen els requisits mínims perquè el projecte tingui èxit.
- 2. Anàlisi i Disseny:** Un cop definits els objectius, es procedeix a cobrir els requisits tècnics mitjançant la definició dels primers dissenys per poder complir els objectius. Per aquesta tasca, s'ha implementat el programari QT Designer, que permet el desenvolupament d'interfícies gràfiques de manera fàcil i visual.
- 3. Implementació:** És en aquest tercer pas, es creen les primeres iteracions del projecte. Aquesta iteració estarà basada en l'anàlisi i el disseny, del pas dos, amb la intenció que funcioni i compleixi els objectius proposats. Per aquesta tasca s'ha decidit utilitzar el llenguatge de programació Python pel fet que proporciona una gran versatilitat i nombrosos recursos com llibreries o frameworks que faciliten la implementació de l'aplicació.
- 4. Proves:** Un cop la primera iteració està creada correctament, s'ha de provar mitjançant diferents jocs de proves buscant que funcioni de la millor manera possible.
- 5. Avaluació:** Un cop finalitzats els jocs de proves, es valorà l'èxit de la iteració i et permet centrar-te en tot el que s'ha de canviar. En cas que fos necessari modificar coses, es pot generar una nova iteració realitzant de nou tots les passes mencionades anteriorment per intentar assolir nous objectius intentant evitar els errors ja comesos.

En el nostre cas, es realitza un disseny inicial del projecte on es planifiquen i defineixen els diferents requisits que ha de tenir l'aplicació. Al mateix temps es planifiquen i defineixen subrequisits sobre els quals anirem duent a terme les diferents iteracions fins a enllestir l'aplicació i generar una iteració completa de tota l'aplicació. En la part d'anàlisi i disseny, cada subrequisit té unes necessitats de disseny que s'hauran d'anar complint i posteriorment, en la fase tres, implementant per cobrir les necessitats establertes en la planificació i disseny. Posteriorment, es duran a terme les proves de cada component dissenyat per comprovar la seva correcta funcionalitat i finalment serà avaluat per l'usuari final per rebre retroalimentació i corregir els diferents errors o inconvenients que puguin aparèixer. Un cop finalitzats tots els subrequisits, es durà a terme una nova iteració amb el conjunt de l'aplicació per comprovar la correcta cohesió de tots els components i, per tant, el correcte funcionament. De nou, aquest conjunt de proves serà previ a una avaluació final dels usuaris per poder corregir qualsevol incongruència que aparegui.

En la figura 2 es mostra les cinc etapes de la Metodologia Iterativa

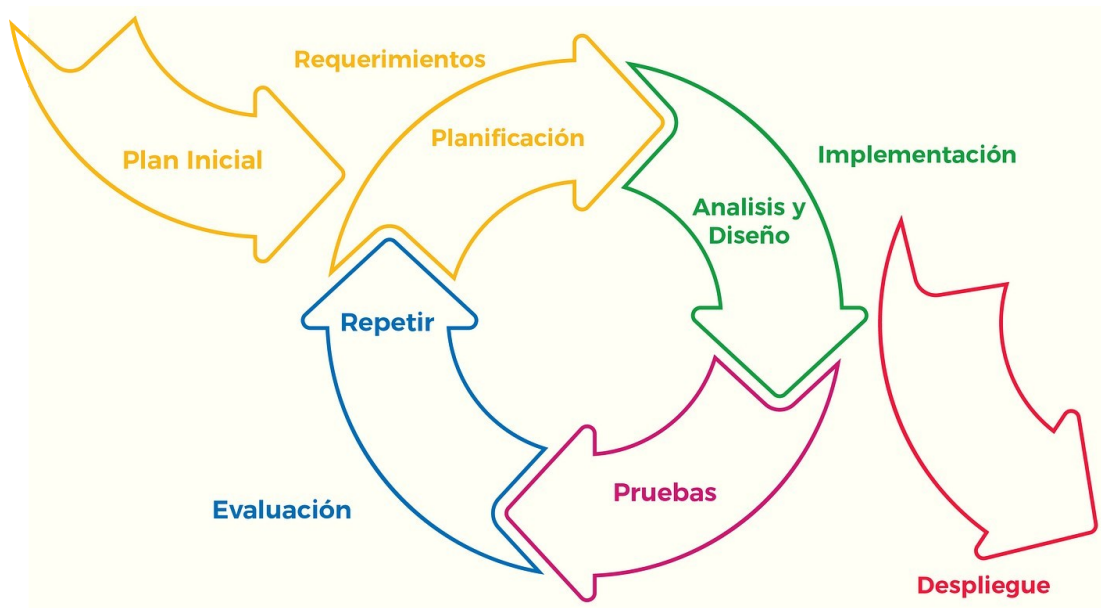


Figura 2. Diagrama de la Metodologia Iterativa

Font. medium.com[6]

Tenint això en compte, es duran a terme reunions setmanals amb el director del projecte i diferents membres de l'empresa per anar avaluant el progrés del projecte, poder realitzar canvis per tal de complir les necessitats específiques i resoldre problemes en cas de trobar qualsevol obstacle.

D'altra banda, s'utilitzaran un conjunt d'eines de gestió de projectes per poder planificar i estructurar les tasques del projecte, facilitant el seguiment per part del responsable del projecte. Les eines emparades són:

- **Celoxis[7]:** És una eina web especialitzada en la gestió de projectes. Permet la creació de projectes amb les diferents tasques a dur a terme, especificar les hores dedicades a cada tasca i afegir comentaris. Es farà servir per a indicar el seguiment del projecte i la dedicació d'hores a cada tasca.
- **Microsoft Teams[8]:** És una de les diferents aplicacions que disposa gratuïtament Microsoft, pensat per poder crear i mantenir conversacions mitjançant missatgeria instantània o celebrar reunions i videoconferències. Durant el projecte es farà servir per dur a terme les diferents reunions de seguiment amb el director del treball.

## 5. Planificació temporal

En aquest apartat, es descriu la planificació temporal de la càrrega de feina associada a l'elaboració del projecte. La finalitat d'aquesta planificació temporal és proporcionar una guia per al seguiment temporal del projecte i assegurar-ne la finalització dins del termini establert.

La data d'inici del projecte és el 14 de febrer de 2024 i s'allargarà fins al 28 de juny, quan es realitzarà la lectura i defensa d'aquest. Es preveu una dedicació diària d'unes 5 hores diàries, amb un total estimat de 25 hores setmanals. No obstant això, aquesta dedicació, pot variar segons els compromisos personals que puguin sorgir durant la setmana, de manera que el càlcul proporcionat és merament indicatiu. En total, es preveu una durada aproximada de 17 setmanes i un total d'unes 435 hores de feina.

Cada tasca del projecte està agrupada en un bloc principal i subdividit en petits blocs segons el tipus de tasca a dur a terme i el seu propòsit dins del projecte. Aquestes tasques s'identifiquen amb els noms de Gestió del Projecte (GP), Recerca i Investigació (R), Disseny de Projecte (D) i Desenvolupament del Projecte (DP).

### 5.1. Tasques de Gestió del Projecte

A continuació, es detallen les tasques que es realitzaran durant el projecte. Es presenten de forma ordenada, juntament amb una breu descripció i una estimació en nombre d'hores que es volen dedicar a cada tasca, en elles es contemplen les hores dedicades a la tasca i alguna hora extra per resoldre possibles inconvenients que puguin aparèixer.

#### 5.1.1. Gestió del Projecte

Aquest bloc abasta totes les tasques relacionades amb la gestió integral del projecte, incloent-hi la seva documentació, estilització i preparació prèvia. Aquestes tasques es desenvoluparan de manera contínua durant tota la durada del projecte, en paral·lel amb altres activitats. Es calcula una durada total aproximada de 159 hores.

- **GP1 - Contextualització i Abast:** Consisteix en la primera tasca del projecte, que implica l'elaboració del primer document. En aquest document es realitza un estudi previ del projecte per poder definir l'abast i en conseqüència els objectius, i metodologia del treball. Es calcula que la dedicació serà d'unes 22 hores.
- **GP2 - Planificació Temporal:** Aquesta tasca té com a objectiu definir una planificació temporal inicial que serveixi com a guia de progrés durant el desenvolupament del projecte. La planificació es divideix en diferents tasques a fer on s'indica la tipologia, durada prevista per finalitzar la tasca i l'anàlisi dels possibles riscos i obstacles que poden aparèixer i la seva possible remediació. Es calcula que la dedicació serà d'unes 15 hores. Dependència de GP1.
- **GP3 - Pressupost i sostenibilitat:** En aquesta tercera tasca, es defineix el pressupost necessari per dur a terme el projecte. En ell s'inclouen els costos de realització,

l'impacte econòmic i ambiental. Es calcula que la dedicació serà d'unes 12 hores. Dependència de GP2.

- **GP4 - Revisió de la documentació relativa a la preparació prèvia:** Aquesta és l'última tasca pertanyent a la Gestió del Projecte, on es realitza un recull de totes les entregues prèvies realitzades (GP1, GP2, GP3). En aquesta entrega, es porten a cap totes les modificacions necessàries per a millorar les entregues ja fetes. Es calcula que la dedicació serà d'unes 8 hores. Dependència de GP1, GP2 i GP3.
- **GP5 - Documentació de la fase de Disseny inicial i desenvolupament:** Es redacta tota la informació referent a les accions dutes a terme durant el desenvolupament del projecte, juntament amb la informació obtinguda pel desenvolupament del disseny inicial. Aquesta tasca s'iniciarà quan s'hagi dut a terme un 20% de les tasques D1 i DP2, ja que, es considera que a partir d'aquest moment ja es tindrà informació suficient per a començar a desenvolupar-la. Es calcula que la dedicació serà d'unes 32 hores. Dependència de D1 i DP2.
- **GP6 - Reunions amb el director:** Durant el projecte, es fan reunions periòdiques amb el director de projecte a mètode de control i rectificació d'aquest. La freqüència de les reunions serà cada quinze dies, però pot variar segons l'evolució del projecte. Es farà servir l'eina de Microsoft Teams per dur a terme les reunions i es calcula que la dedicació serà d'unes 20 hores.
- **GP7 - Memòria:** Aquesta tasca consisteix en la preparació de la documentació a entregar al final del projecte. La documentació inclou la documentació generada anteriorment, GP4, i la generada progressivament durant el projecte, GP5. Es calcula que la dedicació serà d'unes 20 hores. Dependència de GP5.
- **GP8 - Preparació Defensa del Projecte:** Un cop finalitzada la part tècnica i documental, es prepararà tota la documentació gràfica de suport necessària per a la defensa del projecte. Al mateix temps, es duran a terme assajos per preparar la presentació de la defensa oral. Es calcula que la dedicació serà d'unes 30 hores. Dependència de GP7.

### **5.1.2. Recerca i Investigació**

Aquest segon bloc s'enfoca en les tasques relatives a la recerca i investigació, que inclou la lectura d'articles, llibres, informes, visualització de vídeos i consulta de diverses fonts d'informació relacionades amb el tema del projecte. Es calcula una durada total aproximada de 7 hores per aquest bloc.

- **R1 - Estudi de mercat:** Aquesta tasca implica realitzar una investigació sobre les solucions existents al mercat, analitzant els serveis i funcions que ofereixen, com estan programats i recopilar informació rellevant pel disseny inicial. Es calcula que la dedicació serà d'unes 7 hores.

### 5.1.3. Disseny del Projecte

Aquest tercer bloc aborda les tasques relatives a la definició tant de la primera com de la versió final del disseny de l'aplicació. Es calcula una durada total aproximada de 40 hores per aquest bloc.

- **D1 - Disseny Inicial:** Aquesta tasca implica la creació d'un disseny inicial basat en la informació recollida durant la fase de recerca. Aquesta inclou, esbossos inicials, llistes de necessitats o dissenys rudimentaris. Per aquesta tasca s'utilitzarà l'aplicació Qt Designer 5 i es calcula una dedicació d'unes 21 hores.
- **D2 - Disseny Final:** Aquesta tasca recull de totes les modificacions realitzades al llarg del projecte per adaptar el disseny inicial a les necessitats concretes, així com la definició del disseny definitiu. Per aquesta tasca es farà servir l'aplicació Qt Designer 5 i es calcula una dedicació d'unes 19 hores. Dependència de D1 i DP3.

### 5.1.4. Desenvolupament del Projecte

Aquest bloc representa la fase clau del projecte. Englobant totes les tasques relatives a la preparació, programació, configuració del programari creat, així com, la instal·lació d'aquest sobre el maquinari escollit. També implica la realització de proves exhaustives per garantir el funcionament adequat del sistema. Es calcula una durada total aproximada de 229 hores.

- **DP1 - Instal·lació del programari:** Aquesta tasca implica la recerca de tot el programari necessari per al desenvolupament de l'aplicació, juntament amb la seva posterior descarrega i configuració. Es calcula una dedicació d'unes 5 hores. Dependència de D1.
- **DP2 - Desenvolupament inicial:** En aquesta fase, es realitza la primera implementació del codi en consonància amb el disseny realitzat, adaptant els elements de l'aplicació a les seves funcionalitats en concordança amb els requisits bàsics de l'aplicació. Aquesta tasca es desenvolupa al mateix temps que la tasca DP6. Es calcula una dedicació d'unes 40 hores. Dependència de DP1.
- **DP3 - Desenvolupament funcions:** En aquesta fase, es duu a terme la implementació de totes les funcions principals del codi. Aquesta tasca es desenvolupa al mateix temps que la tasca DP6. Es calcula una dedicació d'unes 40 hores. Dependència de DP2.
- **DP4 - Desenvolupament final:** Aquesta tasca se centra en desenvolupament de les funcionalitats addicionals del projecte per generar una millor experiència d'usuari. Es desenvolupa al mateix temps que la tasca DP6. Aquesta tasca es podrà començar a desenvolupar un cop s'hagi finalitzat la tasca DP3 i la tasca D2 s'hagi implementat en un 70%. Es calcula una dedicació d'unes 50 hores. Dependència de DP3 i D2.
- **DP5 - Desenvolupament de millores:** Aquesta tasca consisteix a dur a terme revisions del codi per optimitzar els processos de les funcions desenvolupades.

Aquesta tasca es duu a terme simultàniament amb la tasca DP6 i es calcula una dedicació d'unes 50 hores. Dependència de DP4.

- **DP6 – Testing:** Aquesta tasca se centra principalment a verificar el correcte funcionament de les diferents funcionalitats desenvolupades i la modificació del codi per resoldre qualssevol errors detectats. Aquesta tasca s'anirà desenvolupant a partir que es realitzi un 25% de les tasques DP3, DP4 i DP5, ja que aquestes estaran prou avançades per a poder mostrar els primers resultats congruents. Es calcula una dedicació d'unes 30 hores. Dependència de DP3, DP4 i DP5.
- **DP7 - Adaptació de sistemes:** Aquesta tasca implica la modificació del codi per adaptar-lo als diferents sistemes operatius. Es calcula una dedicació d'unes 14 hores. Dependència de DP4.

A mesura que es desenvolupi una nova funcionalitat i aquesta sigui independent, ja sigui en la tasca DP2, DP3, DP4 o DP5 es procedirà a la tasca DP6 per comprovar el seu correcte funcionament, abans de començar a crear o millorar una altra de nova. Per totes les tasques de desenvolupament, adaptació i testing, es farà servir l'eina Visual Studio Code perquè permet l'edició i implementació de codi amb facilitat.

## 5.2. Recursos

Donada la naturalesa del projecte, el qual es desenvolupa en una petita escala, els recursos disponibles són limitats. Tots els rols i responsabilitats del projecte seran assumides per l'autor del projecte, incloent-hi el rol de cap de projecte, encarregat de la planificació i gestió, així com els rols d'investigador, desenvolupador i tester del projecte. El director del projecte col·laborarà aportant la seva experiència pel desenvolupament del projecte.

Quant als recursos materials, farem servir un ordinador amb dos sistemes operatius, en mode dual-boot, per poder treballar tant en Ubuntu, Linux com amb Windows 10, juntament amb les aplicacions necessàries. Pel que fa al programari de desenvolupament, s'utilitzarà Visual Studio Code i Qt Designer 5, mentre que per la gestió del projecte s'emprarà, Microsoft Teams i Celoxis, així com la creació de diagrames mitjançant, l'aplicació web GanttPro. A més, es disposarà d'un espai de treball que inclourà dues pantalles per augmentar l'eficiència, així com una bona connexió a internet.

### 5.3. Estimació de les tasques

A continuació es mostren llistades totes les tasques de manera resumida a la Taula 1. Indicant el seu identificador, nom de la tasca, dedicació en hores i les seves dependències.

Identificador	Tasca	Dedicació	Dependències
<b>Gestió del Projecte</b>		<b>159 hores</b>	
GP1	Contextualització i Abast	22 hores	
GP2	Planificació Temporal	15 hores	GP1
GP3	Pressupost i sostenibilitat	12 hores	GP2
GP4	Revisió de la documentació relativa a la preparació prèvia	8 hores	GP1, GP2, GP3
GP5	Documentació de la fase de Disseny inicial i desenvolupament	32 hores	D1, DP2
GP6	Reunions amb el director	20 hores	
GP7	Memòria	20 hores	GP5
GP8	Preparació Defensa del Projecte	30 hores	GP7
<b>Recerca i Investigació</b>		<b>7 hores</b>	
R1	Estudi de mercat	7 hores	
<b>Disseny del Projecte</b>		<b>40 hores</b>	
D1	Disseny Inicial	21 hores	
D2	Disseny Final	19 hores	D1, DP3
<b>Desenvolupament del Projecte</b>		<b>229 hores</b>	
DP1	Instal·lació del programari	5 hores	D1
DP2	Desenvolupament inicial	40 hores	DP1
DP3	Desenvolupament funcions	40 hores	DP2
DP4	Desenvolupament final	50 hores	D2, DP3
DP5	Desenvolupament de millores	50 hores	DP4
DP6	Testing	30 hores	DP3, DP4, DP5
DP7	Adaptació de sistemes	14 hores	DP4

*Taula 1. Taula resum de Tasques*  
*Font. Elaboració pròpia.*

## 5.4. Diagrama de Gantt

A continuació es presenta el diagrama de Gantt elaborat a partir de les tasques definides pel projecte. Aquest diagrama ha sigut creat amb GanttPro:



Figura 3. Diagrama de Gantt del Projecte  
Font. Elaboració pròpia.

## 5.5. Gestió dels Riscos

Un dels desafiaments principals d'una bona planificació és anticipar i considerar els possibles contratemps que poden sorgir durant el projecte. Per aquest motiu, és crucial generar un pla de resposta per afrontar incidents o imprevistos que ens permeti resoldre aquests obstacles mantenint el desenvolupament de les tasques i el projecte en el termini previst. Com s'esmenta en el primer lliurament, en aquest apartat definim les solucions als possibles obstacles i riscos identificats. En la següent taula es mostren els riscos, el seu impacte i mitigació.

Risc	Probabilitat	Impacte	Mitigació
Planificació errònia	Baixa (10%)	Baix	En el marc de l'ús d'una metodologia iterativa, cada fase del projecte s'inicia quan acaba la fase anterior. Això implica que, si es produeix un contratemps durant alguna de les tasques, caldrà ajustar la planificació una vegada s'hagi completat aquesta tasca, recalculant les noves hores de dedicació per cada tasca.
Desconeixement de tecnologies	Mitjana (55%)	Mitjà	En cas que sigui necessari més temps per aprendre o perfeccionar alguna tecnologia específica, existeix la possibilitat d'incrementar el nombre d'hores dedicades al projecte o revisar la planificació per ajustar-la al nombre d'hores destinades a cada tasca.
Error (Bugs)	Alta (80%)	Mitjà	El fet de fer servir la metodologia iterativa, és comú que es detectin errors cada vegada que es prova una funció. Per sort, els errors de codi no acostumen a ser grans i acostumen a estar identificats en el missatge d'error, el seu origen. Això fa que sigui poc probable la seva afectació a la planificació. En el cas d'errors d'implantació de codi, aquests sí que poden afectar a la planificació temporal pel fet que poden suposar una reestructuració de part del codi. Aquesta dedicació extra de temps per resoldre aquests errors, ja estan contemplades dins les hores de desenvolupament i testatge.
Desplegament de l'Arquitectura Distribuïda	Mitjana (30%)	Alt	En cas que la transició del producte d'un sistema operatiu a un altre presenti problemes, es requeriran una sèrie de mesures que afectaran la planificació del temps i el pressupost. Això implica una reestructuració del pla de treball, amb la possible necessitat d'incrementar les hores dedicades al projecte, tenint un impacte

			directe en el pressupost a causa del cost addicional del treballador o, si és necessari, la contractació d'un especialista en la tecnologia o la reconstrucció de l'aplicació des de zero per al nou sistema operatiu.
Problemes de comunicació	Mitjana (35%)	Mitjà	Una comunicació deficient entre el client final i el desenvolupador pot tenir un impacte directe sobre el resultat final del producte. És fonamental mantenir un contacte constant amb l'usuari final a través de reunions per evitar aquesta problemàtica. En cas que sigui necessari, es podria reduir l'interval de temps entre les reunions per millorar la comunicació i prevenir confusions o seguir camins de desenvolupament inadequats.
Acceptació de l'Usuari Final	Baixa (15%)	Alt	És possible que, un cop l'aplicació estigui desenvolupada, l'usuari final no la consideri adequada per la tasca per la qual s'ha desenvolupat i, per tant, no faci l'ús complet com s'esperava. No obstant això, mitjançant les reunions amb el responsable del projecte, juntament amb algun treballador de l'empresa, podem abordar aquesta problemàtica i minimitzar-ne els efectes. En cas que hi hagués algun problema addicional, sempre hi ha l'opció de solucionar-ho mitjançant actualitzacions de l'aplicació, proporcionant una nova versió que pugui abordar les necessitats o les preocupacions.

*Taula 2. Taula sobre els Riscos i la seva mitigació  
Font. Elaboració pròpia.*

## 5.6. Actualització Planificació Temporal

Durant la realització del projecte, hi ha hagut diversos canvis en la planificació temporal inicial. Aquests ajustos han estat necessaris per a assegurar el desenvolupament correcte i funcional del projecte. Els canvis són els següents:

A la secció **Gestió del Projecte** s'han actualitzat dues tasques:

- **Reunions amb el Director (GP6).** Les reunions amb el director s'han reduït de 20 hores a 15 hores. Aquest ajust ha estat possible gràcies a una planificació acurada de contingut a tractar, permetent que es necessitessin menys hores del previst inicialment. Destacaria l'eficàcia i claredat en la comunicació.
- **Redacció de la Memòria (GP7).** S'han dedicat 5 hores addicionals a la redacció de la memòria, augmentant el temps total de 20 a 25 hores. Aquest increment s'ha produït a

causa de la necessitat de realitzar modificacions, millorar la redacció, elaborar taules i millorar l'estil i format general del document.

A la secció **Desenvolupament del Projecte** s'ha actualitzat dues tasques:

- **Desenvolupament de Funcions (DP3)**. El desenvolupament de certes funcions ha requerit 10 hores més del previst, passant de 40 a 50 hores. Aquest augment es deu a contratemps com ara llibreries que no funcionaven correctament i errors en el plantejament inicial del codi. Aquests problemes han estat solucionats amb una depuració exhaustiva i la reescriptura necessària per garantir el correcte funcionament de les funcions.
- **Adaptació de sistemes (DP7)**. L'adaptació del codi pel sistema operatiu Windows ha requerit 7 hores més del previst, passant de 14 a 21 hores. Aquest increment es deu a contratemps amb la interpretació del codi per part del sistema operatiu i la necessitat de redactar nou codi per corregir problemes que no afectaven el sistema Linux. Aquests inconvenients s'han solucionat amb una revisió detallada del codi i la reescriptura necessària per assegurar el correcte funcionament de les funcions.

En la taula 3 es mostren les noves dedicacions un cop finalitzades cadascuna de les tasques.

Identificador	Tasca	Dedicació	Dependències
<b>Gestió del Projecte</b>		<b>159 hores</b>	
GP6	Reunions amb el director	15 hores	
GP7	Memòria	25 hores	GP5
<b>Desenvolupament del Projecte</b>		<b>229 hores</b>	
DP3	Desenvolupament funcions	50 hores	DP2
DP7	Adaptació de sistemes	21 hores	DP4

*Taula 3. Modificació de la Taula resum de Tasques*  
*Font. Elaboració pròpia.*

En conjunt, els ajustos en el càlcul d'hores de les seccions de gestió del projecte i desenvolupament han estat possibles gràcies a una gestió rigorosa del temps. Això ha permès que no s'observessin desviacions significatives respecte a la planificació inicial, mantenint el projecte en el camí correcte.

## 6. Gestió econòmica del projecte

Cada projecte necessita un pressupost que tingui en compte els costos econòmics relacionats amb el seu desenvolupament. Aquesta secció proporcionarà una enumeració dels recursos humans i tecnològics necessaris, així com els seus costos corresponents. A més, s'inclouran els costos derivats, com ara les contingències o imprevistos que poden sorgir durant el transcurs del projecte. Finalment, es defineix un conjunt de mètriques a seguir en cas que hi hagi desviacions respecte al pla inicial.

### 6.1 Pressupost

A l'hora de desenvolupar un projecte, és imprescindible delinear amb precisió tots els recursos requerits i estimar el cost associat a la seva utilització o adquisició. Aquests recursos poden ser categoritzats principalment en tres àmbits: el personal, el material (incloent-hi maquinari i programari) i els recursos generals.

#### 6.1.1. Personal

Els costos associats al personal corresponen als salaris dels diferents treballadors de la plantilla de l'empresa. En aquest projecte concret, l'únic treballador serà el mateix autor del projecte, assumint diferents rols. En un cas real, cada rol estaria desenvolupat per un treballador diferent. Per poder realitzar el càlcul dels costos de personal, s'ha utilitzat la pàgina web Tecnoempleo (1), la qual ofereix una visió de les mitjanes salarials per a cada professió i rol. Tenint en compte on es desenvolupa el projecte, els paràmetres fets servir seran els rols descrits a continuació en la zona de Barcelona. A la següent taula es mostren els resultats obtinguts:

Rol	Salari Anual	Salari/Hora	Salari+ Seguretat Social
Cap de projecte (C)	56.000 €	31,11 €	41,38 €
Programador (P)	43.500 €	24,17 €	32,14 €
Dissenyador UI/UX (D)	38.907 €	21,62 €	28,75 €
Tester (T)	33.000 €	18,33 €	24,38 €

*Taula 4. Taula de Salaris per Rol desenvolupat al Projecte*

*Font. Elaboració pròpia.*

El salari anual representa el salari en brut per a cada posició, mentre que el Salari/Hora indica aquest salari en brut dividit per les hores de treball anual, estimades en unes 1800 hores per any per a cada posició. A més, a la darrera columna es detalla el salari incrementat amb la contribució a la Seguretat Social que assumeix l'empresa, suposant un 33% addicional sobre el salari anual del treballador. Aquest salari es calcula multiplicant el salari brut per 1,33.

(1) Tecnoempleo: <https://www.tecnoempleo.com/tecnocalculadora.php>

Ara que disposem del cost per hora de cada treballador, es pot fer una estimació aproximada del cost econòmic total del projecte en relació amb el personal. En la següent taula es mostra una estimació econòmica tenint en compte cada tasca i qui la desenvolupa.

Identificador	Tasca	Dedicació	Rols	Cost
<b>Gestió del Projecte</b>		<b>159 hores</b>		<b>6.093,50 €</b>
GP1	Contextualització i Abast	22 hores	C	910,36 €
GP2	Planificació Temporal	15 hores	C	620,70 €
GP3	Pressupost i sostenibilitat	12 hores	C	496,56 €
GP4	Revisió de la documentació relativa a la preparació prèvia	8 hores	C	331,04 €
GP5	Documentació de la fase de Disseny inicial i desenvolupament	32 hores	C/P/D/T 8/8/8/8	1.013,20 €
GP6	Reunions amb el director	20 hores	C/P/D 4/8/8	652,64
GP7	Memòria	20 hores	C	827,60 €
GP8	Preparació Defensa del Projecte	30 hores	C	1.241,40 €
<b>Recerca i Investigació</b>		<b>7 hores</b>		<b>289,66 €</b>
R1	Estudi de mercat	7 hores	C	289,66 €
<b>Disseny del Projecte</b>		<b>40 hores</b>		<b>1.352,08 €</b>
D1	Disseny Inicial	21 hores	C/D 9/12	717,42 €
D2	Disseny Final	19 hores	C/D 6/13	624,66 €
<b>Desenvolupament del Projecte</b>		<b>229 hores</b>		<b>7.127,26 €</b>
DP1	Instal·lació del programari	5 hores	P	160,70 €
DP2	Desenvolupament inicial	40 hores	P	1.285,60 €
DP3	Desenvolupament funcions	40 hores	P	1.285,60 €
DP4	Desenvolupament final	50 hores	P	1607 €
DP5	Desenvolupament de millores	50 hores	P	1607 €
DP6	Testing	30 hores	T	731,40 €
DP7	Adaptació de sistemes	14 hores	P	449,96 €
<b>TOTAL</b>				<b>14.862,50 €</b>

Taula 5. Taula dels Costos Humans de l'Estimació de les Tasques

Font. Elaboració pròpia.

### 6.1.2. Recursos de Maquinari

Per al desenvolupament del projecte, serà necessari un conjunt de maquinari per poder dur-lo a terme. Com ja s'ha comentat prèviament, l'autor d'aquest projecte serà l'únic participant, per la qual cosa tot el maquinari consistirà d'una sola unitat, en un cas real, aquesta quantitat es multiplicaria per quatre. En cas que fos necessari més d'un element de cada tipologia per aquest projecte, s'indicaria a la taula que es mostrarà a continuació:

Dispositiu	Preu	Vida Útil	Amortització
Asus TUF Gaming FX505DT-BQ624	648,85 €	4 anys	39,20 €
<b>Total</b>			39,20 €

*Taula 6. Taula de Costos i Amortització del Maquinari*

*Font.* Elaboració pròpia.

Per calcular l'amortització d'un producte tecnològic, s'utilitza una fórmula que pren en consideració diversos valors. En primer lloc, s'ha de tenir en compte el cost del dispositiu, el qual s'obté en el moment de la seva compra. Aquest es multiplica per la quantitat d'hores que es dedicarà al projecte, en aquest cas 435 hores. A continuació, aquest resultat es divideix per la vida útil del producte, que en el cas dels articles tecnològics sol ser de quatre anys, multiplicat pel nombre de dies laborables i les hores diàries de treball, els quals es prenen com a 225 dies i 8 hores respectivament.

$$\frac{\text{Cost dispositiu}(\text{€}) * \text{Durada TFG}(\text{hores})}{\text{Vida útil}(\text{anys}) * \text{Dies laborals} / \text{any}(\text{dies}) * \text{Dedicació diària}(\text{hores})}$$

### 6.1.3. Recursos de Programari

Durant el desenvolupament del projecte es requerirà un seguit de programari per la seva realització. A continuació, es mostra una taula amb els costos del programari de pagament:

Programari	Preu
Microsoft Teams	247,20 €
<b>Total</b>	247,20 €

*Taula 7. Taula de Costos de Programari.*

*Font.* Elaboració pròpia.

Microsoft Teams, es troba inclosa en el paquet de Microsoft 365. En aquest cas, i tenint en compte que el projecte es desenvolupa dins d'una empresa especialitzada en la seguretat informàtica i que en ella es tracten dades crítiques, és crucial optar per la subscripció que ofereixi el màxim nivell de seguretat. En aquest context, la subscripció adequada seria la

Microsoft 365 Enterprise Prèmium que és d'ús individual. En un cas real farien falta 4 subscripcions.

#### 6.1.4. Despeses Generals

Les despeses generals són aquelles que estan relacionades amb elements presents en qualsevol projecte com la llum, internet o el mobiliari. Aquest projecte es desenvolupa des de casa, per tant, no es realitzen desplaçaments i tampoc hi ha necessitat de llogar o comprar cap espai addicional. Únicament cal comprar el material necessari per poder desenvolupar el projecte sense problemes. Tots els costos relatius a aquests elements es mostren en la següent taula:

Recurs	Preu	Amortització
Llum	11,20 €	
Internet	162,67 €	
Cadira	151 €	29,19 €
Taula	39 €	7,54 €
Pantalla	199 €	96,18 €
Làmpada	56,70 €	10,96 €
<b>Total</b>	<b>619,31 €</b>	<b>317,74 €</b>

*Taula 8. Taula de Costos i Amortització d'elements generals*

*Font. Elaboració pròpia.*

El càlcul del cost de l'electricitat es basa en l'examinació del consum de cada dispositiu electrònic i el preu mitjà de l'electricitat a Espanya, ja que els costos energètics varien al llarg del temps i necessitem una referència fixa. Un cop això, s'agafa el consum de l'ordinador, 150 W per hora resultant en 0,15 kWh; per a la pantalla, 24 W, suposant uns 0,024 kWh; i per la làmpada, amb un consum de 35 W, suposant uns 0,035 kWh. Finalment, la suma de tots aquests consums es multiplica pel preu mitjà de la llum, 0,1232 € i pel nombre d'hores dedicades al Treball de Fi de Grau (TFG), obtenint un total de consum elèctric durant el TFG, que és d'11,20 €.

$$0,209 \text{ kWh} * 0,1232 \text{ €/kWh} * 435 \text{ h} = 11,20 \text{ €}.$$

D'altra banda, cal tenir en compte el consum de xarxa, que es calcula dividint la tarifa mensual contractada, 40 €/mes, entre 30, suposant que un més té trenta dies. Aquest cost diari es multiplica pel total de dies aproximats que es calcula que s'estarà realitzant el projecte, 122 dies, suposant un total de 162,67 €.

Finalment, es calcula el cost d'amortització de tots els elements pertanyents a la zona de treball, com la taula, la cadira, la pantalla i la làmpada. Per calcular l'amortització, es multiplica el preu de cada element per la durada del TFG i es divideix tot per la vida útil dels elements, sent de 4 anys per la pantalla i de 10 anys pel mobiliari, ajustat pel nombre de dies laborables anuals, que són 225 dies. A continuació, es mostra el càlcul utilitzat:

$$\frac{\text{Cost Moble}(\text{€}) * \text{Durada TFG}(\text{hores})}{\text{Vida útil}(\text{anys}) * \text{Dies laborals/any}(\text{dies})}$$

### 6.1.5. Cost de Contingència

Els costos de contingència es refereixen a aquells associats amb la previsió de despeses derivades d'imprevistos no anticipats. Aquest cost s'acostuma a calcular mitjançant una anàlisi de riscos; no obstant això, en el nostre cas, optarem per calcular-lo utilitzant percentatges, assumint un percentatge del 10% per al nostre projecte. Aquest percentatge es justifica per la naturalesa del projecte, on es minimitza l'ús de materials propensos a fallar, amb l'excepció dels equips informàtics. Per tant, el cost de contingència es calcula multiplicant tots els costos anteriors, que ascendeixen a 15.466,64 €, per 0,1 que correspon al percentatge.

$$(\text{Cost personal} + \text{Cost Maquinaria} + \text{Cost Programari} + \text{Cost General}) * 0,1 = 1.546,664$$

### 6.1.6. Cost d'Imprevistos

Durant la secció de gestió de riscos, es presenten una sèrie de contratemps que poden sorgir durant el procés de desenvolupament del projecte, juntament amb les accions a prendre per abordar-los. Aquests inconvenients i la seva solució impliquen uns costos addicionals a causa de la desviació temporal que comporten, generant un impacte directe en el pressupost. Durant aquest apartat, es calculen els costos addicionals que implica la resolució de qualsevol imprevist. Per fer-ho, es fa servir la següent fórmula:

$$\text{Probabilitat d'ocurrència}(\%) * \text{Sou del Rol/Tasca}(\text{€}) * \text{Dedicació extra}(\text{hores}) = \text{Cost extra}(\text{€}).$$

En la següent taula es mostra la predicció de Costos d'Imprevistos:

Risc	Probabilitat	Rol	Dedicació	Cost
Planificació errònia	10%	C	5 hores	20,69 €
Desconeixement de tecnologies	55%	P	10 hores	176,77 €
Error (Bugs)	80%	P/T	7 hores	158,26 €
Desplegament de l'Arquitectura Distribuïda	30%	P/T	8 hores	67,82 €
Problemes de comunicació	35%	C	5 hores	72,42 €
Acceptació de l'Usuari Final	15%	C/P	5 hores	27,57 €
<b>Total</b>				<b>523,53 €</b>

Taula 9. Taula de predicció de Costos d'Imprevistos

Font. Elaboració pròpia.

Per abordar les despeses imprevistes i inesperades, s'estableixen partides pressupostàries específiques per contemplar les contingències durant el desenvolupament del projecte. En el

cas que no es produeixin esdeveniments inesperats o si aquests són menors dels esperats, el pressupost no utilitzat serà deduït del total final. Això comportarà que el pressupost final acabarà sent més baix però més acurat a la realitat. Aquesta mesura s'adopta perquè s'ha sobreestimat els costos, els quals es basen en càlculs associats a variables poc probables. Com a resultat, els fons reservats no fets servir s'eliminaran del pressupost, el qual es modificarà adaptant-lo a les noves necessitats.

### 6.1.7. Estimació Resum Final

A continuació es mostra una taula amb el resum dels costos del projecte:

Personal	Maquinari	Programari	Generals	Contingència	Imprevistos	Total
14.862,50 €	39,20 €	247,20 €	317,74 €	1.546,664 €	523,53 €	17.536,834 €

*Taula 10. Taula de Resum dels Costos*

*Font. Elaboració pròpia.*

## 6.2. Control de Gestió

Malgrat tenir un pressupost definit, és essencial fer un seguiment del mateix per assegurar-ne el compliment. Per això, establim un conjunt de paràmetres que permeten observar i calcular la seva desviació. Durant el projecte, poden aparèixer inconvenients com els ja mencionats amb anterioritat, que provoquin desviacions respecte al pressupost inicial. Aquestes desviacions queden definides pels paràmetres esmentats i es calcularan al llarg del projecte.

- **Desviació d'hores dedicades a una tasca:** El càlcul ens permet identificar si s'han necessitat més o menys hores del previst inicialment. El càlcul de la desviació es realitzarà cada cop que es finalitzi una tasca, revisant el total d'hores destinades a aquesta. En cas que la tasca tingues una llarga durada, a la meitat de la mateixa es farà una valoració parcial per veure si existeix algun tipus de desviació.

$$\text{Hores estimades tasca} - \text{Hores reals tasca}$$

- **Desviació d'hores totals del projecte:** Aquesta desviació ens dona una visió global de com el temps dedicat s'ajusta al planificat. Aquesta revisió es realitzarà un cop s'hagi acabat el projecte per poder ser més precís, al mateix temps que es tindrà en compte les desviacions de cada tasca com a mesurador parcial del projecte.

$$\text{Hores estimades totals projecte} - \text{Hores reals totals projecte}$$

- **Desviació en cost per la realització de la tasca:** Aquest càlcul obté una estimació del cost addicional o estalviat. El càlcul de la desviació es realitzarà cada cop que es finalitzi una tasca, revisant el cost d'aquesta. En cas que la tasca tingues una llarga durada, a la meitat de la mateixa es farà una valoració parcial per veure si existeix algun tipus de desviació.

$$(\text{Hores estimades tasca} - \text{Cores reals tasca}) * \text{Cost tasca}$$

- **Desviació del cost general:** Ens permet identificar si s'han superat o estalviat recursos respecte a la previsió inicial. Mantenint un registre dels costos inclosos en el projecte, com ara el material utilitzat i altres despeses relacionades.

Cost estimat genèric - Cost real genèric

- **Desviació de costos imprevistos:** Aquesta desviació ens ajuda a gestionar els riscos i a mesurar la precisió de les nostres estimacions per a aquestes eventualitats.

Cost estimat imprevistos - Cost real imprevistos

- **Desviació total del pressupost:** Ens permet avaluar si el projecte s'està gestionant dins dels marges econòmics previs o si és necessari prendre mesures correctives. Aquesta revisió es realitzarà un cop s'hagi acabat el projecte per poder ser més precís, al mateix temps que es tindrà en compte les desviacions de cada tasca com a mesurador parcial del projecte.

Cost estimat projecte - Cost real projecte

Qualsevol desviació pot tenir tant repercussions positives com negatives, segons com es reflecteixi en el balanç final. Si aquest balanç és negatiu i supera el 25% de la desviació prevista, caldrà intervenir, ja que això ja suposaria un problema per al desenvolupament de la tasca o per al projecte en conjunt. Les accions que emprendrem per abordar aquestes desviacions en els indicadors del projecte poden variar segons la naturalesa específica de la desviació i el seu impacte en el projecte. Això inclouria la replanificació de terminis, reassignació de recursos, com la contractació de personal o utilització de recursos alternatius. Reforçar la gestió de riscos per mitigar possibles amenaces al projecte, implementant plans de contingència o actualització de l'anàlisi de riscos. Les mesures que prendrem dependran de la naturalesa i de l'impacte de la desviació en el projecte i les abordarem de manera individual i per adoptar accions de manera més efectiva.

### 6.3. Actualització Gestió Econòmica

Després de revisar la planificació temporal del projecte, s'ha dut a terme una actualització dels costos del projecte per garantir una execució eficient i acurada. A continuació, es detalla com els canvis realitzats han afectat les seccions modificades i les seves repercussions finals:

- Pel que fa a la secció de **Gestió del Projecte** el cost ha disminuït un 0,23%.
- Pel que fa a la secció **Desenvolupament del Projecte** les hores invertides han augmentat en un 7,42% que repercuteix en un increment sobre el cost final tant en costos humans com energètics.

En la taula 11 es pot veure el cost final d'aquestes seccions:

Identificador	Tasca	Dedicació	Rols	Cost
<b>Gestió del Projecte</b>		<b>159 hores</b>		<b>6.079,64 €</b>
GP6	Reunions amb el director	15 hores	C/P/D 3/4/8	606,84 €
GP7	Memòria	25 hores	C	1.034,50 €
<b>Desenvolupament del Projecte</b>		<b>246 hores</b>		<b>7.679,64 €</b>
DP3	Desenvolupament funcions	50 hores	P	1.607 €
DP7	Adaptació de sistemes	21 hores	P	674,94 €
<b>TOTAL</b>				<b>15.395,02 €</b>

*Taula 11. Modificació de la Taula dels Costos Humans de l'Estimació de les Tasques*  
*Font. Elaboració pròpia.*

- Pel que fa als **costos de contingència**, s'ha utilitzat la mateixa fórmula establerta en la planificació inicial, modificant el camp «costos de personal» amb el nou valor obtingut:

$$(Cost\ personal + Cost\ Maquinaria + Cost\ Programari + Cost\ General) * 0,1 = 1,599.916\ €$$

Això ha resultat en una despesa extra de 53,252 €. Tot i això, l'ús dels costos de contingència no seran necessaris i, per tant, no estaran inclosos en el resum de costos finals.

L'actualització dels **costos finals** del projecte posa de manifest la capacitat per adaptar el projecte en funció dels canvis proposats i els imprevistos al llarg del seu desenvolupament, garantint alhora la cobertura completa de les necessitats del projecte. El cost final del projecte seria de 16.523,13 euros.

La taula 12 es mostra el resum dels **costos finals**, reflectint aquestes adaptacions.

Personal	Maquinari	Programari	Generals	Imprevistos	Total
15.395,02 €	39,20 €	247,20 €	318,18 €	523,53 €	16.523,13 €

*Taula 12. Taula de Resum dels Costos*  
*Font. Elaboració pròpia.*

## 7. Sostenibilitat del Projecte

Després de completar l'enquesta del PROJECTE EDINSOST2-ODS[9], he reflexionat sobre la importància cabdal de la sostenibilitat en la gestió de projectes. Si bé ja era conscient de la necessitat d'una transició cap a un estil de vida més sostenible per a la conservació de la biodiversitat del planeta, he arribat a comprendre, després de realitzar una breu investigació, que aquesta transició implica canvis en els hàbits quotidians, l'ús responsable dels materials i la gestió adequada dels residus, així com la consideració de les fonts d'energia renovable, entre altres mesures. Encara que aquests aspectes estaven presents en el meu pensament i accions quotidianes, mai havia considerat la seva rellevància dins del sector de les Tecnologies de la Informació i la Comunicació (TIC), on acostuma a predominar l'impacte econòmic sobre el mediambiental. Aquesta percepció, en gran part, està influïda pel nostre entorn, on la contaminació generada per la recol·lecció, producció i eliminació dels materials utilitzats en aquest sector va en augment. Això ha generat un fort interès i una ideologia centrada en les tres «Rs», reducció, reutilització i reciclatge.

A través de l'enquesta, he observat que en la majoria dels casos som capaços de comprendre i treballar amb mesures per reduir l'impacte en l'àmbit de la sostenibilitat. No obstant això, crec que molts de nosaltres no sabem aplicar indicadors o mesures per analitzar l'impacte real i arribar a conclusions precises. Per això, un dels objectius d'aquest projecte serà millorar la sostenibilitat al màxim possible.

D'altra banda, un altre aspecte interessant a investigar és l'impacte mediambiental dels productes tecnològics un cop es descarten. Si bé reconeixem la problemàtica tant mediambiental com social associada amb la recol·lecció de minerals i la fabricació de productes tecnològics, sovint oblidem considerar el que succeeix amb aquests productes un cop són eliminats. Tot i que pot semblar inesperat, els elements tecnològics continuen generant residus i contaminació un cop ja no són útils i es descarten. Això és pel fet que els components que el conformen, poden alliberar substàncies nocives pel medi ambient que poden intoxicar tant el sòl com l'aire. És per això, que serà important revisar la gestió dels residus que es puguin generar derivats d'aquest projecte, sempre que no puguin ser reutilitzables.

### 7.1. Dimensió Econòmica

L'aspecte econòmic ha estat objecte d'una atenció especial, i el pressupost s'ha elaborat prenent en consideració i estimant tots els possibles costos associats al projecte. Donada la seva naturalesa, inclosa la fase d'investigació i desenvolupament, s'han utilitzat totes les eines gratuïtes i de qualitat per tal de reduir costos. A més, el pressupost s'ajusta estrictament a les despeses essencials, evitant així l'ús innecessari de recursos.

S'ha realitzat una estimació dels costos de realització del projecte a partir dels recursos humans, materials i de programari necessaris per poder desenvolupar el projecte. Un cop realitzat el projecte, s'ha revisat l'estimació inicial de costos i s'han modificat aquests costos per adaptar-los al cost real del projecte. En la secció 6.1 es pot trobar l'anàlisi inicial i final dels costos juntament amb les justificacions de les diferències.

Els principals costos durant la vida útil serien l'actualització del programari a noves versions dels diferents sistemes operatius o tipologies d'arxius. Tot i això, aquests costos estan contemplats en part gràcies a la segmentació del codi que facilita la modificació d'aquest. El principal risc que podria afectar la viabilitat del projecte seria una mala acceptació per part de l'usuari final, no fent ús de l'aplicació, o la decisió de no actualitzar el programari en funció de les noves versions de sistemes operatius o tipologies d'arxius.

## 7.2. Dimensió Mediambiental

Considerant la naturalesa del projecte, que és de petita escala, no es preveu que generi un impacte ambiental significatiu. Pel que fa a maquinari utilitzat, cap dels components ha sigut comprat específicament pel projecte, han sigut reutilitzats. Tot i que el consum d'energia no és elevat, aquest sí que suposa un impacte negatiu sobre el medi ambient pel fet que l'energia consumida no prové d'energies renovables.

En el plantejament inicial del projecte s'han valorat aspectes com els recursos humans i materials juntament amb la despesa energètica per estimar l'impacte ambiental. Si es tornés a realitzar el projecte, es creu que es podria completar amb menys recursos, mantenint el mateix enfocament en minimitzar l'impacte ambiental mitjançant la reutilització de recursos existents i l'optimització dels processos.

Actualment, el problema es resol mitjançant el programari de pagament CardRecon. El projecte no implicarà cap millora mediambiental respecte a l'opció disponible en el mercat. S'ha tingut en compte l'impacte mediambiental de la realització del projecte fins a cert punt i en la secció 6.1 s'han estimat els recursos necessaris per a la realització del projecte juntament amb el seu impacte. No obstant això, no s'ha quantificat ni certificat formalment aquesta reducció.

Dins del projecte no existeixen possibles riscos que puguin fer augmentar l'empremta ecològica.

## 7.3. Dimensió Social

En primer lloc, és essencial ressaltar els coneixements, l'experiència i les habilitats que aquest projecte fomentarà i em permetrà adquirir. És important assenyalar que aquest és el meu primer projecte d'aquesta envergadura que desenvolupo de manera independent, el qual suposarà un desafiament per a les meves capacitats i alhora m'aportarà una visió professional sobre la gestió, documentació i desenvolupament de projectes tècnics a escala professional.

Avui en dia, les transaccions amb targetes de crèdit representen una part significativa de les transaccions mundials, generant un risc d'exposició de les dades de pagament, també conegudes com a CHD. És per això que és important garantir la seguretat i protecció d'aquestes dades. Pel que fa a l'objectiu principal d'aquest projecte, no té un impacte directe sobre la societat, ja que no repercuteix directament en la població. No obstant això, la seguretat de les dades de les targetes té un impacte indirecte, ja que contribueix a generar confiança en la societat respecte a les transaccions electròniques.

El projecte no aborda una necessitat real perquè el programari existent ja compleix les necessitats de la problemàtica que s'aborda. Tot i això, el cost del programari existent és elevat i poc accessible per algunes empreses. Com ja s'ha explicat a la secció 1.4, els principals beneficiats serien l'empresa Isec Auditors i les diferents empreses que fessin ús de l'aplicació. Per contra, l'empresa responsable de CardRecon seria la principal perjudicada del projecte en la mesura que perdria volum de clients.

Aquest projecte m'ha fet plantejar l'ètica dels professionals que treballen en aquest sector. S'ha de tenir en compte que per aquest tipus de projectes es treballa amb dades delicades que tenen un gran valor mercantil.

L'únic escenari amb afectació real sobre la població seria en el cas que algun membre de l'empresa amb accés a l'aplicació decidís fer-ne un ús inapropiat robant les dades de targeta trobades. El projecte automatitza una feina que s'hauria de fer manualment, però en cap moment generaria una situació de dependència.

## 8. Desenvolupament Tècnic

En la pròxima secció, es farà una explicació detallada dels diferents passos seguits per poder desenvolupar completament la part pràctica del projecte. Els diferents subapartats que es tractaran estan relacionats amb les diferents tasques que es defineixen en el punt 5.1.3 i 5.1.4. Aquests s'aniran mencionant en funció del seu procés de desenvolupament.

### 8.1. Disseny

Després de la fase de recopilació, es van realitzar un seguit d'esbossos que es van presentar als responsables per la seva aprovació. Aquests esbossos van ser realitzats directament amb l'eina que utilitzarem per dissenyar el projecte, el Qt Designer 5 [10]. Dels esbossos inicials, es va generar un primer disseny que inclou dues pestanyes. La primera pestanya és per seleccionar targetes i veure les opcions disponibles. La segona pestanya té una barra de progrés i dues seccions on es mostra la quantitat de troballes per a cada targeta, juntament amb les característiques de les troballes.

Durant el desenvolupament del projecte, s'han experimentat diverses situacions que han requerit modificacions sobre el disseny inicial. Aquestes adaptacions han estat motivades per les sol·licituds del director del projecte i per les problemàtiques o necessitats sorgides durant el desenvolupament del codi.

Les modificacions realitzades han tingut diversos impactes en el disseny de l'aplicació. En alguns casos, s'han creat noves pantalles per abordar noves funcionalitats o requisits. En altres casos, les modificacions han implicat modificacions o l'addició de nous elements a les pantalles ja existents per millorar la usabilitat o respondre a les necessitats emergents dels usuaris.

Després de realitzar el disseny inicial i aplicar les diferents modificacions pertinents s'ha generat un disseny final que s'ha simplificat en el diagrama que es mostra a continuació, en la figura 4.

En el diagrama es pot veure la relació entre les diferents pantalles, frames i arxius que té el projecte dels que es parlarà més endavant amb més profunditat. La primera pantalla que es mostra és la de Login, que juntament amb un fitxer extern que conté la base de dades, realitza el procés d'autenticació. Un cop fet el login, es mostra la pantalla principal, al mateix temps que es configura l'aplicació gràcies a l'arxiu de configuració que apareix en taronja. Per acabar, es mostra la pantalla secundària, que en funció de l'opció triada, seguirà un camí o un altre per ser mostrada. En el cas que s'exportin els resultats en format PDF, s'utilitzarà un fitxer extern tipus script per poder realitzar el procés correctament.

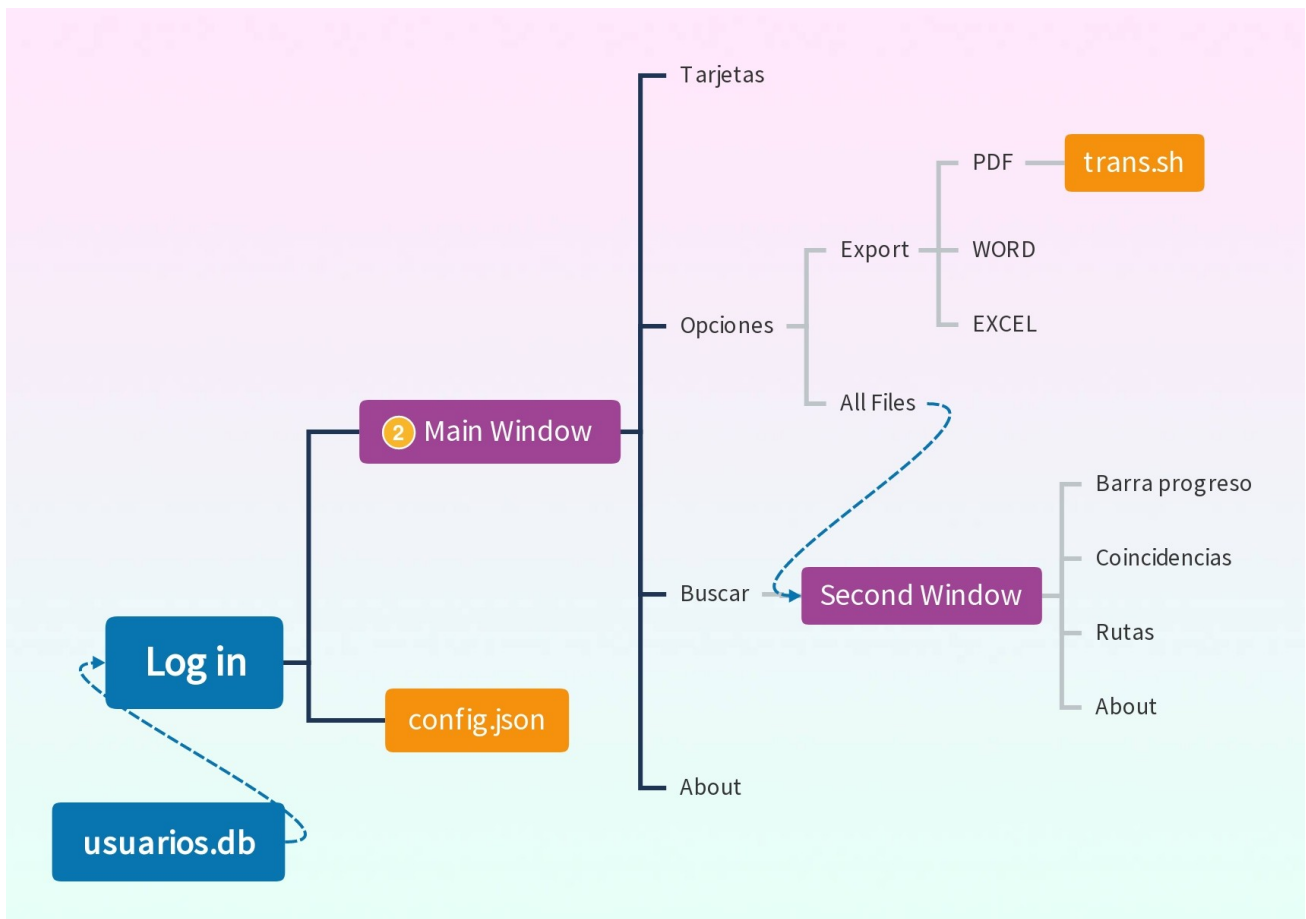


Figura 4. Diagrama de flux de l'aplicació  
 Font: <https://gitmind.com/app/docs>

Pel procés de disseny, s'ha optat per fer servir l'eina de programari anomenada Qt Designer 5, ja que ofereix una interfície gràfica molt fàcil de fer ús i, al mateix temps, proporciona una àmplia gamma d'opcions i recursos que ens permeten dissenyar i modificar interfícies gràfiques en Python de manera eficient.

### 8.1.1. Pantalla Login

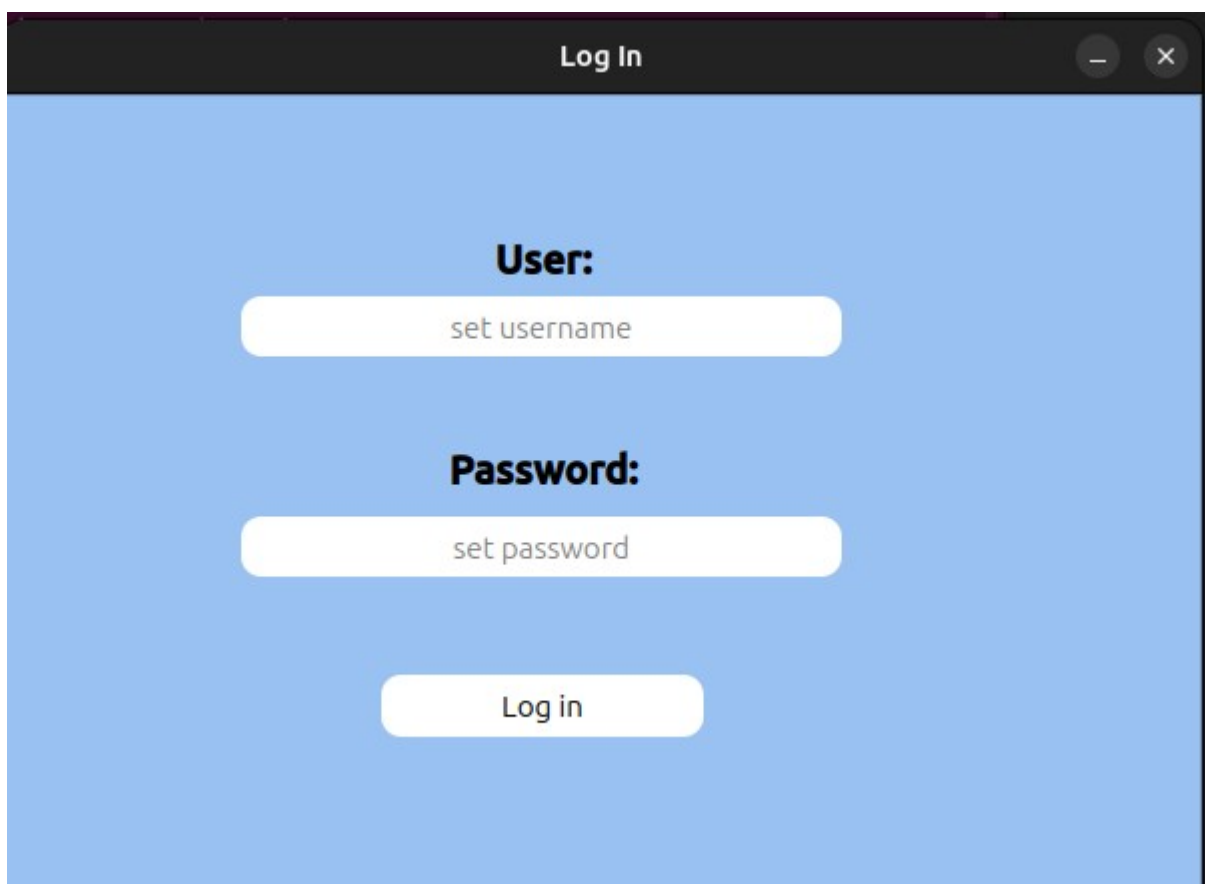
Durant la fase de desenvolupament del projecte, es va proposar la creació d'una pantalla inicial d'inici de sessió destinada tant als clients com als empleats de l'empresa. Aquesta pantalla té com a objectiu proporcionar un punt de confidencialitat i permetre una millor gestió de la distribució i ús de l'aplicació, restringint l'accés a usuaris específics. Aquesta pantalla de login apareix en iniciar l'aplicació i consta d'un frame principal amb tres elements, tal com es mostra a la figura 5, coneguda d'ara endavant com a "login frame". Aquest frame s'ha configurat amb una mida estàtica de 600x400 píxels.

El primer element dins del frame principal és un QVBoxLayout. Hem optat per aquesta disposició perquè permet alinear els elements verticalment. En aquest cas, els elements són un QLabel i un QLineEdit. El QLabel té el text "Usuari: ", indicant què s'espera que l'usuari introdueixi en el QLineEdit situat a continuació. Aquest QLineEdit recull el nom d'usuari necessari per accedir a l'aplicació.

Seguidament, trobem un segon QVBoxLayout amb els mateixos elements, un QLabel i un QLineEdit. En aquest cas, el QLabel té el text "Contrasenya:", indicant que s'espera que l'usuari introdueixi la seva contrasenya en el QLineEdit situat a continuació. Aquest QLineEdit està configurat per amagar el text introduït, assegurant així la confidencialitat de la contrasenya.

A la part inferior de la pantalla, hi ha un QPushButton amb el text "Inicia sessió". Aquest botó permet enviar una petició d'accés un cop tots els camps estan omplerts. En prémer el QPushButton, es genera un disparador que valida les credencials de l'usuari. Aquest enfocament s'ha adoptat per simplificar i assegurar el procés d'inici de sessió.

A més, el sistema d'inici de sessió està integrat amb un fitxer extern que conté la base de dades dels usuaris, assegurant que només els usuaris autoritzats puguin accedir a l'aplicació.



*Figura 5. Pantalla log in  
Font. Elaboració pròpia*

## 8.1.2. Pantalla principal

La pantalla principal de l'aplicació, està composta per un frame principal, que a partir d'ara anomenarem «main frame», que conté en el seu interior quatre frames més petits, com es mostra a la Figura 6.

El primer frame està situat a la part superior del main frame i s'ha anomenat «sct\_sup» Hem decidit anomenar-lo així per poder fer-ne referència al codi Python més endavant. Aquest subframe inclou dos elements: un botó d'eines (QToolButton) a la dreta i una etiqueta (QLabel) a l'esquerra. El QToolButton s'ha triat perquè permet mostrar finestres emergents (pop-ups) quan es fa clic, i s'ha personalitzat la seva icona per fer més evident la seva funcionalitat. La QLabel s'utilitza per mostrar el nom de l'aplicació.

Per continuar, ens trobem amb el segon frame, anomenat "sct\_tarjetas". Aquest frame està format per set elements: una etiqueta (QLabel) i sis QCheckBox, que ens ajuden a organitzar els elements d'aquest frame de manera efectiva. Aquest frame està dissenyat per mostrar les diferents targetes amb les quals es podrà treballar.

En primer lloc, tenim l'etiqueta, que s'utilitza per afegir un quadre de text dins de l'aplicació. En aquest cas, s'utilitza per indicar el propòsit d'aquest frame: seleccionar les targetes que es volen trobar.

Seguidament, trobem les sis targetes dividides en dos grups de tres QCheckBox. Aquestes caselles de selecció es poden marcar o desmarcar per seleccionar quin tipus de targeta es vol buscar en el procés d'anàlisi. Cada QCheckBox s'ha personalitzat amb un valor inicial i una imatge per defecte, de manera que apareixen sense marcar quan s'inicia l'aplicació. Cada QCheckBox correspon a un tipus de targeta, i s'ha seleccionat una imatge específica per modificar el logotip predeterminat.

Les sis targetes disponibles són de les principals emissores de targetes a escala mundial: VISA, MASTERCARD, AMERICAN EXPRESS i JCB (fundadores de PCI DSS), UNION PAY (principal emissor a la Xina i un dels més grans a escala mundial), i DISCOVER (una de les principals emissores a la regió dels Estats Units i Centreamèrica). [11]

El tercer subframe, anomenat sct\_opciones, conté les dues opcions disponibles a l'aplicació, representades per un QCheckBox cadascuna. Les QCheckBox inicialment no estan marcades i s'han modificat amb una etiqueta identificativa en el text.

Finalment, el subframe situat a la part inferior, anomenat «sct\_buscar», està compost per un únic element, un QPushButton anomenat «boton\_buscar», al qual se li ha modificat el valor "Text" per afegir-hi una etiqueta identificativa.

Si es marca l'opció export en el subframe sct\_opciones i es prem el botó boton\_buscar, es mostra una finestra emergent amb tres opcions per escollir: exportar en format PDF, EXCEL o WORD. Aquestes opcions apareixen sense marcar inicialment, permetent a l'usuari seleccionar segons les seves necessitats.

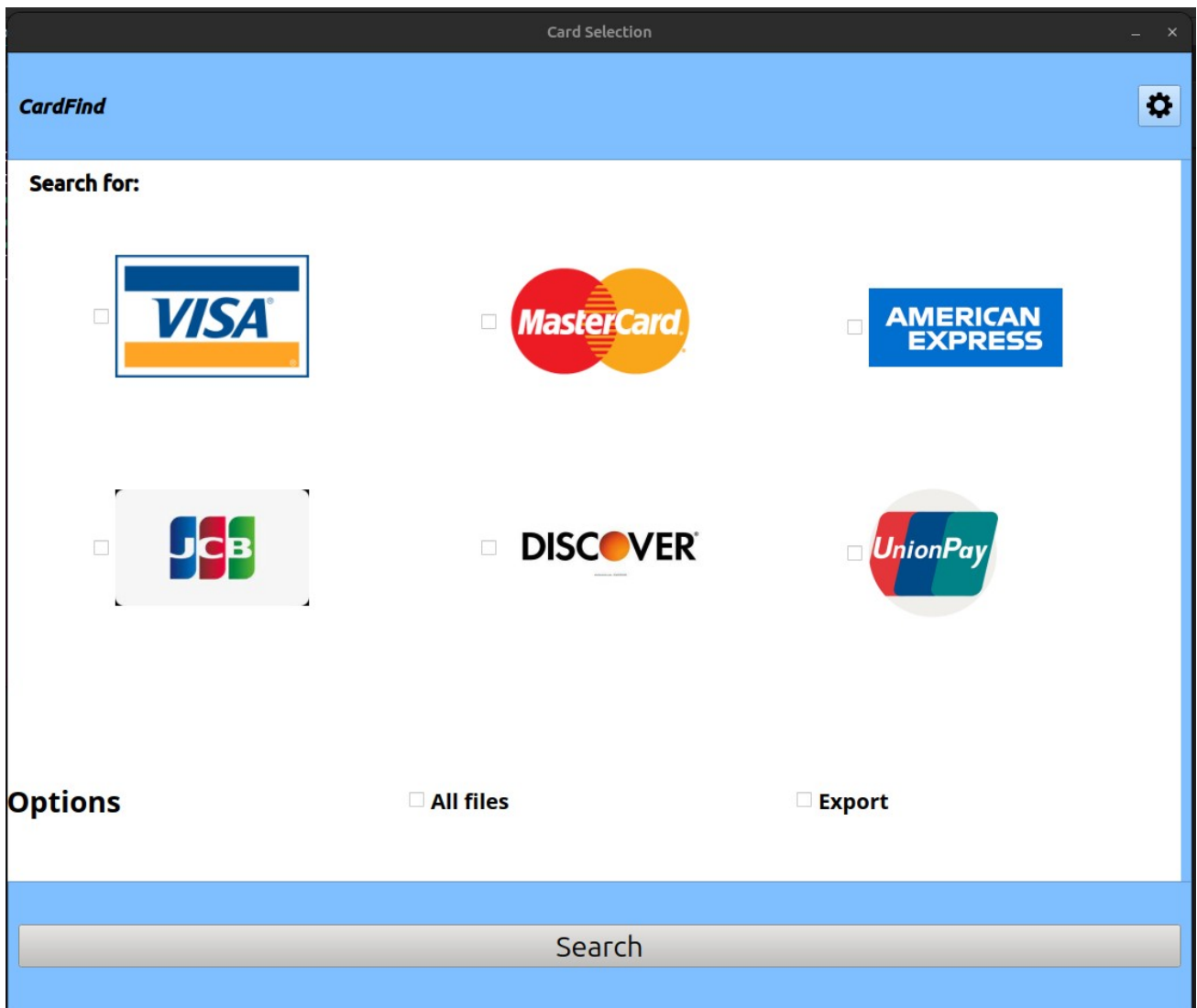


Figura 6. Pantalla principal  
Font. Elaboració pròpia

### 8.1.3. Pantalla Secundària

Aquesta segona pantalla, que a partir d'ara anomenarem "second frame", es mostra després d'iniciar la recerca de les targetes. Està formada per un frame principal igual que la pantalla principal i quatre frames més petits, com es pot observar a la figura 7.

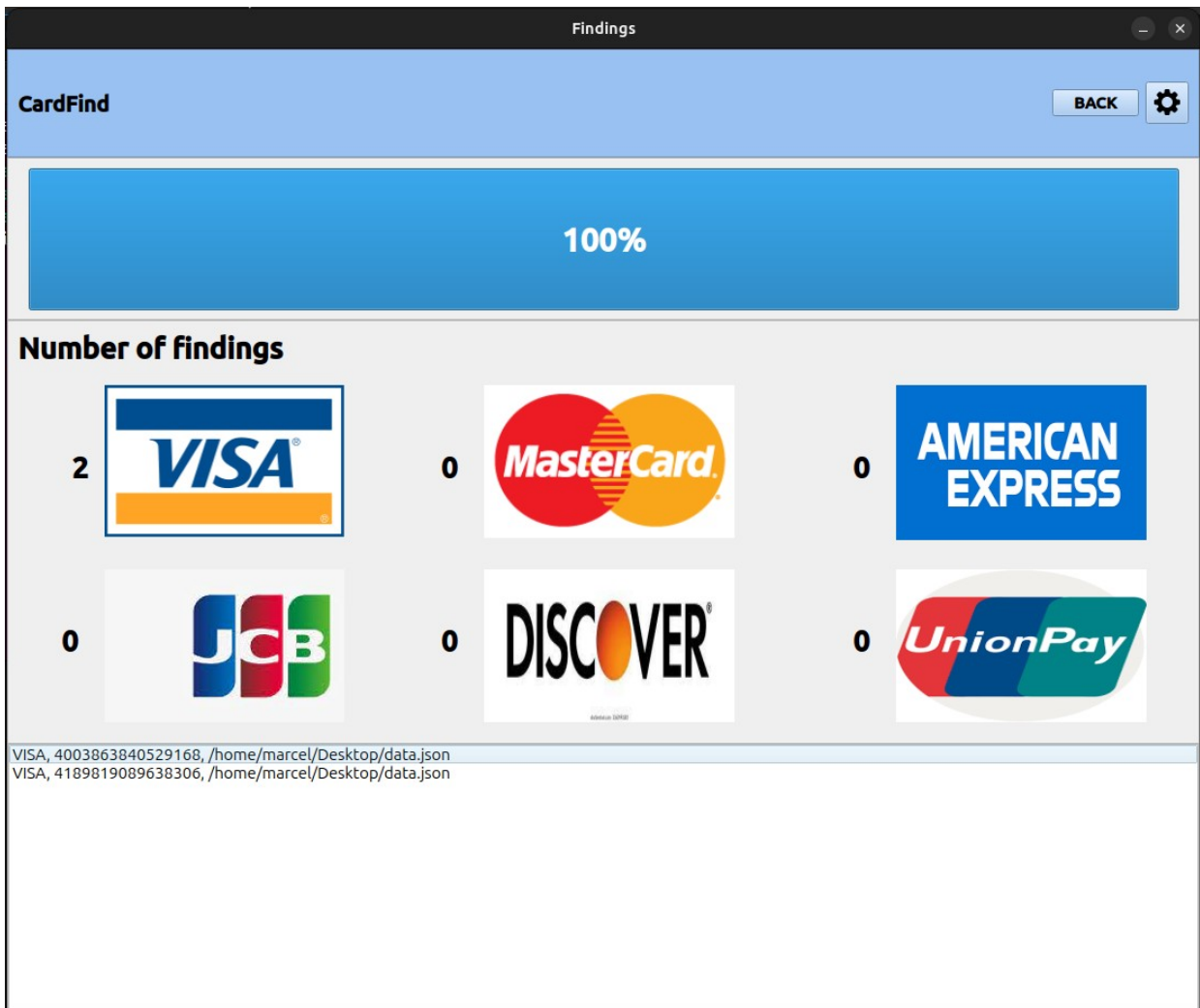
El primer frame, situat a la part superior del "second frame", és anomenat "frame\_tool". És semblant al mencionat en la pantalla principal i està compost per tres elements: un QLabel a la part esquerra i un QToolButton juntament amb un QPushButton a la part dreta. S'ha triat aquest tipus d'element per les seves característiques i per la necessitat de mostrar una finestra emergent amb la versió de l'aplicació. Hem canviat la icona per donar-li una aparença més actualitzada i fer més evident la seva funcionalitat. El QPushButton té la funció de tornar a la pàgina inicial de l'aplicació per poder realitzar de nou el procés d'anàlisi en cas que fes falta. El QLabel s'utilitza per escriure text a la interfície gràfica i mostra el nom de l'aplicació.

Seguidament, trobem el segon frame anomenat "frame\_progreso", que conté un únic element: un QProgressBar. La seva finalitat és mostrar el progrés de la cerca dins del sistema, dividit per carpetes. A mesura que s'inspeccionen els directoris i les seves carpetes, la barra de progrés s'actualitza amb un valor percentual.

En tercer lloc, tenim el frame "frame\_tarjetas", que inclou set elements. A la part superior hi ha un QLabel amb el text "Number of findings" per definir el frame visualment i facilitar la comprensió per part de l'usuari. Seguidament, hi ha sis QHBoxLayout, els quals agrupen dos elements: un comptador i una imatge. Aquests QHBoxLayout s'utilitzen per alinear els elements. Els comptadors, implementats amb QLabel, tenen un valor inicial de "0" i s'actualitzen dinàmicament amb el nombre de troballes. Les imatges, també QLabel, mostren el logotip de la targeta corresponent a cada troballa.

Finalment, trobem el frame "frame\_archivos" situat a la part inferior, compost per un únic element: un QListWidget. Aquest frame mostra tots els números de targeta trobats i les seves ubicacions dins de la màquina revisada. Hem utilitzat aquest element per a llistar clarament les troballes i, en cas de moltes, permetre filtrar entre els resultats.

Un cop acabat tot el procés, i en el cas que s'hagi decidit exportar el resultat, es mostra una finestra emergent personalitzada per cada fitxer que es genera, indicant que s'ha guardat el document correctament.



*Figura 7. Pantalla secundària*  
*Font. Elaboració pròpia*

## 8.2. Desenvolupament inicial

### 8.2.1. Configuració de botons

Un cop finalitzat el disseny de les diferents pantalles de l'aplicació, s'ha procedit a transformar el disseny generat mitjançant QtDesigner a codi en Python. Primerament, s'ha creat un nou fitxer amb el nom de l'aplicació i l'extensió ".py". En aquest fitxer s'han definit les llibreries necessàries per reconèixer els dissenys i s'han afegit algunes línies de codi per carregar l'aplicació, invocant la pantalla principal una vegada s'executa a la terminal, tal com es pot veure a la figura 8.

```

if __name__ == "__main__":
    app = QApplication(sys.argv)

    window = MainWindow()
    window.show()
    app.exec_()

```

*Figura 8. Codi d'execució de l'aplicació*  
*Font. Elaboració pròpia*

Després de diverses reunions amb el responsable del projecte, s'han establert diverses característiques addicionals que l'aplicació havia de contemplar. Una d'aquestes funcionalitats destacades és la capacitat d'iniciar sessió mitjançant usuaris nominatius. Aquesta característica és essencial perquè l'aplicació, una vegada desenvolupada completament, es desplegarà per a diversos clients, els quals accediran mitjançant credencials d'accés proporcionades per l'empresa Isec Auditors. Aquest sistema de control d'accés garantirà la seguretat i la traçabilitat de les operacions realitzades per cada usuari, assegurant que només el personal autoritzat pugui accedir a les funcionalitats de l'aplicació.

Per habilitar aquest procés, s'ha creat una nova pantalla, com s'ha esmentat a la secció 8.1 del disseny final de l'aplicació. A més, s'han realitzat diverses modificacions al codi ja existent per tal que aquesta nova pantalla es mostri quan s'inicia l'aplicació, tal com es pot veure a la figura 9. En aquesta modificació, s'ha canviat la pantalla inicial que es carrega perquè ara mostri la "LoginDialog", coneguda en el disseny com la pantalla de login.

```

if __name__ == "__main__":
    app = QApplication(sys.argv)

    login_dialog = LoginDialog()
    login_dialog.show()

    app.exec_()

```

*Figura 9. Modificació del codi d'execució de l'aplicació*  
*Font. Elaboració pròpia*

Per assegurar que aquesta modificació del codi s'executi correctament, s'ha creat una nova classe anomenada LoginDialog, amb la seva inicialització corresponent. En aquesta classe, s'ha carregat l'arxiu de disseny "login.ui" a la variable "ui", s'ha configurat la posició a la pantalla i s'ha establert una connexió amb la base de dades "usuarios.db". Aquesta base de dades ha estat creada per emmagatzemar els diferents usuaris i contrasenyes mitjançant l'ús de la llibreria de base de dades "sqlite3".

L'última línia de codi, que es pot observar a la figura 10, abans de la funció "login", s'executarà cada vegada que es premi el botó de login. Aquesta línia crida a la funció "login", passant com a paràmetres el valor introduït al camp d'usuari i el valor introduït al camp de contrasenya.

```

class LoginDialog(QMainWindow):
    def __init__(self):
        super().__init__()
        ul = loadUi('login.ui', self)

        self.setGeometry(650, 350, 600, 400)

        # Data base connection
        self.conn = sqlite3.connect('usuarios.db')
        self.cursor = self.conn.cursor()

        # Login button
        ul.boton_login.clicked.connect(lambda: self.login(ul.user_text.text(), ul.passwords_text.text()))

    def login(self, usuario, contrasena):

        hashed_contrasena = hashlib.sha256(contrasena.encode('utf-8')).hexdigest()

        # Data base query
        self.cursor.execute("SELECT * FROM usuarios WHERE usuario=? AND contrasena=?", (usuario, hashed_contrasena))
        usuario_encontrado = self.cursor.fetchone()

        if usuario_encontrado:
            # Mostrar la ventana principal
            self.hide()
            main_window = MainWindow()
            main_window.show()
        else:
            # Show error message
            QMessageBox.warning(self, "Log in", "User or Password incorrect")

```

*Figura 10. Codi de la classe LoginDialog i la seva funció Login*  
*Font. Elaboració pròpia*

Dins de la funció "login", com es mostra a la figura situada en la part superior, s'ha creat una consulta SQL per accedir a la base de dades. En aquesta consulta, es busca l'usuari i la contrasenya introduïts com a valors de la per la comanda SELECT, i el resultat s'ha guardat a la variable "usuario\_encontrado".

En cas que el resultat de la consulta sigui incorrecte, es mostra un missatge d'error indicant que l'usuari o la contrasenya són incorrectes. Per contra, si el resultat és correcte, s'amaga la pantalla actual i es carrega i mostra la classe «MainWindow», que s'havia utilitzat fins aleshores per a l'ús de l'aplicació.

Seguidament, es carrega la pantalla principal en el codi juntament amb tots els seus components mitjançant una variable anomenada "ui". Això permet accedir a tots els recursos i funcionalitats d'aquesta pantalla.

Per comprovar el correcte funcionament, es van definir diversos usuaris amb les seves corresponents contrasenyes per poder provar d'accedir amb diferents usuaris. En conseqüència, també es comprovava que es realitzés el canvi de la pantalla de login a la principal, mostrant tot el seu contingut.

Després, es procedeix a carregar la pantalla secundària i els seus components de la mateixa manera que s'ha fet amb la pantalla principal. En aquest cas, es carrega el disseny en la variable "second\_ui" mitjançant la crida "findChild", indicant el component que es vol recollir, el tipus d'element que és i el seu nom dins del disseny en Qt Designer, com es pot observar en les figures 11 i 12.

```

class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        ui = loadUi('main.ui', self)

        self.setGeometry(450, 150, 1018, 818)

        # Load second.ui
        self.change_window()

        ruta_inicial = self.configuracion()

```

Figura 11. Inicialització de les diferents pantalles

Font. Elaboració pròpia

```

def change_window(self):
    self.second_ui = uic.loadUi("second.ui")
    self.progressBar = self.second_ui.findChild(QtWidgets.QProgressBar, "progressBar")
    self.contador1 = self.second_ui.findChild(QtWidgets.QLabel, "contador1")
    self.contador2 = self.second_ui.findChild(QtWidgets.QLabel, "contador2")
    self.contador3 = self.second_ui.findChild(QtWidgets.QLabel, "contador3")
    self.contador4 = self.second_ui.findChild(QtWidgets.QLabel, "contador4")
    self.contador5 = self.second_ui.findChild(QtWidgets.QLabel, "contador5")
    self.contador6 = self.second_ui.findChild(QtWidgets.QLabel, "contador6")
    self.listatarjetas = self.second_ui.findChild(QtWidgets.QListWidget, "listatarjetas")
    self.second_ui.toolButton.clicked.connect(lambda: self.action_about())
    self.second_ui.backbutton.clicked.connect(lambda: self.action_back())

```

Figura 12. Carrega pantalla secundària

Font. Elaboració pròpia

Durant aquesta primera fase de desenvolupament, s'ha fet una selecció de les diferents pantalles mencionades en l'apartat de disseny i els seus elements, dividint-los en dos grups principals: els elements que interactuen dins de la mateixa pantalla i els elements que interactuen entre pantalles.

En el primer grup es troben els elements relacionats amb la selecció de targetes, les opcions i la secció «About us». En aquest cas, els elements de tipus `QCheckBox` ja tenen un valor per defecte establert i només cal declarar-los dins del codi. Aquest valor pot ser llegit directament del `QCheckBox` mitjançant la línia de codi `ui.element.isChecked()`. Aquest valor es modifica automàticament segons l'acció de l'usuari de seleccionar o desseleccionar la casella corresponent, tant en el cas de les targetes com en el de les opcions, com es pot apreciar en la figura 13.

```

ui.boton_tool.clicked.connect(lambda: self.action_about())

ui.boton_buscar.clicked.connect(lambda: self.buscar_coincidencias(ui.tarjeta1.isChecked(), ui.tarjeta2.isChecked(), ui.tarjeta3.isChecked(), ui.tarjeta4.isChecked(), ui.tarjeta5.isChecked()

```

Figura 13. Declaració de `QtoolButton` i `QCheckBox`

Font. Elaboració pròpia

Dins del primer apartat de l'annex podem trobar explicada amb tot de detall la funció about us juntament amb les diferents crides i accions que aquesta realitza.

En el segon grup, trobem l'element que interactua amb les dues pantalles, en aquest cas el botó "boton\_buscar". Quan aquest botó es prem, modifica la vista de l'aplicació, passant de la pantalla principal, que està emmagatzemada a la variable "ui", a la pantalla secundària, que es troba guardada a la variable "second\_ui". Aquesta acció es realitza cridant la funció "on\_pushButton\_buscar\_clicked()". Dins d'aquesta funció, es duu a terme el canvi de pantalla esmentat, a més de reubicar la pantalla al centre, tal com es mostra a la figura 14.

```
def on_pushButton_buscar_clicked(self):  
    # Close the current window  
    self.close()  
  
    # show the second window  
    self.second_ui.show()  
  
    self.second_ui.setGeometry(450, 150, 1018, 818)
```

Figura 14. Declaració funció on\_pushButton\_buscar\_clicked()

*Font.* Elaboració pròpia

Un cop implementades les declaracions dels botons de les sis targetes i les dues opcions, s'ha procedit a provar el funcionament de l'aplicació segons les indicacions de la tasca DP6 per assegurar-nos que l'aplicació llegeix correctament els valors seleccionats per l'usuari. També s'han duut a terme proves per confirmar que el botó "About us" mostra el missatge de manera adequada i es pot tancar correctament.

A més, s'ha verificat que en prémer el botó "boton\_buscar", es canviï la vista de la pantalla principal a la pantalla secundària. Aquesta comprovació no només implica verificar que es mostri correctament la nova pantalla, sinó també assegurar-se que els botons que es troben en aquesta nova pantalla funcionin correctament. En particular, s'ha confirmat que el botó "About us" mostra el missatge de la mateixa manera que a la pantalla principal.

També s'ha revisat el funcionament dels tres elements presents a la pantalla secundària. Mitjançant un seguit de proves, s'ha comprovat que el valor de l'element QProgressBar es modifica correctament amb la línia de codi "self.Qprogressbar.setValue(valor)", on la variable "valor" s'ajusta per reflectir el progrés en percentatge a la barra. Quant als QLabels que actuen com a comptadors del nombre d'evidències trobades de cada tipus de targeta, s'ha verificat que el seu valor es modifiqui correctament amb la línia de codi "self.contador\_nº.setText(valor)".

Finalment, s'ha confirmat que es mostra el text a l'element QListWidget. Per fer-ho, s'ha creat una variable amb text desitjat i una variable QListWidgetItem on s'ha carregat aquest text mitjançant la funció "setText". Un cop carregat el text a la variable QListWidgetItem, aquesta s'ha afegit a la QListWidget i s'ha verificat que el text mostrat coincideixi amb el text esperat.

## 8.2.2. Funcions bàsiques

Un cop configurats tots els elements implementats en el disseny inicial, es procedeix a implementar un seguit de funcions que seran bàsiques en el desenvolupament del projecte. La primera d'aquestes funcions és la d'obrir un arxiu, anomenada "action\_open". La seva funció principal és obrir un arxiu dins del sistema operatiu per al seu posterior ús, que en el nostre cas serà la lectura i anàlisi del seu contingut.

Inicialment, aquesta funció només obre un arxiu, una de les opcions d'anàlisi. Més endavant, es tractarà el cas de voler obrir tots els arxius del sistema. La funció rep com a paràmetre la ruta inicial on començar a buscar l'arxiu a obrir, aquest element que rep per paràmetre s'anomena "ruta\_inicial".

Dins de la funció, es defineix un nou element, un QFileDialog, que genera una finestra emergent on es poden observar tots els arxius del sistema, com es pot veure a la figura 15.

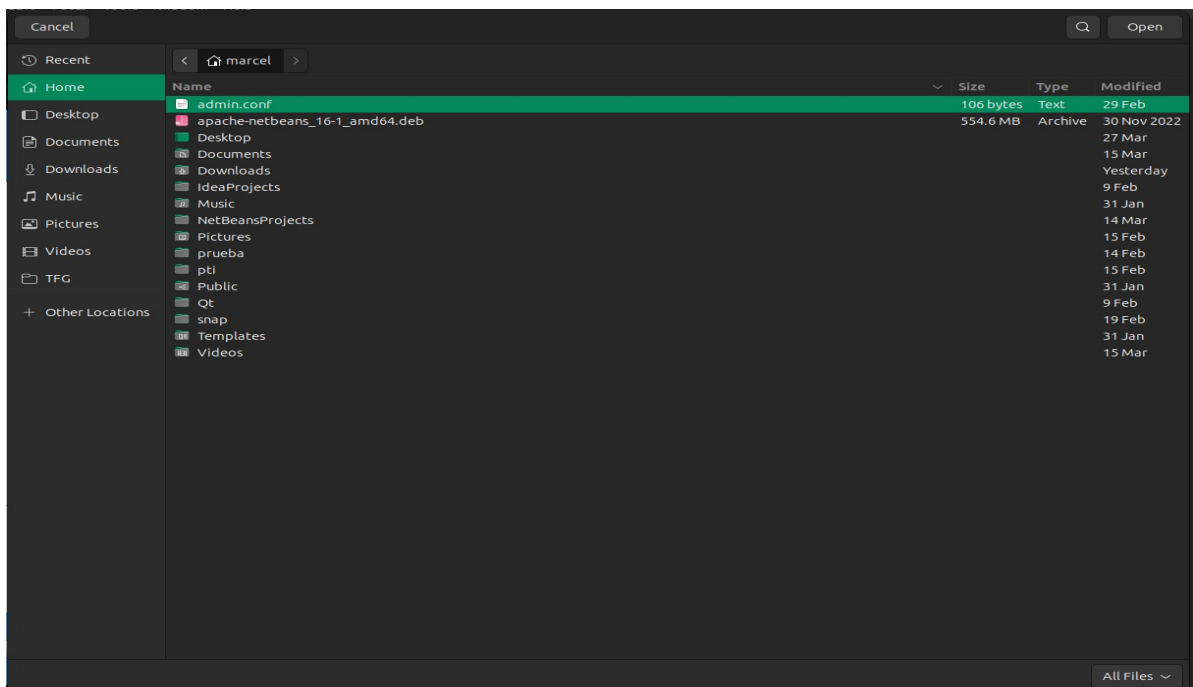


Figura 15. Pestanya emergent de QFileDialog

Font. Elaboració pròpia

S'afegeix un títol al QFileDialog utilitzant l'opció "setWindowTitle" i s'apliquen uns filtres per mostrar resultats a la finestra emergent amb l'opció "setNameFilters". Finalment, com s'ha esmentat al començament, es defineix un directori inicial utilitzant la variable "ruta\_inicial". Un cop tot configurat, es mostra la finestra emergent amb la configuració esmentada anteriorment i, amb l'ajuda de la funció "getOpenFileName()", s'obté la ruta de l'arxiu sobre el qual es treballarà. A continuació, es mostra el codi utilitzat a la figura 16.

```

def action_open(self, ruta_inicial):
    # Open a file explorer dialog
    file_dialog = QFileDialog(self)
    file_dialog.setWindowTitle("Select a file")
    file_dialog.setNameFilters(["Text files (*.txt)", "All files (*)"])
    file_dialog.setDirectory(ruta_inicial)
    file_path, _ = file_dialog.getOpenFileName()

```

Figura 16. Codi de la funció `action_open`  
 Font. Elaboració pròpia

De la mateixa manera que s'ha pensat i creat aquesta funció, s'ha desenvolupat una funció molt similar «`action_save_as`» per guardar elements que podem trobar explicada amb detall al segon punt de l'annex juntament amb la funció que permet mostrar per pantalla els errors que es generen al codi.

Una de les principals funcions definides inicialment és la funció "configuració". Aquesta funció té una rellevància extra dins del projecte, ja que, com s'ha comentat a l'inici del projecte, aquest és multiplataforma, i aquesta funció s'encarrega de detectar el sistema operatiu sobre el qual es treballa i definir la configuració de l'aplicació i la ruta base sobre la qual treballarà l'aplicació. Aquesta ruta base és la variable "ruta\_inicial" mencionada en les funcions "action\_open" i "action\_save\_as". Per poder realitzar aquesta configuració de l'aplicació, es carrega a una nova variable creada anomenada "config" definida en un arxiu extern de configuració anomenat "config.json" que conté les configuracions tant per a Windows com per a Linux, com es pot apreciar a la figura 17.

```

{
  "directorios": {
    "Windows": {
      "inicio": "%USERPROFILE%\"
    },
    "Linux": {
      "inicio": "/home/marcel"
    }
  }
}

```

Figura 17. Arxiu de configuració  
 Font. Elaboració pròpia

Per realitzar la comprovació de la correcta identificació del sistema operatiu i, per tant, la ruta que es farà servir s'ha creat un petit codi en el qual es carrega a l'arxiu de configuració en una variable i posteriorment, mitjançant una condició, comprova si el valor és «Linux» o «Windows» i es mostra per terminal la ruta que cada cas té definida.

Un cop s'ha carregat l'arxiu de configuració a la variable, aquesta és cridada per la variable "ruta\_directorio" per tal d'obtenir la ruta base de treball de l'aplicació, que al final de la funció

es retornarà. En aquesta crida, com es pot observar a la figura 18, es busca dins del vector "directoris" si algun dels sistemes mencionats coincideix amb el sistema detectat per la línia de codi "platform.system()". En cas que coincideixi, se selecciona la ruta especificada a la secció "inicio" de l'arxiu de configuració.

```
def configuracion(self):
    with open("config.json", "r") as f:
        config = json.load(f)

    ruta_directorio = config["directorios"][platform.system()]["inicio"]

    return ruta_directorio
```

Figura 18. Codi de la funció Configuracion

Font. Elaboració pròpia

L'última funció definida dins de les funcions bàsiques és la funció "select\_export". Aquesta funció parteix de la implementació inicial del botó d'opcions "Export" i el funcionament definit en les funcions inicials. De la mateixa manera que es realitza el canvi de pantalla quan es polsa el botó "Search", també s'invoca a aquesta funció que mostra una finestra emergent en la qual s'indiquen les diferents opcions d'exportació.

Inicialment, aquesta funció crea un element QDialog amb títol "Select Export Options" i se li defineix la mida i la disposició, vertical, en un QVBoxLayout. Després, s'afegeixen les diferents opcions que es podran seleccionar, en aquest cas tres (PDF, Word i Excel), seguint el mateix procediment. Primer es crea un element tipus QCheckBox i se li assigna un text explicatiu, és a dir, s'indica l'opció que s'estarà triant. Al mateix temps es defineix l'estat en el qual es troben els QCheckBox, no marcats, i finalment, s'afegeixen aquests elements al layout creat al començament de la funció, com es pot observar a la figura 19.

En últim lloc, just abans d'executar la finestra emergent amb la funció d'executar, es defineix un nou layout horitzontal amb l'element QHBoxLayout i en el seu interior es defineixen dos botons del tipus QPushButton. El primer botó rep el títol d' "Accept" i la seva funcionalitat és tancar la finestra emergent i saltar fins al final de la funció. El segon botó rep el títol de "Cancel"; en aquest cas, es tanca la finestra emergent sense acabar tot el codi de la funció i, per tant, no retornant res. L'última línia de la funció, en el cas que s'hagi premut el botó d'acceptar, retorna l'estat de cadascun dels QCheckBox definits durant la funció per poder ser tractats posteriorment.

```

def select_export(self):
    """ ...

    # Create a dialog window
    dialogo = QDialog(self)
    dialogo.setWindowTitle("Select Export Options")

    # Set initial minimum size of the window
    dialogo.setMinimumSize(QtCore.QSize(500, 150))

    # Create a main layout
    layout_principal = QVBoxLayout()
    dialogo.setLayout(layout_principal)

    # Create a checkbox for PDF
    self.check_pdf = QCheckBox("Exportar to PDF")
    self.check_pdf.setChecked(False)
    layout_principal.addWidget(self.check_pdf)

    # Create a checkbox for Excel
    self.check_excel = QCheckBox("Exportar to Excel")
    self.check_excel.setChecked(False)
    layout_principal.addWidget(self.check_excel)

    # Create a checkbox for Word
    self.check_word = QCheckBox("Exportar to Word")
    self.check_word.setChecked(False)
    layout_principal.addWidget(self.check_word)

    # Create buttons layout
    layout_botones = QHBoxLayout()
    layout_principal.addLayout(layout_botones)

    boton_aceptar = QPushButton("Accept")
    boton_aceptar.clicked.connect(dialogo.close)
    layout_botones.addWidget(boton_aceptar)

    boton_cancelar = QPushButton("Cancel")
    boton_cancelar.clicked.connect(dialogo.reject)
    layout_botones.addWidget(boton_cancelar)

    # Show the dialog window
    dialogo.exec_()

    return self.check_pdf.isChecked(), self.check_excel.isChecked(), self.check_word.isChecked()

```

*Figura 19. Codi de la funció select\_export  
Font. Elaboració pròpia*

De la mateixa manera que s'havia fet amb els diferents botons, un cop implementades totes les funcions, s'han anat testejant una per una per comprovar el seu correcte funcionament. Durant aquest procés, s'han detectat i solucionat petites errades al codi que podrien haver provocat resultats inesperats o errors en la seva execució. Això ha permès assegurar que cada funció operi de manera adequada i que l'aplicació funcioni de manera coherent i sense problemes detectables.

### 8.3. Desenvolupament de funcions

Després de definir les funcionalitats bàsiques de tots els botons i alguna funció per començar a navegar i provar l'aplicació, s'ha avançat cap al desenvolupament de la tasca principal: el reconeixement de dades de targeta dins d'arxius.

La primera funció implementada és "buscar\_coincidencies", que és cridada en iniciar l'aplicació quan es prem el botó "buscar". Aquesta funció rep com a paràmetres tots els estats dels diferents QCheckBox de les targetes i de les opcions. En començar l'execució de la funció, es realitza una comprovació inicial per assegurar-se que s'ha seleccionat com a mínim una targeta. Això es fa mitjançant els booleans que es reben per paràmetre de les diferents targetes, com es detalla a la figura 20.

```
def buscar_coincidencias(self, t1, t2, t3, t4, t5, t6, op1, op2, ruta_inicial):
    tarjeta = []
    if not t1 and not t2 and not t3 and not t4 and not t5 and not t6:
        self.mostrar_error("You must choose at least one card type to search for.")
    else:
```

*Figura 20. Inici del codi de la funció «buscar\_coincidencias»*

*Font. Elaboració pròpia*

La funció s'explica amb més detall en el tercer punt de l'annex. En ell s'especifiquen els diferents casos, com es tracten cadascun d'ells i quines funcions criden.

Pel que fa a les crides a funcions dins de la funció anterior, la primera crida que s'implementa en les diferents casuístiques és la funció "buscar\_directorios", com es pot observar a la figura 21. Aquesta funció, com s'ha comentat, rep com a paràmetre la ruta d'un directori a revisar.

Abans d'iniciar l'execució del codi, es declara la variable anomenada "directoris" que guardarà tots els arxius trobats. Seguidament, es comença un bucle que revisa tots els arxius dins de la ruta passada com a paràmetre mitjançant la comanda "os.listdir" i crea la ruta de l'arxiu a revisar amb la comanda "os.path.join", amb la ruta passada com a paràmetre i l'arxiu a revisar. S'han aplicat un seguit de condicions per evitar que l'aplicació col·lapsés amb algun tipus d'arxius que podem trobar en alguns sistemes, al mateix temps que es comprova si l'arxiu a tractar és de tipus arxiu o és de tipus enllaç amb la funció «is\_junction».

A continuació, es verifica si l'arxiu és un directori o un arxiu simple amb la comanda "os.path.isdir". En cas que sigui un directori, es crida recursivament la mateixa funció. Si és un arxiu, s'extreu l'extensió de l'arxiu. Si totes les condicions es compleixen, s'afegeix la ruta de l'arxiu a la llista de directoris a revisar amb la comanda "append", que posteriorment es retornarà amb el "return", com es pot apreciar a la figura 21.

Actualment, hi ha un seguit d'extensions que no es poden tractar pel fet que generen problemes a l'hora d'obrir els arxius. Tot i això, aquests elements que no es poden obrir, primer es tracten d'obrir i, en cas de no poder, s'ometen passant al següent element. Les extensions sobre les quals s'ha fet un tractament especial són aquelles que amb major probabilitat poden tenir dades de targeta, com poden ser arxius pdf, xlsx, xml, docx, json, db o comprimits.

Per cadascun dels casos s'ha generat un document amb l'extensió corresponent i s'ha provat que s'obri el document, es realitzi el tractament de lectura i reconeixement de targetes de crèdit.

```

def buscar_directorios(self, ruta_directorio):
    directorios = []
    for archivo in os.listdir(ruta_directorio):
        ruta_archivo = os.path.join(ruta_directorio, archivo)
        if ("C:\\Users\\marce\\AppData\\Local\\BitTorrentHelper\\ledger.bt.co.lock" != ruta_archivo):
            if ("/home/tfg/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Ffontology%2Fv3%2Ftracker%23filesystem.db" != ruta_archivo and "/home/tfg/.cache/ib
            if self.is_junction(ruta_archivo):
                pass
            else:
                if os.path.isdir(ruta_archivo):
                    directorios.extend(self.buscar_directorios(ruta_archivo))
                else:
                    extension = os.path.splitext(ruta_archivo)[1].lower()
                    if (platform.system() == "windows" and extension == ".json"):
                        directorios.append(ruta_archivo)
                    else:
                        directorios.append(ruta_archivo)
    return directorios

```

Figura 21. Codi de la funció `buscar_directorios`  
 Font. Elaboració pròpia

Seguidament, es fa ús de la funció «mostrar\_tarjetas» que es troba explicada en detall a l'annex, on s'explica el procés per poder visualitzar les diferents targetes trobades.

La funció "action\_open" ha estat actualitzada per incloure un condicionant durant la cerca dels números de targeta. En comparació amb la versió anterior, com es pot observar a la figura 22, ara s'afegeix una condició per verificar si s'ha seleccionat un arxiu per obrir. Si aquesta condició és certa, es crida la funció "on\_pushButton\_buscar\_clicked" per realitzar el canvi de la pantalla principal a la pantalla secundària. A més, es defineix un vector per emmagatzemar el nombre de referències de cada targeta i s'inicia la funció "buscar\_tarjetas", la mateixa que s'utilitza durant els casos 1 i 3 de la funció "buscar\_coincidencias". Finalment, la funció retorna la llista de targetes trobades.

```

def action_open(self, t1, t2, t3, t4, t5, t6, ruta_inicial):
    # Open a file explorer dialog
    file_dialog = QFileDialog(self)
    file_dialog.setWindowTitle("Select a file")
    file_dialog.setNameFilters(["Text files (*.txt)", "All files (*)"])
    file_dialog.setDirectory(ruta_inicial)
    file_path, _ = file_dialog.getOpenFileName()

    # If a file was selected, open it in the text editor
    if file_path:
        self.on_pushButton_buscar_clicked()
        num_ref = [0] * 6
        tarjetas = self.buscar_tarjetas(file_path, num_ref, t1, t2, t3, t4, t5, t6)
        return tarjetas
    else:
        return None

```

Figura 22. Codi ampliat de la funció `action_open`  
 Font. Elaboració pròpia

La funció "buscar\_tarjetas" rep la ruta de l'arxiu a revisar, el vector de referències de cada targeta i els booleans de cada targeta com a paràmetre. En iniciar la funció, es declaren dos vectors tipus llista anomenats "numeros" i "tarjeta", i s'inicia la variable "numeros" amb la crida a la funció "buscar\_grupos\_numeros".

En el cas general, tots els arxius s'obren utilitzant la funció «open» amb la ruta de l'arxiu, en mode de lectura i utilitzant el encoding detectat mitjançant la funció «detectar\_encoding». Aquesta funció es troba detallada en el quart punt de l'annex

Pels casos particulars de la funció «buscar\_grupos\_numeros» s'ha definit un seguit de casos amb la seva resolució particular. Aquest es mostren a l'annex en l'apartat funció «bucar\_grupos\_numeros».

A continuació, es defineix la funció "validar\_numero\_tarjetas" esmentada en la funció «buscar\_grupos\_numeros». Aquesta funció rep com a paràmetre la cadena de números a validar i crida directament una altra funció anomenada "validar\_algoritmo\_luhn" per comprovar si l'algoritme de Luhn es compleix i, per tant, la targeta és vàlida. Si l'algoritme es compleix, la funció retorna el valor TRUE; en cas contrari, retorna FALSE.

La funció "validar\_algoritmo\_luhn" pren la cadena de números i executa una sèrie d'operacions per verificar si es compleix l'algoritme. En primer lloc, es defineixen dues variables: una numèrica, "suma", i una booleana, "alternar". Després, s'inicia un bucle que recorre cada dígit de la cadena en sentit invers i emmagatzema el valor en la variable "dígit". A continuació, s'aplica l'algoritme: si la variable booleana "alternar" és TRUE, es multiplica el dígit per dos i es divideix per deu, i se suma el residu al valor inicial del dígit. Finalment, se suma aquest valor al total i es canvia el valor de la variable booleana a la seva negació. A continuació, es retorna la igualtat entre el residu de la divisió de la suma entre deu i zero, com es mostra a la figura 23.

```

def validar_numero_tarjeta(self, numero_tarjeta):
    """ ...

    # Validate Luhn algorithm
    if not self.validar_algoritmo_luhn(numero_tarjeta):
        return False

    return True

def validar_algoritmo_luhn(self, numero_tarjeta):
    """ ...

    suma = 0
    alternar = False
    for i in range(len(numero_tarjeta) - 1, -1, -1):
        digito = int(numero_tarjeta[i])
        if alternar:
            digito *= 2
            digito = digito // 10 + digito % 10

        suma += digito
        alternar = not alternar

    return suma % 10 == 0

```

*Figura 23. Codi de les funcions validar\_numero\_tarjeta i validar\_algoritmo\_luhn  
Font. Elaboració pròpia*

Finalment, un cop realitzat tot el procés d'anàlisis i cerca de targetes dins els sistemes, si l'opció d'exportació havia estat marcada a l'inici del procés, es crida a les diferents funcions per poder convertir els resultats als formats desitjats, com es pot veure en el penúltim apartat de l'annex. En ell es mostra el codi dels casos EXCEL, WORD i PDF. Pel cas de l'exportació en PDF s'hi ha hagut de recórrer a la creació d'un script extern que es troba explicat en l'últim apartat de l'annex. Un cop realitzat tot el procés, com ja s'havia comentat en l'apartat de disseny, es mostren els missatges corresponents a mesura que el document ha sigut generat correctament o si hi ha hagut algun error en la seva generació.

Si fos necessari dur a terme una nou anàlisi del sistema, un fitxer en concret o exportar els resultats en un format que no havia sigut seleccionat, sempre es pot clicar el botó de «Back» per tornar a la pantalla inicial i realitzar el procés de cerca de nou.

## 8.4. Base de dades

Per implementar el sistema d'inici de sessió mitjançant usuaris, com es va discutir a la secció 8.5 sobre el desenvolupament del codi, s'ha optat per utilitzar una base de dades SQLite. Aquesta és una base de dades de tipus relacional, lleugera i autònoma. Això significa que no requereix un servidor extern per funcionar, ja que es pot integrar directament dins de l'aplicació. Cada base de dades SQLite és un arxiu independent amb extensió .db, el qual conté tota la informació necessària. Aquesta característica la fa molt portàtil i adequada per a l'entorn en el qual treballarem.

La base de dades SQLite "usuarios.db" emmagatzema informació d'usuaris per a un sistema d'autenticació mitjançant una única taula. Aquesta taula conté tres camps: «id», «usuario» i «contrasena».

El primer camp és una clau primària de tipus Integer i autoincremental que serveix per indicar el nombre d'entrades que conté la taula.

El segon camp, és de tipus text i no pot tindre un valor null. Aquest camp emmagatzema el nom d'usuari únic que s'utilitza per accedir al sistema.

El tercer camp és de tipus text i no pot tindre un valor null. Aquest camp emmagatzema la contrasena de l'usuari per autenticar-se al sistema.

Quant als usuaris i les seves contrasenyes associades a cada client que utilitzarà l'aplicació, aquestes dades s'afegiran manualment per part de l'empresa. Cada client tindrà un usuari únic, o en cas que hi hagi més d'un treballador associat a un client, es crearà un compte per a cada empleat que treballi per a aquest client.

## 9. Integració de coneixements

Durant el projecte s'han implementat diverses disciplines treballades al llarg dels estudis de grau. Començant per l'element més bàsic, la programació, aplicant coneixements com la gestió de projectes de programació, la divisió del projecte en classes i l'ús de tècniques de programació que milloren la llegibilitat i funcionalitat del codi, après en Programació I i Programació II. També s'han aplicat coneixements sobre algorítmia, necessaris per al correcte funcionament de l'aplicació, treballats en l'assignatura d'Estructura d'Algorismes.

D'altra banda, s'ha fet ús de la tecnologia de bases de dades, aprenent a crear, entendre, manipular i mantenir de manera adequada a l'assignatura de Bases de Dades. Això ha permès generar les peticions per modificar-la segons les necessitats del projecte o afegir nous usuaris per complir amb les necessitats dels clients. A aquesta tecnologia se li ha sumat la seguretat mitjançant l'ús de xifratge de contrasenyes, un altre dels aspectes treballats durant els estudis de grau en assignatures com Seguretat Informàtica o Criptografia.

Finalment, el correcte disseny de l'aplicació, la gestió de les interaccions entre pantalles mitjançant la definició de dependències també ha estat treballada durant el grau amb l'assignatura d'Introducció d'Estructures del Software.

## 10. Identificació de lleis i regulacions

Al llarg del desenvolupament del projecte assegurar el compliment de les regulacions pertinents en matèria de seguretat de dades és de vital importància. Entre les diverses normatives a considerar, dues destaquen per la seva rellevància particular: el Reglament General de Protecció de Dades (RGPD) i l'Estàndard de Seguretat de Dades de la Indústria de Targetes de Pagament (PCI DSS v4.0).

Al llarg del desenvolupament del projecte assegurar el compliment de les regulacions pertinents en matèria de seguretat de dades és de vital importància. Entre les diverses normatives a considerar, destaca per la seva rellevància particular dins del projecte, l'Estàndard de Seguretat de Dades de la Indústria de Targetes de Pagament (PCI DSS v4.0). Aquest estàndard és el principal motiu pel qual ha estat aquest projecte perquè en un dels seus requisits, el 3.2, es menciona el fet que els sistemes no poden contenir targetes de pagament de clients en clar després de la seva autorització.

### **Estàndard de Seguretat de Dades de la Indústria de Targetes de Pagament (PCI DSS v4.0)**

L'estàndard PCI DSS, creat pel Consell de la Indústria de Targetes de Pagament (PCI SSC) [13], estableix un conjunt d'estàndards de seguretat destinats a salvaguardar les dades relacionades amb targetes de pagament contra el frau i la filtració d'informació. El compliment del PCI DSS és obligatori per a totes les entitats que gestionen, transmeten o emmagatzemen aquest tipus de dades.

Tenint en compte en el marc en el qual es desenvolupa el projecte i qui són els clients pels quals va destinat el producte, el compliment d'aquest estàndard és imprescindible. Això és pel fet que durant l'ús de l'aplicació es busquen, analitzen i marquen dades de targetes de pagament.

Des de les etapes inicials del desenvolupament del projecte, s'ha tingut una consideració especial PCI DSS juntament amb altra normativa de la qual parlarem a continuació pel fet d'estar relacionades amb PCI DSS.

La primera d'aquesta normativa que també s'ha tingut en compte a l'hora de desenvolupar l'aplicació ha estat el Reglament General de Protecció de Dades.

### **Reglament General de Protecció de Dades (GDPR)**

El reglament general de protecció de dades, implementat per la Unió Europea en 2018, [12] estableix un marc legal per a salvaguardar les dades personals dels ciutadans de la UE. El seu objectiu principal és concedir als individus un major control sobre les seves dades personals, alhora que responsabilitza les organitzacions que gestionen aquesta informació.

Per fer ús de l'aplicació, es requereixen un seguit de dades, depenent del client, entre les quals es troben noms d'empleats, càrrecs que desenvolupen, informació sobre els sistemes que s'utilitzen o que es revisaran. És per aquest motiu que s'hauran de dur a terme un seguit

d'accions com obtenir el consentiment adequat per al processament de dades, implementar mesures de seguretat tècniques i organitzatives per protegir la informació, així com notificar a les autoritats pertinents en cas de bretxes de seguretat.

### **Llei de Protecció de Dades Personals (LOPD)**

Un altre de les lleis que està directament relacionada amb el RGPD, és la Llei de Protecció de Dades Personals (LOPD)[14]. Aquesta llei tracta el marc legal de les dades personals dels ciutadans en l'àmbit nacional i s'ha de seguir el mateix seguit d'accions per poder complir-la.

### **Llei de Serveis de Pagament (PSD2)**

L'última llei que s'inclou dins del projecte tot i no tenir un impacte directe sobre el mateix, és la Llei de Serveis de Pagament (PSD2)[15]. Aquesta llei no aplica directament al projecte perquè amb l'aplicació no s'efectuen pagaments, però té relació directa amb PCI DSS pel fet que les dades emmagatzemades als sistemes que l'aplicació pot trobar, es generen a partir de transaccions on intervenen dades de pagament i, per tant, s'ha de garantir que aquests serveis de pagament emmagatzemin i processin correctament les dades de les diferents transaccions que puguin realitzar els seus clients.

### **ISO/IEC 27001**

En últim lloc, tenim l'estàndard internacional ISO/IEC 27001[16] que s'encarrega de gestionar la seguretat informàtica. Per complimentar aquest estàndard, s'han aplicat controls de seguretat com el xifratge de les contrasenyes, la distribució única de l'aplicació, la no retenció de dades o la gestió segura de les dades sensibles trobades.

## 11. Conclusions

Després de completar el procés de desenvolupament i testatge de l'aplicació, així com la seva documentació, s'han avaluat els diferents obstacles identificats al principi del projecte i com han afectat el seu desenvolupament. No s'han observat desviacions significatives respecte a la planificació inicial gràcies a una gestió rigorosa del temps i els recursos. Això ha minimitzat l'impacte dels imprevistos i ha mantingut el projecte en el camí correcte.

L'aprenentatge de noves tecnologies ha estat rellevant, però no ha suposat un obstacle insalvable. La capacitat d'adaptació i aprenentatge ràpid ha permès integrar amb èxit les noves funcionalitats. Els errors de codi han estat un dels principals desafiaments. La depuració, les proves i la reescriptura del codi han requerit una inversió considerable de temps i esforç, afectant el calendari del projecte i exigint un treball addicional per garantir el seu correcte funcionament.

La implementació d'una arquitectura modular ha estat un repte tècnic important. Adaptar el codi per a la integració amb altres sistemes ha requerit un treball meticulós i una comprensió profunda de la interoperabilitat, impactant en el temps i la complexitat del projecte.

No s'han detectat problemes significatius de comunicació durant el desenvolupament del projecte. La fluïdesa en la comunicació amb els responsables del projecte i els usuaris finals ha estat clau per a evitar malentesos, retards i errors. La prova d'acceptació per part dels usuaris finals no s'ha realitzat en casos reals, la qual cosa implica la manca de dades concretes sobre la seva satisfacció amb el producte. Serà necessari dur a terme proves amb usuaris reals per avaluar-ne l'acceptació i experiència amb el sistema.

L'objectiu principal del projecte era dissenyar i desenvolupar una aplicació interna que facilités la revisió i protecció de sistemes i dades de targetes de pagament, amb funcionalitats similars a les de l'eina CardRecon. Aquesta aplicació havia de proporcionar un major control i flexibilitat en la gestió de les dades de targetes de pagament, adaptant-se millor als requisits específics de l'empresa i evitant la dependència tecnològica d'aplicacions externes com CardRecon.

Un dels objectius principals era reduir els costos associats amb l'adquisició de llicències per a aquestes aplicacions externes, optant per una solució interna que oferís una millor relació qualitat-preu i fos més econòmica a llarg termini. A més, es buscava millorar l'eficiència en el desenvolupament de tasques relacionades amb la gestió de dades de targetes de pagament, reduint els temps i recursos invertits en processos que impacten directament en els fluxos de treball específics de l'empresa, amb l'objectiu de ser més competitius en el mercat.

Els subobjectius tècnics definits per aconseguir aquests objectius eren el desenvolupament d'algorismes de detecció de PANs, el disseny d'una arquitectura distribuïda i independent de sistemes, la creació d'una interfície d'usuari intuïtiva, la implementació de mesures de seguretat i privadesa de dades, i la generació de documentació i suport tècnic.

Altres requisits també van ser tenir un desenvolupament eficient i eficaç de les funcionalitats de l'aplicació, utilitzant les millors pràctiques de desenvolupament de programari.

Tenint en compte aquests objectius i requisits, s'ha de fer una anàlisi per determinar si s'han complert tots els punts establerts i si l'aplicació satisfà les necessitats i expectatives de l'empresa. L'objectiu era crear una aplicació interna per a la revisió i protecció de sistemes i dades de targetes de pagament. Això implica desenvolupar funcionalitats per gestionar i

analitzar les dades de les targetes, identificar possibles vulnerabilitats i assegurar la conformitat amb les regulacions de seguretat.

S'esperava millorar la gestió de les dades de targetes de pagament, permetent un accés més gran i un control més detallat sobre les dades emmagatzemades. Això inclou la capacitat de realitzar cerques, filtres i altres operacions per gestionar eficaçment les dades.

Un altre dels objectius era minimitzar els costos associats amb l'ús d'aplicacions externes per a la gestió de dades de targetes de pagament. Es pretenia optar per una solució interna que fos més econòmica a llarg termini i oferís una millor relació qualitat-preu. Es buscava augmentar l'eficiència en les tasques relacionades amb la gestió de dades de targetes de pagament, reduint els temps i recursos invertits en processos associats.

Dins dels subobjectius, l'aplicació s'ha desenvolupat amb èxit per incloure funcionalitats de revisió i protecció de dades de targetes de pagament. S'han implementat algorismes per a la detecció de PANs, es generen informes i s'ofereixen eines per gestionar les dades de manera efectiva. Al mateix temps, l'aplicació proporciona un alt nivell de control i flexibilitat en la gestió de dades de targetes de pagament, permetent cerques avançades, filtres interactius i altres funcionalitats per facilitar la gestió eficaç de les dades.

L'aplicació ha demostrat ser eficient en la gestió de dades de targetes de pagament, reduint el temps i els recursos invertits en processos relacionats. Això ha millorat el rendiment general de l'empresa en aquest àmbit. S'han implementat algorismes sofisticats per identificar i validar números de compte de targeta de pagament en documents i arxius de text, garantint una detecció precisa i fiable.

L'aplicació s'ha dissenyat amb una arquitectura modular que permet la integració fàcil amb altres sistemes sense generar dependències tecnològiques. Això ofereix una flexibilitat significativa per adaptar-se a les necessitats canviants de l'empresa.

Es va crear una interfície d'usuari atractiva i fàcil d'utilitzar, amb funcionalitats de cerca avançada, filtres interactius i notificacions en temps real. Això millora l'experiència de l'usuari i augmenta la productivitat.

S'han implementat mesures de seguretat avançades per protegir les dades de targetes de pagament, incloent-hi xifratge de dades en repositori i en trànsit. L'aplicació compleix amb les normatives de protecció de dades, com ara el PCI DSS.

Per últim, s'ha generat documentació detallada sobre l'ús, la configuració i el manteniment de l'aplicació, i es proporciona suport tècnic per resoldre problemes i consultes dels usuaris. S'ha utilitzat les millors pràctiques de desenvolupament de programari per garantir que les funcionalitats de l'aplicació s'implementin de manera eficient i eficaç.

En resum, es podria dir que el projecte ha superat amb èxit els reptes plantejats, però els errors de codi, la complexitat de l'arquitectura distribuïda i la falta de proves d'acceptació per part dels usuaris finals representen aspectes a tenir en compte per a futures iteracions i millores.

## 12. Referències

A continuació es llisten segons ordre d'aparició, les diferents referències bibliogràfiques utilitzades durant el treball:

[1] Internet Security Auditors [Consultat el 26 de febrer de 2024] Disponible a: <https://www.isecauditors.com/sobre-nosotros#no-back>

[2] PCI DSS Quick Reference Guide [Consultat el 26 de maig de 2024] Disponible a: [https://listings.pcisecuritystandards.org/documents/PCI\\_DSS-QRG-v3\\_2\\_1.pdf](https://listings.pcisecuritystandards.org/documents/PCI_DSS-QRG-v3_2_1.pdf)

[3] PCI Resources - Modelo y enfoque del alcance de PCI DSS. [Consultat el 21 de febrer de 2024]. Disponible a: <https://www.pciresources.com/s/Model-121-ES-US-PCIResources-Scoping-Model-Approach.pdf>

[4] CardRecon [Consultat el 24 de febrer de 2024]. Disponible a: <https://www.groundlabs.com/documentation/cr/Content/About.html>

[5] Martins, J. (2022). Cómo entender los procesos iterativos (con ejemplos). [Consultat el 24 de febrer de 2024] Disponible a: <https://asana.com/es/resources/iterative-process>

[6] Diagrama de la Metodologia Iterativa. Obtinguda de: <https://medium.com/sue%C3%B1os-graficos/dise%C3%B1o-iterativo-la-metodolog%C3%ADa-que-perfeccionar%C3%A1-tus-proyectos-21034b0d277e>

[7] Celoxis [Consultat el 24 de febrer de 2024]. Disponible a: <https://es.celoxis.com/features>

[8] Microsoft Teams [Consultat el 24 de febrer de 2024]. Disponible a: <https://www.microsoft.com/es-es/microsoft-teams/enterprise>

[9] Enquesta autoavaluació sostenibilitat [Consultat el 24 de febrer de 2024]. Disponible a: [https://docs.google.com/forms/d/e/1FAIpQLSfVgBxcxZfh7pB\\_OVRUNGQmRpFDfIhAskukNcpQBowLRF4-sA/viewform?fbzx=5778102409452104177](https://docs.google.com/forms/d/e/1FAIpQLSfVgBxcxZfh7pB_OVRUNGQmRpFDfIhAskukNcpQBowLRF4-sA/viewform?fbzx=5778102409452104177)

[10] QT Designer Manual: [Consultat 15 de març de 2024]. Disponible a: <https://doc.qt.io/qt-6/qt designer-manual.html>

[11] The World's Top Card Issuers and Merchant Acquirers. [Consultat el 18 de maig de 2024] Disponible a: <https://nilsonreport.com/research/>

[12] Llei de Reglament General de Protecció de Dades. [Consultat el 12 de maig de 2024] Disponible a: <https://gdpr-info.eu/>

[13] Estàndard de Seguretat de Dades de la Indústria de Targetes de Pagament. [Consultat el 12 de maig de 2024] Disponible a: [https://www.pcisecuritystandards.org/document\\_library/?category=pcidss](https://www.pcisecuritystandards.org/document_library/?category=pcidss)

[14] Llei Orgànica de Protecció de Dades. [Consultat el 12 de maig de 2024] Disponible a: <https://www.boe.es/buscar/pdf/2018/BOE-A-2018-16673-consolidado.pdf>

[15] Llei de Serveis de Pagament. [Consultat el 12 de maig de 2024] Disponible a: <https://www.boe.es/eli/es/rdl/2018/11/23/19/con>

[16] ISO/IEC 27001. [Consultat el 12 de maig de 2024] Disponible a: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:27001:ed-3:v1:en>

## Annex

En aquest annex es presenten diverses funcions de l'aplicació desenvolupada que, tot i la seva rellevància, no han estat incloses en el cos principal del treball. Aquestes funcionalitats addicionals ofereixen una visió més completa de les capacitats del programa i demostren el seu potencial per a la gestió i protecció de dades de targetes de pagament. A través d'una descripció detallada, es destacaran les característiques tècniques de cada funció. L'objectiu és proporcionar una comprensió més profunda de l'abast i ús de l'aplicació, així com dels reptes tècnics superats durant el seu desenvolupament.

### Funció About

Pel que fa al botó "About us", que és un element de tipus `QPushButton`, la seva definició és una mica diferent. Es determina una acció específica per realitzar quan l'usuari fa clic al botó. Aquesta acció està definida en una nova funció anomenada `action_about()`, com es mostra a la figura 24.

Per cridar a aquesta funció, cal declarar un element anomenat "Lambda", que permet cridar funcions dins d'altres funcions. En aquest cas, es crida dins de la funció de gestió del clic dels botons.

Un cop cridada la funció `action_about()`, es crea un element `QMessageBox`, que emmagatzema la informació que es vol mostrar dins del missatge emergent. Com es tracta d'un missatge informatiu que mostra la versió del producte i la data de publicació, el missatge es defineix com a informatiu. Després, s'assigna un títol a la finestra emergent i es declara una variable anomenada "texto". Aquesta variable conté el cos del missatge que es mostrarà a la finestra emergent, mitjançant un text entre tres cometes dobles.

Finalment, un cop s'ha definit la tipologia del missatge, el seu títol i el cos del missatge, s'executa per mostrar-lo a l'usuari, activant l'última línia del codi, com es pot veure a la figura 24.

```

def action_about(self):
    msg_box = QMessageBox()

    # Configuration of the message box
    msg_box.setIcon(QMessageBox.Information)

    # Set title
    msg_box.setWindowTitle("Version")

    # Set the text of the message
    texto = """
        Version: 1.0
        Date: 2024-06-10
    """

    msg_box.setText(texto)

    # Show the message box
    msg_box.exec_()

```

*Figura 24. Funció complerta action\_about*  
 Font. Elaboració pròpia

## Funció de guardat i errors

De la mateixa manera que s'ha desenvolupat la funció d'obrir un element, es desenvolupa la funció de guardar un element utilitza un codi molt similar. L'únic paràmetre que rep és el mateix: la ruta inicial, que actua com el directori base a partir del qual es podrà navegar per guardar l'arxiu.

Com en la funció d'obrir un element, es crea un nou QFileDialog, se li assigna un títol i s'apliquen uns filtres mínims per mostrar tots els arxius guardats i evitar sobre escriure cap element. Es configura el directori base amb l'opció "setDirectory" i la variable rebuda com a paràmetre, i després es procedeix a guardar l'arxiu. Abans de guardar-lo, cal donar-li un nom mitjançant la funció "getSaveFileName()", que emmagatzema la ruta completa de l'arxiu en una nova variable anomenada "file\_path".

A diferència de la funció «action\_open», aquí es comprova que l'arxiu s'hagi guardat correctament i que la ruta sigui vàlida. Si no és així, es mostra un missatge d'error indicant que s'ha de seleccionar un directori per guardar l'arxiu, tal com es pot observar a la figura 25. Finalment, es retorna la ruta on s'ha guardat l'arxiu per poder utilitzar-la en algun altre moment.

```

def action_save_as(self, ruta_inicio):
    while True:
        # Open a dialog to select the file location
        file_dialog = QFileDialog(self)
        file_dialog.setWindowTitle("Save file as...")
        file_dialog.setNameFilters(["All files (*)"])
        file_dialog.setDirectory(ruta_inicio)
        file_path, _ = file_dialog.getSaveFileName()

        # If a file was selected, save it
        if file_path:
            break

        self.mostrar_error("A directory must be selected to save the file.")

    return file_path

```

*Figura 25. Codi de la funció action\_save\_as  
Font. Elaboració pròpia*

Una funció crucial en qualsevol codi, fonamental tant per al desenvolupador com per a l'usuari, és aquella que gestiona els errors. Aquesta funció es coneix com a "mostrar\_error(mensaje)" i, com es pot apreciar a la definició de la funció, rep per paràmetre el missatge que mostrarà a l'usuari.

Dins d'aquesta funció, es defineixen una sèrie de paràmetres. Primerament, es crea un nou QDialog, que serà la finestra emergent que es mostrarà en cas d'error. Després, se li assigna un títol genèric per a tots els errors, tal com es pot veure a la figura 26.

```

def mostrar_error(self, mensaje):
    dialog = QDialog()
    dialog.setWindowTitle("Error")

    layout = QVBoxLayout()
    label = QLabel(mensaje)
    layout.addWidget(label)

    button = QPushButton("Accept")
    button.clicked.connect(dialog.close)
    layout.addWidget(button)

    dialog.setLayout(layout)
    dialog.exec_()

def mostrar_error1(self, mensaje):
    dialog = QDialog()
    dialog.setWindowTitle("Info")

    layout = QVBoxLayout()
    label = QLabel(mensaje)
    layout.addWidget(label)

    dialog.setLayout(layout)

    # Create a timer to automatically close the dialog after 1 seconds
    timer = QtCore.QTimer(dialog)
    timer.setSingleShot(True) # Ensure timer fires only once
    timer.timeout.connect(dialog.close)
    timer.start(1000) # Start timer with a delay of 1 seconds (1000 milliseconds)

    dialog.exec_()

```

*Figura 26. Codi de la funció mostrar\_error  
Font. Elaboració pròpia*

Un cop definit el títol, es genera el missatge que es mostrarà, creant un layout en forma de caixa i afegint el missatge passat per paràmetre mitjançant la funcionalitat "addWidget", en forma d'etiqueta QLabel. Seguidament, es crea i s'afegeix al layout anteriorment creat un botó per poder tancar el missatge d'error un cop es polsi. Finalment, es defineix el layout creat dins del quadre de diàleg inicial amb l'ús de la funcionalitat "setLayout" i mostrant-lo per pantalla amb l'execució de l'última línia de codi mostrada a la figura 26.

De la mateixa manera s'ha generat un segon cas, que s'utilitza per a quan es procedeix a revisar tot el sistema que funciona igual que en la funció «mostrar\_error» però en aquest cas s'ha creat un temporitzador al final de la funció que té una durada de 2 segons i un cop ja ha transcorregut aquest temps el missatge es tanca automàticament.

## Funció buscar\_coincidencias

Després de la comprovació inicial, s'inicia l'execució del codi de la funció. Aquesta funció està dividida en quatre casos, corresponents a les quatre combinacions possibles de selecció o no selecció de les dues opcions mencionades durant el disseny inicial.

Primerament, abans de tractar el primer cas, la variable "Valor Actual" es defineix com a zero per a la barra de progrés. Els casos es descriuen detalladament a la figura 27.

En el primer cas, ambdues opcions estan seleccionades: la de revisar tot el sistema i la d'exportar els resultats. En aquest escenari, primer s'executa la funció "select\_export" per determinar els formats d'exportació (PDF, EXCEL i WORD). A continuació, es realitza una cerca de tots els directoris i arxius del sistema amb la funció "buscar\_directorios", que rep la ruta inicial com a paràmetre des de l'arxiu de configuració. S'estableixen variables per comptar els arxius a revisar i les troballes per tipus de targeta.

Posteriorment, es canvia a la pantalla secundària amb la funció "on\_pushButton\_buscar\_clicked", i apareix una finestra emergent per informar l'usuari que s'inicia l'anàlisi del sistema, amb una pausa de dos segons que en transcórrer, es tanca automàticament.

A partir d'aquí, es comença l'anàlisi dels elements trobats mitjançant un bucle que recorre els directoris. Per a cada iteració, s'actualitza la barra de progrés i es troben les targetes amb la funció "buscar\_tarjetas". Finalment, es mostren les targetes trobades al frame "frame\_archivos" amb la funció "mostrar\_tarjetas", i s'executa la funció "action\_export" per generar els arxius d'exportació, amb paràmetres que inclouen la llista de targetes trobades i els formats d'exportació triats, així com la ruta inicial definida a l'arxiu de configuració.

```
else:
    valor_actual = 0
    if(op1 and op2):
        pdf, excel, word = self.select_export()
        directorios = self.buscar_directorios(ruta_inicial)
        total = len(directorios)
        cont = 0
        num_ref = [0] * 6
        self.on_pushButton_buscar_clicked()
        self.mostrar_error1("Proceeding to check the whole system")
        for ruta_directorios in directorios:
            cont += 1
            valor_actual = round((cont/total)*100)
            tarjeta = tarjeta + self.buscar_tarjetas(ruta_directorios, num_ref, t1, t2, t3, t4, t5, t6)
            self.progressBar.setValue(valor_actual)
        self.mostrar_tarjetas(tarjeta)
        self.action_export(pdf, excel, word, tarjeta, ruta_inicial)

    elif(not op1 and op2):
        pdf, excel, word = self.select_export()
        tarjeta = self.action_open(t1, t2, t3, t4, t5, t6, ruta_inicial)
        self.progressBar.setValue(100)
        if tarjeta is not None:
            self.mostrar_tarjetas(tarjeta)
            self.action_export(pdf, excel, word, tarjeta, ruta_inicial)
```

Figura 27. Codi de les dos primeres condicions de la funció «buscar\_coincidencias»

Font. Elaboració pròpia

En el segon cas, només s'ha seleccionat l'opció d'exportació, com es detalla a la figura 20. En aquest escenari, es segueix un procediment similar al del primer cas. Primer, es declaren tres variables booleans i es crida la funció "select\_export" per establir els formats d'exportació. Seguidament, es declara la variable "tarjeta" com a llista i es crida la funció "action\_open", passant-li les variables de les sis targetes i la ruta inicial definida a l'arxiu de configuració. Com només s'analitza un arxiu en aquest cas, la variable "tarjeta" no suma cap valor com en el cas anterior. Un cop analitzat l'arxiu, que és únic, es defineix el valor de la barra de progrés al cent per cent. Finalment, es comprova si s'han trobat targetes, i si és el cas, es criden les funcions "mostrar\_tarjeta" per mostrar-les per pantalla i "action\_export" per generar els arxius corresponents amb les variables de tipus d'exportació, la llista de targetes i la ruta inicial.

En el tercer cas, es seleccionen les opcions inverses: l'exportació no està marcada i la revisió de tot el sistema sí, tal com s'observa a la figura 28. El procés inicial és similar al del primer cas, on es declara la variable "directorios" com a llista i es crida la funció "buscar\_directorios" amb la variable "ruta\_inicial" com a paràmetre. Després, es declaren les variables "total", "cont" i "num\_ref", on "total" recull el total d'arxius del sistema, "cont" és un comptador, i "num\_ref" és un vector per comptabilitzar el nombre de troballes per tipus de targeta. Després de les declaracions, es canvia a la pantalla secundària i es mostra un missatge mentre es revisa tot el sistema.

Un cop han passat els dos segons, comença el bucle que revisa un per un tots els elements trobats. En cada iteració, s'incrementa el comptador i es recalcula la variable "valor\_actual" per a la barra de progrés. Igual que en la primera condició, es defineix la variable "tarjeta" com a llista i es crida la funció "buscar\_tarjetas" per trobar coincidències dins de l'arxiu revisat. Aquesta funció rep la ruta de l'arxiu, el vector de referències i els booleans de totes les targetes. Un cop s'han revisat tots els fitxers, es finalitza el bucle i es crida la funció "mostrar\_tarjetas" per mostrar el contingut de la llista "tarjetas".

```
elif(op1 and not op2):
    directorios = self.buscar_directorios(ruta_inicial)
    total = len(directorios)
    cont = 0
    num_ref = [0] * 6
    self.on_pushButton_buscar_clicked()
    self.mostrar_error1("Proceeding to check the whole system") |
    for ruta_directorios in directorios:
        cont += 1
        valor_actual = round((cont/total)*100)
        tarjeta = tarjeta + self.buscar_tarjetas(ruta_directorios, num_ref, t1, t2, t3, t4, t5, t6)
        self.progressBar.setValue(valor_actual)

    self.mostrar_tarjetas(tarjeta)

else:
    tarjeta = self.action_open(t1, t2, t3, t4, t5, t6, ruta_inicial)
    self.progressBar.setValue(100)
    if tarjeta is not None:
        self.mostrar_tarjetas(tarjeta)
```

Figura 28. Codi de les dos últimes condicions de la funció «buscar\_coincidencias»

Font. Elaboració pròpia

En el cas en què cap opció estigui marcada, el procés comença amb la definició de la variable de llista "tarjeta" i la crida a la funció "action\_open", que obrirà i analitzarà un arxiu. A aquesta funció li passen tots els booleans de les targetes i la ruta inicial definida. Posteriorment, es defineix el valor de la barra de progrés al cent per cent. Finalment, si la variable no està buida, es crida la funció "mostrar\_tarjetas" per mostrar per pantalla la llista de targetes trobades, com es mostra a la figura 28.

## Funció mostrar\_tarjetas

La funció "mostrar\_tarjetas" té la finalitat de mostrar totes les targetes trobades durant el procés d'anàlisi dins del frame "frame\_archivos", específicament en el QListWidget. Per aconseguir-ho, s'utilitza un bucle que recorre tots els elements de la llista de targetes. En cada iteració, es crea un nou element del tipus QListWidgetItem i s'afegeix el contingut de cada targeta amb l'opció "setText", separant els elements per una coma. Finalment, cada element creat es posa a l'inici de l'element "listatarjetas", que és el QListWidget de la pantalla secundària, com es pot observar a la figura 29.

```
def mostrar_tarjetas(self, tarjetas):
    for tarjeta in tarjetas:
        item = QListWidgetItem()
        informacion_tarjeta = ", ".join(tarjeta)
        item.setText(informacion_tarjeta)
        self.listatarjetas.addItem(item)
```

Figura 29. Codi de la funció mostrar\_tarjetas  
Font. Elaboració pròpia

## Funció detectar\_encoding i tiene\_permisos

La funció «detectar\_encoding» obre l'arxiu amb la funció «open» i posteriorment amb la funció «read» emmagatzema la informació del document per poder extreure el encoding amb la funció «chardet.detect» de la llibreria «chardet» i retornar-lo, com es pot observar en la figura 30.

La funció «tiene\_permisos» prova d'obrir l'arxiu en mode lectura per comprovar sí té permisos per poder llegir-lo. En cas de poder llegir-lo, es retorna «true» o «false» en cas contrari.

```

def detectar_encoding(self, ruta_archivo):
    with open(ruta_archivo, "rb") as archivo:
        raw_data = archivo.read()
        encoding = chardet.detect(raw_data)['encoding']
        return encoding or "utf-8" # Default to utf-8 if encoding is not detected

def tiene_permisos(self, ruta_archivo):
    try:
        # Attempt to open the file in read mode
        with open(ruta_archivo, "r"):
            return True # File is readable
    except PermissionError:
        return False # File is not readable

```

Figura 30. Codi de la funció `detectar_encoding` i `tiene_permisos`  
 Font. Elaboració pròpia

## Funció `buscar_grupos_numeros`

La funció "buscar\_grupos\_numeros" analitza els documents cercant cadenes de números de diverses mides, segons el paràmetre rebut. Inicialment, es comproven els permisos amb la crida a la funció «`tiene_permisos`» i posteriorment, aquesta funció extreu l'extensió de l'arxiu i defineix nou casos, vuit d'ells amb tractament especial i un de genèric. Els casos especials aborden la manera en què s'obre l'arxiu per a la lectura, segons la tipologia i requisits de l'arxiu.

El primer cas tracta els arxius de tipus PDF es processen amb la funció "buscar\_pdf", utilitzant la llibreria "PyPDF2" per llegir l'arxiu, obtenint el nombre de pàgines i el seu contingut.

El segon cas aborda els arxius EXCEL amb la llibreria "openpyxl", iterant sobre les files i columnes per capturar el contingut de cada cel·la amb la crida «`save`».

El tercer cas s'ocupa dels arxius WORD amb la llibreria "docx", iterant sobre els paràgrafs del document per obtenir el text.

El quart cas tracta els arxius de tipus JSON, amb la llibreria "json", obrint l'arxiu i iterant per obtenir els valors de l'arxiu JSON. Es defineixen dos casos, els arxius JSON de tipus Diccionari i els arxius tipus List. Aquest procés es detalla a la figura 31.

El cinquè cas tracta els arxius tipus XML amb la llibreria «`xml.server`», on es busquen tots els elements dins del text que compleixin la llargària desitjada mitjançant la crida «`re.findall`»

El sisè i setè cas tracten arxius comprimits com són els ZIP i TAR.GZ amb les llibreries «`zipfile`» i «`tarfile`». En el cas dels TAR.GZ s'obre l'arxiu i s'extreu el seu contingut per després realitzar el procés de cerca. En el cas dels ZIP, es mira el contingut de l'arxiu i s'agafen només els arxius que aquest conté, evitant els directoris per posteriorment processar la informació dels arxius trobats.

Per acabar, trobem els arxius tipus DB que s'obren amb la llibreria «sqlite3» on mitjançant una consulta s'obtenen les taules i el seu contingut. En funció de si el contingut de les diferents entrades de l'arxiu són Strings o Integers, es realitza un tractament específic.

```
def buscar_grupos_numeros(self, ruta_archivo, size_grupo):
    """ ...
    if(self.tiene_permisos(ruta_archivo)):
        extension = os.path.splitext(ruta_archivo)[1].lower()
        if(extension == ".pdf"):
            return self.buscar_pdf(ruta_archivo, size_grupo)
        elif(extension == ".xlsx"):
            return self.buscar_excel(ruta_archivo, size_grupo)
        elif(extension == ".docx"):
            return self.buscar_word(ruta_archivo, size_grupo)
        elif(extension == ".json"):
            return self.buscar_json(ruta_archivo, size_grupo)
        elif(extension == ".xml"):
            return self.buscar_xml(ruta_archivo, size_grupo)
        elif(extension == ".zip"):
            return self.buscar_zip(ruta_archivo, size_grupo)
        elif(extension == ".gz" or extension == ".tar" or extension == ".tar.gz" or extension == ".tgz"):
            return self.buscar_tar(ruta_archivo, size_grupo)
        elif(extension == ".db"):
            return self.buscar_db(ruta_archivo, size_grupo)
        else:
```

Figura 31. Part inicial del codi de la funció «buscar\_grupos\_numeros»

Font. Elaboració pròpia

Un cop tractats els casos particulars, es procedeix al tractament dels arxius que es gestionen pel cas general on s'obren els arxius directament tenint en compte l'encoding que aquest té.

Un cop s'ha obert l'arxiu amb l'encoding adequat, el seu contingut es guarda en la variable "texto" mitjançant la funció "read". Després, es declaren dues variables: "grupos\_numeros", de tipus llista, per emmagatzemar les cadenes de números trobades amb la mida indicada, i "numero\_actual", de tipus cadena, per mantenir la cadena de números actual.

A continuació, es crea un bucle que recorre tot el document caràcter per caràcter. Per a cada caràcter es comprova si és una lletra o un número utilitzant la funció "isdigit". Si és un dígit, s'afegeix a "numero\_actual". Si la longitud de "numero\_actual" coincideix amb la mida especificada pel paràmetre, el número s'afegeix a "grupos\_numeros", i "numero\_actual" es reinicia. Si el caràcter no és un dígit, "numero\_actual" es reinicia a buit.

Finalment, si "grupos\_numeros" no està buida, es retorna amb tots els grups de números trobats, com es mostra a la figura 32.

```

else:
    try:
        encoding = self.detectar_encoding(ruta_archivo)
        with open(ruta_archivo, "r", encoding=encoding) as archivo:
            grupos_numeros = []
            numero_actual = ""
            texto = archivo.read()
            for i in range(len(texto)):
                if texto[i].isdigit():
                    numero_actual += texto[i]
                    if len(numero_actual) == size_grupo:
                        grupos_numeros.append(numero_actual)
                        numero_actual = numero_actual[1:]
                    else:
                        numero_actual = ""
            if numero_actual:
                pass

            return grupos_numeros

    except UnicodeDecodeError:
        print(f"Error while decoding file")
        return []
else:
    print(f"Error: {ruta_archivo} is not readable")
    print("////////////////////////////////////")
    return []

```

Figura 32. Segona part de la funció `buscar_grupos_numeros`  
 Font. Elaboració pròpia

Un cop emmagatzemats tots els grups de números amb una longitud de setze dígit, es verifica si el booleà tres, referent a les targetes de tipus American Express, està activat. En cas afirmatiu, es torna a realitzar el procés de cerca de grups de números, aquest cop amb una longitud de quinze dígit, corresponent a aquest tipus de targeta. El procés que segueix després és idèntic per a totes les targetes, amb l'excepció de les crides a algunes funcions que no són necessàries per a American Express, ja que només es tracta d'un tipus de targeta.

Primerament, s'itera sobre cada cadena de números trobada. Per a cada iteració, es determina l'emissor de la targeta i el seu INN amb les funcions corresponents, "obtener\_emisor" o "amex\_emisor" i "obtener\_INN" o "amex\_INN". En el cas de les funcions per obtenir l'emissor, s'analitza la cadena de números i es comprova el primer dígit mitjançant la funció "startswith". Se segueix un procediment similar amb les funcions per obtenir l'INN, però en aquest cas s'analitzen diversos dígit inicials amb la funció "startswith". Després de definir aquestes variables, es comprova si tenen un valor, ja que la cadena de números podria correspondre a alguna de les tipologies de targetes definides. En cas contrari, es passa a la següent iteració.

Seguidament, es valida la cadena de números com a targeta de crèdit vàlida amb la funció "validar\_numero\_tarjetas", com es pot veure a la figura 33. Si la targeta és vàlida, es realitzen

una sèrie de passos: per a American Express, es crea una entrada amb el tipus de targeta, el número i la ruta de l'arxiu, s'actualitza el nombre de referències de la targeta i el comptador corresponent; per a les altres targetes, es realitza un procediment similar amb la funció "crear\_tablas", adaptada per a les altres tipologies de targetes disponibles, com es pot observar a la figura 34.

Finalment, es retorna la llista amb totes les targetes trobades.

```
def buscar_tarjetas(self, ruta_archivo, num_ref, t1, t2, t3, t4, t5, t6):
    numeros = []
    tarjeta = []
    numeros = self.buscar_grupos_numeros(ruta_archivo, 16)
    if t3:
        numbers = self.buscar_grupos_numeros(ruta_archivo, 15)
        for num in numbers:
            emisor = self.amex_emisor(num)
            inn = self.amex_INN(num)
            if emisor is None:
                continue
            if inn is None:
                continue
            # Validar el número de tarjeta
            valido = self.validar_numero_tarjeta(num)
            if valido:
                card = ("American Express", num, ruta_archivo)
                tarjeta.append(card)
                num_ref[2] +=1
                self.contador3.setText(str(num_ref[2]))

    for num in numeros:
        emisor = self.obtener_emisor(num, t1, t2, t4, t5, t6)
        inn = self.obtener_INN(num, t1, t2, t4, t5, t6)
        if emisor is None:
            continue
        if inn is None:
            continue
        # Validar el número de tarjeta
        valido = self.validar_numero_tarjeta(num)
        if valido:
            tarjeta.append(self.crea_tablas(int(emisor), int(inn), num, ruta_archivo, num_ref))

    return tarjeta
```

*Figura 33. Codi de la funció buscar\_tarjetas*  
*Font. Elaboració pròpia*

```

def crea_tablas(self, emisor, inn, num, ruta_archivo, num_ref):
    if(emisor == 4):
        tarjeta = ("VISA", num, ruta_archivo)
        num_ref[0] +=1
        self.contador1.setText(str(num_ref[0]))
    elif (emisor == 2 or emisor == 5):
        tarjeta = ("Mastercard", num, ruta_archivo)
        num_ref[1] +=1
        self.contador2.setText(str(num_ref[1]))
    elif(emisor == 3 and inn == 35):
        tarjeta = ("JCB", num, ruta_archivo)
        num_ref[3] +=1
        self.contador4.setText(str(num_ref[3]))
    elif(emisor == 6 and inn == 62):
        tarjeta = ("UnionPay", num, (parameter) num_ref: Any
        num_ref[5] +=1
        self.contador6.setText(str(num_ref[5]))
    else:
        tarjeta = ("Discovery", num, ruta_archivo)
        num_ref[4] +=1
        self.contador5.setText(str(num_ref[4]))

    return tarjeta

```

Figura 34. Codi de la funció crea\_tablas

Font. Elaboració pròpia

## Funció action\_export

Tot i tenir definits els diferents mètodes d'exportació, es va requerir una reunió per definir el format en el qual es mostrava el contingut en cada cas. Primer, es va definir la funció "action\_export", que és cridada dins de la funció "buscar\_coincidencies" explicada anteriorment. En aquesta crida, rep com a paràmetres els booleans de PDF, EXCEL i WORD, juntament amb la llista de targetes trobades i la ruta on s'emmagatzemarà l'arxiu d'exportació.

Fent ús de les variables booleanes, s'han definit cinc casos que es comenten a continuació. En tots els casos, primerament es crida a la funció "action\_save\_as" per obtenir la ruta on s'ha de guardar l'arxiu a exportar. Segons si les variables booleanes PDF o WORD tenen un valor TRUE, es crida a la funció "export\_to\_pdf". Si la variable EXCEL té un valor TRUE, es crida a la funció "export\_to\_excel". En el cas que totes les variables estiguin amb valor TRUE, es criden a les dues funcions, com es pot apreciar a la figura 35.

```

def action_export(self, pdf, excel, word, tarjeta, ruta_inicio):
    # Add options to the submenu
    if(pdf and not excel and not word):
        ruta_p = self.action_save_as(ruta_inicio)
        self.export_to_pdf(tarjeta, ruta_p, pdf, word)
    elif(not pdf and excel and not word):
        ruta_p = self.action_save_as(ruta_inicio)
        self.export_to_excel(tarjeta, ruta_p)
    elif(not pdf and not excel and word):
        ruta_p = self.action_save_as(ruta_inicio)
        self.export_to_pdf(tarjeta, ruta_p, pdf, word)
    elif(pdf and not excel and word):
        ruta_p = self.action_save_as(ruta_inicio)
        self.export_to_pdf(tarjeta, ruta_p, pdf, word)
    else:
        ruta_p = self.action_save_as(ruta_inicio)
        self.export_to_pdf(tarjeta, ruta_p, pdf, word)
        self.export_to_excel(tarjeta, ruta_p)

```

*Figura 35. Codi de la funció action\_export*

*Font. Elaboració pròpia*

La funció "export\_to\_pdf" rep com a paràmetres la llista amb totes les targetes trobades, la ruta on s'ha de guardar l'arxiu i, finalment, les variables PDF i WORD. En primer lloc, es defineix la ruta de guardat, agafant la ruta passada com a paràmetre i afegint l'extensió .DOCX. Després, es crea un nou document de tipus DOCX i es recorre la llista de targetes mitjançant un bucle on, en cada iteració, s'afegeix l'element de la llista en format de llista per punts.

Un cop recorreguda tota la llista de targetes, es guarda el document amb la comanda ".save", tal com es pot veure a la figura 36. Finalment, fent ús de les variables booleanes, es revisa quina és la tipologia de document a exportar i, en cas que cap estigui actiu, s'esborra la ruta generada. Si la variable PDF té un valor TRUE, es crida a un script extern anomenat "trans.sh", que realitza el procés de transformació de WORD a PDF mitjançant la crida a sistema de subprocessos, i es mostra un missatge a mesura que s'ha realitzat correctament l'exportació. L'script es comenta en el següent apartat. En el cas que la variable WORD tingui un valor TRUE, simplement es mostra el missatge que l'element ha sigut exportat amb èxit.

```

def export_to_pdf(self, tarjeta, ruta_p, pdf, word):

    # Convert HTML to PDF using wkhtmltopdf
    ruta_pdf = ruta_p + ".docx"

    # Create a new document
    document = Document()

    for item in tarjeta:
        paragraph = document.add_paragraph(item)
        paragraph.style = 'List Bullet' # Apply the built-in "List Bullet" style

    # Save the document
    document.save(ruta_pdf)

    if pdf:
        subprocess.call(["/home/marcel/Desktop/TFG/trans.sh", ruta_pdf])
        QMessageBox.information(self, "Export", "The PDF file has been created successfully")

    if word:
        QMessageBox.information(self, "Export", "The Word file has been created successfully")
    else:
        os.remove(ruta_pdf)

```

*Figura 36. Codi de la funció export\_to\_pdf*  
*Font. Elaboració pròpia*

La funció "export\_to\_excel" rep com a paràmetres la llista amb totes les targetes i la ruta on es guardarà l'arxiu a exportar. En primer lloc, es crea un document Excel, un Workbook, i s'activa. A continuació, es defineix la ruta a exportar afegint l'extensió ".xlsx" i seguidament s'inicia un bucle que recorre tota la llista de targetes, afegint-ne una per una en una cel·la diferent de la primera columna. Finalment, es guarda el document i es mostra un missatge d'exportació, com es mostra a la figura 37.

```

def export_to_excel(self, tarjeta, ruta_p):

    # Create a workbook
    wb = Workbook()
    ws = wb.active

    ruta_excel = ruta_p + ".xlsx"

    # Write list items to cells (starting from A1)
    for i, item in enumerate(tarjeta):
        item_string = ', '.join(item) # Join elements with comma separator
        ws.cell(row=i+1, column=1).value = item_string

    # Save the workbook
    wb.save(ruta_excel)

    # Mostrar un mensaje de confirmación
    QMessageBox.information(self, "Exportn", "The Excel file has been created successfully")

```

*Figura 37. Codi de la funció export\_to\_excel*  
*Font. Elaboració pròpia*

## Funció `action_back`

Aquesta funció permet tornar a la pantalla principal un cop ha finalitzat el procés de cerca de targetes, donant lloc a la possibilitat de realitzar una nova cerca. Per fer-ho, primer obté totes les pantalles obertes en l'aplicació per seguidament tancar la pantalla principal i poder-la tornar a obrir, com podem observar a la figura 38.

```
def action_back(self):  
  
    open_windows = QApplication.instance().topLevelWidgets()  
  
    # Close all open windows  
    for window in open_windows:  
        if isinstance(window, QMainWindow):  
            window.close()  
  
    new_window = MainWindow()  
    new_window.show()
```

Figura 38. Codi de la funció `action_back`  
Font. Elaboració pròpia

## Funció `is_junction`

La funció «`is_junction`» comprova la tipologia d'arxius que s'està obrint per poder classificar-los entre arxius de text o enllaços. Com es pot veure a la figura 39, primer es diferencia en funció del sistema operatiu, «Linux» o «Windows». En el cas que el sistema operatiu sigui «Linux» es comprova que l'arxiu que es passa com a paràmetre, no sigui un softlink o hardlink. En cas de ser-ho es retorna «true» o «false» en cas contrari. En el cas que el sistema operatiu no sigui cap dels mencionats anteriorment es retornarà sempre «false» i no es realitzarà cap tractament. Per últim, en el cas que el sistema operatiu sigui «Windows», es comprova sí la ruta passada com a paràmetre és un enllaç a un altre arxiu, retornant directament el resultat de la comprovació.

```
def is_junction(self, ruta_archivo):  
    if platform.system() == "Linux":  
        if os.path.islink(ruta_archivo):  
            return True  
        else:  
            return False  
    elif platform.system() != "Windows":  
        return False  
    else:  
        return ctypes.windll.kernel32.GetFileAttributesW(ruta_archivo) & 0x400
```

Figura 39. Codi de la funció `is_junction`  
Font. Elaboració pròpia

## Arxiu trans.sh

Dins de les diferents millores realitzades s'inclou la creació d'un arxiu tipus script per poder realitzar la transformació d'un document Word en PDF. Aquest arxiu s'ha generat pel fet que les diferents implementacions que es podien fer dins del codi no han resultat efectives, sigui per problemes amb els imports o per no generar el resultat esperat.

En aquest arxiu, inicialment es declara una petita funció per gestionar els errors, i s'obté el primer element de la crida de sistema, que fa referència a la ruta de l'arxiu. Seguidament, es comprova si l'arxiu ja existeix. En cas afirmatiu, es mostra un missatge d'error i s'atura l'execució de l'script. Si l'arxiu no existeix, s'extreu la ruta de l'arxiu, eliminant l'extensió ".docx", i es crea una nova ruta afegint l'extensió ".pdf".

Finalment, es crida a LibreOffice amb una sèrie de flags per a generar la conversió de Word a PDF. En cas que la conversió sigui exitosa, es mostra un missatge indicant la correcta conversió i la ruta on s'ha guardat el fitxer PDF. En cas contrari, es mostra un missatge d'error, tal com es pot observar a la figura 40.

```
#!/bin/bash

# Function to show error message and exit
error_message() {
    echo "Error: $1"
    exit 1
}

# Obtain the file path from the first argument
file_path="$1"

# Verify if the file exists
if [ ! -f "$file_path" ]; then
    error_message "The file '$file_path' doesn't exist."
fi

# Extract the file name without the extension
filename="${file_path%.*}"

# Ruta y nombre del archivo PDF de salida
output_path="$filename.pdf"

# Command to convert the file to PDF
libreoffice --headless --convert-to pdf:writer_pdf_Export "$file_path" --outdir "$output_path"

# Verify if the conversion was successful
if [ $? -eq 0 ]; then
    echo "File converted successfully: $output_path"
else
    echo "Error while converting the file."
fi
```

*Figura 40. Codi del script trans.sh*

*Font. Elaboració pròpia*