

Projecte fi de grau

Estudi: Grau en Enginyeria Informàtica

Títol: Desenvolupament d'una aplicació mòbil de codi obert per a fer pagaments entre usuaris mitjançant codis QR

Document: Memòria

Alumna: Sudan Wu

Tutor: Pau Xiberta i Armengol

Departament: Informàtica, Matemàtica Aplicada i Estadística

Àrea: Llenguatges i Sistemes Informàtics

Convocatòria: Setembre de 2024

PROJECTE FI DE GRAU

Desenvolupament d'una aplicació mòbil de codi obert per a fer pagaments entre usuaris mitjançant codis QR

Autora:
SUDAN WU

Setembre 2024

Grau en Enginyeria Informàtica

Tutor:
PAU XIBERTA I ARMENGOL

Resum

Aquest projecte té com a objectiu desenvolupar una aplicació innovadora que permeti realitzar transaccions bancàries de manera ràpida, segura i flexible mitjançant codis QR. Inspirat en la meva experiència a la Xina, on les aplicacions de pagament integrades en plataformes com WeChat són omnipresents, aquest sistema pretén simplificar les operacions bancàries substituint les màquines TPV tradicionals. D'aquesta manera, s'elimina la necessitat d'invertir en hardware addicional i es redueixen els costos associats amb la infraestructura de pagament. Aquesta solució no només simplifica el procés de pagament, sinó que també ofereix una alternativa moderna i tecnològicament avançada a les màquines TPV (Terminal Punt de Venda) tradicionals, contribuint així a una experiència de pagament més fluïda i sense interrupcions.

A llarg termini, el sistema té el potencial d'evolucionar, ja sigui com a funcionalitat incorporada a les aplicacions bancàries existents o com a plataforma de pagament independent similar a PayPal. Les funcionalitats futures podrien incloure un sistema d'anàlisi de transaccions, pagaments recurrents, millores visuals, notificacions per veu... El projecte posa un èmfasi especial en la seguretat, amb l'objectiu de millorar la protecció de les dades utilitzant tecnologies d'encriptació. **Compte aquí: s'han fet servir tècniques avançades?**corregit

Aquest sistema està dissenyat per ser escalable, adaptable tant a petits comerços com a grans empreses, i preparat per a un futur on els pagaments digitals continuïn creixent en importància.

Agraïments

Per començar vull agrair molt especialment al meu tutor Pau Xiberta i Armengol, per la seva guia, paciència i saviesa durant tot el procés. Les seves orientacions han estat fonamentals per a la consecució d'aquest treball, i la seva dedicació ha estat una font constant de motivació.

També voldria agrair als meus companys i amics per estar sempre al meu costat, oferint suport emocional i consells valuosos en moments de dubte i incertesa. La vostra companyia ha fet aquest viatge molt més suportable i enriquidor.

No puc oblidar la meva família, que ha estat el meu pilar més ferm al llarg de tota aquesta etapa acadèmica. Gràcies per creure en mi, per la vostra paciència infinita i per animar-me a seguir endavant, fins i tot en els moments més difícils.

Finalment, vull reconèixer l'esforç i dedicació de tots els professors que m'han format durant aquests anys. Les vostres ensenyances han estat fonamentals per arribar fins aquí, i cada una de les vostres lliçons ha deixat una petjada en aquest treball.

A tots, gràcies de tot cor per fer possible aquest assoliment.

Índex

1	Introducció	1
1.1	Motivació	1
1.2	Finalitat i objectius del projecte	2
2	Estudi de viabilitat	3
2.1	Introducció	3
2.2	Viabilitat tècnica	3
2.3	Viabilitat econòmica	4
2.4	Viabilitat de mercat	5
3	Metodologia	7
3.1	Introducció	7
3.2	Scrum	7
3.3	Adaptació del scrum	9
4	Planificació	11
4.1	Introducció	11
4.2	Desenvolupament	11
5	Marc de treball i conceptes previs	13
5.1	Introducció	13
5.1.1	Propòsit	13
5.1.2	Conceptes clau	13
5.1.3	Codi QR	15
6	Requisits del sistema	17
6.1	Introducció	17
6.1.1	Els requisits funcionals	17
6.1.2	Els requisits no funcionals	19
6.1.3	Matriu de dependències de requisits	19
7	Estudi i decisions	21
7.1	Introducció	21
7.2	Maquinari	21
7.3	Front-end	21
7.3.1	Tecnologies més conegudes	22
7.3.2	Tecnologia Escollida: Flutter	23
7.3.3	IDE Escollida: Android Studio	24

7.4	Back-end	25
7.4.1	Tecnologies més conegudes	25
7.4.2	Tecnologia Escollida: GraphQL	27
7.4.3	Codi obert per implementar servidors GraphQL: Apollo Server	27
7.4.4	Llenguatge Escollit: TS	28
7.5	Base de dades	28
7.5.1	Proveïdors Populars de Bases de Dades en el Núvol	29
7.5.2	Base de Dades Escollida: MongoDB Atlas	30
7.6	Control de Versions	31
7.6.1	Control de Versions Escollida: GitHub	31
8	Anàlisi i disseny del sistema	33
8.1	Introducció	33
8.2	Diagrama de casos d'ús	33
8.2.1	Diagrama de casos d'ús de l'usuari client	33
8.2.2	Diagrama de casos d'ús de l'usuari administrador	35
8.2.3	Diagrama de casos d'ús del sistema	37
8.3	Fitxes de cas d'ús	37
8.3.1	Generar codi QR de cobrament	38
8.3.2	Generar codi QR de pagament	39
8.3.3	Escanejar un codi QR de pagament	40
8.3.4	Escanejar un codi QR de cobrament	41
8.3.5	Administrador modificar l'estat d'un compte bancari	42
8.4	Diagrama de la base de dades	43
8.4.1	User	44
8.4.2	Account	44
8.4.3	Dictionary	46
8.4.4	Transaction	47
8.5	Diagrama de processos	47
8.5.1	Descripció del procés	48
8.6	Interfícies d'usuari	49
8.6.1	Client	50
8.6.2	Administrador	63
9	Implementació i proves	73
9.1	Introducció	73
9.2	Idea inicial (potser millor “Idea inicial”)	73
9.3	Biblioteques utilitzades	74
9.3.1	Generar codi QR	74
9.3.2	Llegir codi QR	75

9.3.3 Encriptar Missatge	75
9.3.4 Encriptar la contrasenya	75
9.3.5 Gestionar les credencials	76
9.4 Desenvolupament d'aplicació	76
9.5 Problemes i Solucions en la Implementació	77
9.5.1 Android Studio	78
9.5.2 Flutter	79
9.5.3 GitHub	80
9.5.4 MongoDB	81
9.6 Proves	82
9.6.1 Proves d'unitat	82
9.6.2 Proves d'integració	83
9.6.3 Proves de regressió	84
9.6.4 Proves de sistema	84
9.6.5 Altres proves	84
10 Implantació i resultats	87
10.1 Implantació	87
10.1.1 Opcions per a la Implantació Futura	87
10.1.2 Mesures de Protecció de Dades i Compliment Normatiu	88
10.2 Resultats	89
10.2.1 La pàgina principal de l'usuari client	89
10.2.2 La pàgina principal de l'usuari administrador	91
11 Conclusions	93
12 Treball futur	95
12.1 Introducció	95
12.2 Funcionalitats de l'aplicació	95
12.3 Seguretat de dades	96
12.4 Millora del codi	96
13 Bibliografia	97
A Pressupost	99
B Planificació	103

Índex de figures

2.1	Un codi QR de PayPal.	6
3.1	Sistema de treball Scrum.	8
4.1	Diagrama de Gantt	12
5.1	Imatge d'un codi QR.	15
6.1	Matriu de dependències de requisits funcionals	20
8.1	Diagrama de casos d'ús de context de client.	34
8.2	Diagrama de casos d'ús de context d'administrador. “Llitar” hauria de ser “Llistar”, i “Cambiar” hauria de ser “Canviar”.	36
8.3	Diagrama de casos d'ús de context del sistema. “bancàries” hauria de ser “bancaris”.	37
8.4	Generar codi QR de cobrament	38
8.5	Generar codi QR de pagament	39
8.6	Generar codi QR de cobrament	40
8.7	Escanejar un codi QR de cobrament	41
8.8	Administrador modificar l'estat d'un compte bancari	42
8.9	Diagrama de la base de dades	43
8.10	Ús de Dictionary. Canviar “informaciones” per “informacions”, “inclouent missatge” per “incloent missatge”, “en DICTIONARY” per “a la taula Dictionary”, “a DICTIONARY per trobat” per “a la taula Dictionary per trobar”, i “a ACCOUNTS” per “a la taula Account”.	46
8.11	Diagrama de processos.	48
8.12	Primeres pàgines	50
8.13	Registrar nou usuari client	51
8.14	Pàgina de <i>login</i> - Client.	52
8.15	Escanejar codi QR	53
8.16	Generar un codi QR de pagament.	54
8.17	Pàgina per generar un codi QR de cobrament.	55
8.18	Pàgina per generar un codi QR de cobrament.	56
8.19	Modificar l'estat del compte bancari	57
8.20	Modificar l'estat del compte bancari	58
8.21	Afegir un compte bancari per a l'usuari	59
8.22	Refrescar la pàgina	60

8.23 Sortir de la secció	61
8.24 Altres gestions de l'usuari	62
8.25 Pàgina de <i>login</i> - Administrador	63
8.26 Llistar els comptes dels clients i modificar-ne l'estat	64
8.27 Crear una nova contrasenya per client	65
8.28 Modificar l'estat de l'usuari client	66
8.29 Eliminar un usuari client	67
8.30 Visualitzar els moviments d'un usuari	68
8.31 Registrar un nou usuari	69
8.32 Llistar tots els administradors i veure els seus moviments	70
8.33 Canviar la pròpia contrasenya	71
9.1 Versió antiga vs. versió nova	74
9.2 Problemes d'implementació amb Android Studio	78
9.3 Pàgina de prova	79
9.4 Comanda flutter doctor	80
9.5 Eliminar la taula de mongoDB	82
9.6 Les contrasenyes codificades	83
9.7 Resolució d'una transferència QR	83
9.8 Xifratge del codi QR de pagament	84
10.1 La pàgina principal de l'usuari client	90
10.2 La pàgina principal de l'usuari client	92

Índex de taules

- 2.1 Cost estimat del desenvolupament de l'aplicació amb hores multiplicades per 2 i preu/hora de 15€ 4

CAPÍTOL 1

Introducció

Si ho prefereixes, pots anomenar el capítol només com a “Introducció”, i fer que la secció “Motivació” sigui la primera (1.1), deixant el text actual de la 1.1 sense títol.corregit

A l’hora de processar un pagament, la majoria de botigues utilitzen un Terminal Punt de Venda (TPV) físic, i els clients fan servir una targeta bancària o una cartera digital que incorpora aquesta targeta. El TPV actua com a intermediari en l’operació, introduint un procés addicional que pot augmentar tant els costos com el risc d’errors. Tot i que existeixen plataformes de pagament en línia, aquestes no estan dissenyades per substituir completament el TPV físic, i sovint no són de codi obert ni ofereixen l’opció de realitzar transaccions mitjançant codis QR.

1.1 Motivació

L’evolució tecnològica i la digitalització han transformat profundament la manera com les empreses gestionen els seus processos de pagament. Tot i els avenços, el sistema tradicional basat en terminals físics de punt de venda (TPV) encara predomina, arrossegant amb si costos operatius elevats i riscos associats a errors humans i frauds. Aquesta situació, sumada a la creixent adopció de tecnologies de pagament mòbil i l’ús de codis QR en diverses parts del món, posa de manifest la necessitat d’explorar alternatives més eficients, segures i accessibles.

Aquest projecte neix de la voluntat d’abordar aquestes limitacions, oferint una solució que pugui substituir o complementar els TPV físics amb una plataforma de pagament en línia de codi obert. Una solució que permeti als comerciants processar transaccions mitjançant codis QR, oferint així una experiència de pagament més ràpida, senzilla i segura tant per a les botigues com per als clients.

La motivació principal d’aquest projecte és reduir els costos operatius i minimitzar els riscos associats als sistemes de pagament tradicionals, alhora que es promou la inclusió tecnològica a través d’una eina de codi obert, accessible per a qualsevol empresa o desenvolupador que desitgi implementar-la. Creiem que, mitjançant aquesta iniciativa, podem contribuir a la transformació digital

del comerç, afavorint la seva competitivitat i adaptabilitat en un mercat cada cop més globalitzat i dinàmic.

1.2 Finalitat i objectius del projecte

L'objectiu d'aquest treball de final de grau és desenvolupar una aplicació *full stack* que inclogui totes les funcions necessàries per gestionar pagaments mitjançant codis QR. Aquest desenvolupament inclourà:

- El disseny i la implementació de la base de dades per emmagatzemar la informació de les transaccions i els usuaris.
- El desenvolupament d'una API que permeti interactuar de manera eficient amb la base de dades, facilitant la creació, gestió i consulta de pagaments.
- La creació d'una interfície d'usuari senzilla i intuïtiva, que permeti als usuaris generar i escanejar codis QR per realitzar transaccions.
- El desenvolupament de les aplicacions frontals basades en aplicacions que permetin als usuaris accedir al sistema, generar codis QR i realitzar els pagaments de manera segura i ràpida.

L'aplicació permetrà als clients generar codis QR tant per realitzar pagaments com per rebre cobraments, que podran ser escanejats amb qualsevol dispositiu que tingui càmera per completar la transacció.

CAPÍTOL 2

Estudi de viabilitat

2.1 Introducció

L'estudi de viabilitat és una tècnica àmpliament coneguda i utilitzada en diversos camps, com ara la gestió de projectes, la planificació empresarial i el desenvolupament de noves tecnologies. Aquesta tècnica s'utilitza per avaluar si un projecte o producte és factible des de diferents perspectives, incloent-hi la tècnica, econòmica i mercat. **Revisar la frase: és una tècnica coneguda? corregit**

En conjunt demostra que el projecte és viable i té un alt potencial de creixement en l'entorn actual, oferint una solució innovadora i segura per als pagaments mòbils a través de codis QR.

2.2 Viabilitat tècnica

Per assegurar la viabilitat tècnica de l'aplicació de pagaments mitjançant codis QR, és essencial considerar la disponibilitat de tecnologies que permetin desenvolupar tant el *front-end* com el *back-end* de l'aplicació. Existeixen tecnologies àmpliament utilitzades per a aquest tipus d'aplicacions, que inclouen solucions per a la generació i lectura de codis QR, així com plataformes de desenvolupament per a dispositius mòbils.

A més, és important assegurar-se que l'aplicació pugui operar en una infraestructura robusta i segura, i que el desenvolupament es pugui dur a terme amb les competències tècniques disponibles dins l'equip. Tal com es detallarà al Capítol 7, es farà una selecció de les eines i tecnologies més adequades per garantir la viabilitat tècnica del projecte, les quals seran avaluades amb detall per assegurar que compleixin amb els requisits del sistema i els objectius del projecte.

En teoria aquí encara no has decidit quines eines utilitzaràs; ho faràs al Capítol 7. Pots fer dues coses: (1) podries dir igualment que hi ha tecnologies àmpliament utilitzades, així en general, per implementar aplicacions d'aquest tipus, incloent *front-end* i *back-end*, o bé (2) podries dir que, tal com es veurà al

Capítol 7, s'ha decidit utilitzar aquestes eines (abans de parlar d'elles i dir que són viables).corregit

2.3 Viabilitat econòmica

Els costos principals del projecte inclouen el desenvolupament de l'aplicació, que es realitzarà internament, així com els costos d'allotjament en el núvol, proporcionats per serveis com AWS, amb un pressupost mensual o anual.

Per calcular el cost de desenvolupament de l'aplicació, es tindran en compte les hores de desenvolupament multiplicades pel preu per hora d'un programador. El preu base d'un programador júnior és de 15 €/hora.

Fase de Desenvolupament	Hores estimades	Cost (€)
Anàlisi de Requisits	80 h	1.200 €
Disseny de l'Aplicació	120 h	1.800 €
Desenvolupament del front-end	240 h	3.600 €
Desenvolupament del back-end	300 h	4.500 €
Proves i Depuració	100 h	1.500 €
Implementació i Lliurament	60 h	900 €
Total	900 h	13.500 €

Taula 2.1: Cost estimat del desenvolupament de l'aplicació amb hores multiplicades per 2 i preu/hora de 15€

El cost estimat del desenvolupament de l'aplicació es mostra a la Taula 2.1. Aquesta taula detalla els costos associats a cada fase del projecte, amb un total de 13.500 € per al desenvolupament intern.

Tot i que el desenvolupament es realitzi internament, estaria bé calcular quant costaria. Bàsicament, hores de desenvolupament multiplicades pel preu/hora d'un programador.corregit

En algun lloc del text hauries de citar la figura, com per exemple “tal com es veu a la Figura ??, ...”. Recorda el mateix que abans, també, que en principi encara no has decidit si ho faràs amb MongoDB Atlas (pots fer com abans: “tal com es veurà al Capítol 7...”).corregit

Per determinar la viabilitat econòmica del projecte, és crucial estimar els ingressos potencials i calcular el temps necessari per amortitzar la inversió inicial. A continuació es presenten les estimacions basades en les dades actuals:

El cost inicial del projecte es compon de 13.500 € per al desenvolupament de l'aplicació i 5.000 € per al manteniment anual, sumant un total de 18.500 €.

Amb una previsió de 1.000 clients, cadascun amb un dipòsit de 1.000 €, el total de dipòsits seria de 1.000.000 €. Aquesta quantitat permetrà realitzar inversions que generaran interessos. Si assumim una taxa d'interès anual del 2%, els interessos generats seran de 20.000 € cada any.

Així, amb aquests guanys d'interessos, la recuperació dels 18.500 € invertits es podria aconseguir en aproximadament 1 anys. Aquest càlcul indica que, amb una base sòlida de clients i dipòsits, la inversió inicial podria ser amortitzada en un període relativament curt, mostrant així un bon potencial de rendibilitat per al projecte.

Aquí podria estar bé fer una estimació dels ingressos, i així poder dir quan es creu que pot amortitzar-se la inversió inicial, comptant el desenvolupament de l'aplicació i el seu manteniment.corregit

El sistema de pagament mitjançant codis QR ofereix una solució econòmica en comparació amb els terminals de TPV tradicionals, amb una possible font d'ingressos a través de comissions per transacció i opcions de subscripció per serveis addicionals.

2.4 Viabilitat de mercat

Amb l'augment de l'ús de pagaments mòbils i la necessitat de solucions de pagament *contactless*, hi ha una demanda creixent per a sistemes de pagament basats en codis QR. Aquesta tendència proporciona una oportunitat significativa per a la nostra aplicació.

Encara que hi ha altres solucions de pagament basades en codis QR, el nostre sistema es diferencia per la seva integració fàcil amb altres plataformes i la seva estructura de codi obert, que permet personalització i adaptació a les necessitats específiques dels usuaris.

Recorda citar la Figura 2.1; si no, no se sap d'on baixa... Per exemple, just quan dius que hi ha altres solucions basades en codis QR, pots posar entre parèntesis alguna cosa com “veure, per exemple, la Figura 2.1, on es mostra un codi QR de PayPal”.corregit

La imatge mostrada a la Figura 2.1 és un exemple d'un anunci de PayPal. Com es pot observar, l'anunci destaca la facilitat d'ús del sistema: només cal escanejar el codi QR, efectuar el pagament i continuar amb les activitats diàries.



Figura 2.1: Un codi QR de PayPal.

Aquest procés de pagament és extremadament còmode i simplificat per a l'usuari i comercials.

CAPÍTOL 3

Metodologia

3.1 Introducció

La metodologia utilitzada en aquest projecte és la metodologia àgil, en concret **Scrum**, tot i que es tracta d'un projecte individual.

Les metodologies àgils són un conjunt de principis i pràctiques de gestió de projectes dissenyades per a la flexibilitat i la rapidesa en el desenvolupament de productes. A diferència dels enfocaments tradicionals, que sovint són rígids i seqüencials, les metodologies àgils se centren en la iteració, la col·laboració i l'adaptabilitat. Ofereixen diversos avantatges respecte a les metodologies tradicionals:

- **Adaptabilitat als Canvis:** Permeten ajustar els requisits i prioritats al llarg del projecte, mentre que les metodologies tradicionals són més rígides i difícils de modificar un cop el desenvolupament està en marxa.
- **Desenvolupament Iteratiu:** Es treballa en iteracions curtes i incrementals, proporcionant resultats parcials i millorant contínuament el produpte. En canvi, les metodologies tradicionals solen seguir un enfocament seqüencial, amb un lliurament complet al final del projecte.
- **Reducció de Riscos:** La revisió contínua i les iteracions àgils ajuden a identificar i abordar problemes abans que es converteixin en grans obstacles, mentre que les metodologies tradicionals poden identificar riscos tardanament, quan ja poden ser més difícils de gestionar.

Pots posar una mica més de context: per exemple, explicar què són les metodologies àgils i quins avantatges tenen respecte metodologies més tradicionals. Pots fer-ho en una subsecció, si vols. corregit

3.2 Scrum

Scrum és un marc de treball per a la gestió de projectes, conegut per la seva flexibilitat i per permetre gestionar projectes complexos a través de cicles iteratius

de treball anomenats *sprints*. Encara que tradicionalment Scrum està dissenyat per a equips, es van adaptar els seus principis per optimitzar la gestió del temps i els recursos durant el desenvolupament del projecte.

L'elecció d'Scrum es va basar en la necessitat de mantenir una alta capacitat de resposta davant els canvis en els requisits i per estructurar el procés de desenvolupament en fases manejables. Scrum va permetre dividir el projecte en *sprints*, cosa que va facilitar la planificació i el seguiment del progrés, assegurant-se que cada fase del projecte es completava abans de passar a la següent.

Per a la planificació del projecte, es defineixen *sprints* de dues setmanes, cada un amb objectius específics que cal complir abans d'iniciar el següent *sprint*. Al principi de cada *sprint*, s'elabora un pla detallat de les tasques a realitzar, priorititzant les més crítiques i assegurant que estan alineades amb els objectius generals del projecte.

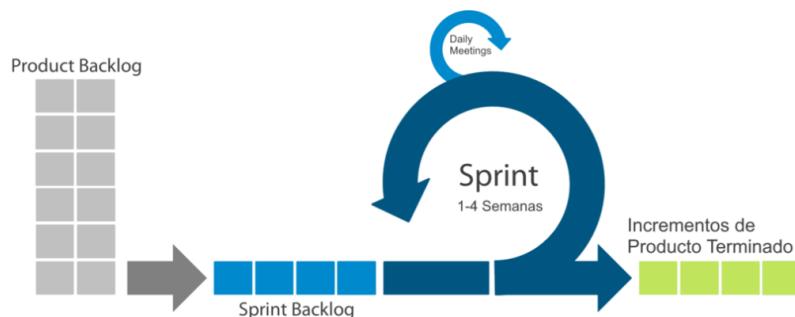


Figura 3.1: Sistema de treball Scrum.

Citar figura al text.[corregit](#)

Tal com es veu a la figura 3.1, es crea i es gestiona un *product backlog* que llista totes les funcionalitats a desenvolupar. Aquest *backlog* es prioritza d'acord amb la seva importància i la complexitat de la implementació. Cada *sprint* es planifica tenint en compte aquesta llista, garantint que es treballa sempre en les tasques més rellevants per al progrés del projecte.

L'ús d'Scrum, adaptat a un entorn individual, va ser fonamental per a la gestió eficient del temps i la consecució dels objectius del projecte. Va permetre que el desenvolupament es realitzés de manera ordenada i progressiva, proporcionant la flexibilitat necessària per fer ajustos sobre la marxa. Aquesta metodolo-

gia va resultar especialment útil per mantenir el ritme de treball i assegurar que cada fase del projecte es complets amb èxit abans de passar a la següent.

3.3 Adaptació del scrum

Un cop definida la idea principal de l'aplicació, el següent pas és la seva implementació, que es realitza en les següents fases:

- **Planificació:** Establir els requisits del projecte, crear un product backlog amb les funcionalitats prioritzaes i definir els objectius de cada sprint.
- **Disseny:** Desenvolupar l'arquitectura del sistema i el disseny de la interfície d'usuari, amb revisions contínues basades en el feedback rebut.
- **Desenvolupament:** Implementar les funcionalitats en sprints curts, completant tasques específiques i ajustant segons les necessitats identificades durant el procés.
- **Revisió i ajustament:** Realitzar revisió per avaluar el treball completat i ajustar les tasques futures. Utilitzar les retrospectives per millorar el procés de desenvolupament.
- **Entrega:** Lliurar els resultats parcials al final de cada sprint, permetent una validació progressiva i realitzant ajustaments abans de la finalització completa del projecte

Per al projecte actual, seguint la metodologia Scrum, primer es crea un backlog amb l'esquelet de codi que estableix la relació entre el client i el servidor. A partir d'aquí, s'afegeixen funcionalitats en els sprints successius. Si en algun moment cal fer modificacions, es inclouen en el següent backlog, avançant progressivament fins a l'entrega final.

En Scrum, el procés de desenvolupament permet realitzar canvis en qualsevol moment. Aquesta flexibilitat és una de les principals avantatges, ja que ens permet adaptar-nos de manera contínua als nous requisits. L'enfocament àgil facilita l'ajustament dels objectius i les funcionalitats del projecte al llarg del seu desenvolupament, assegurant que el producte final s'alineï amb les necessitats canviants i les expectatives dels clients. Aquesta capacitat d'adaptació contínua garanteix que el projecte pugui respondre eficaçment a les variacions del mercat i a les noves oportunitats que puguin sorgir.

Podria estar bé explicar una mica com s'ha adaptat aquesta metodologia al projecte actual, explicant-ne una mica les fases, per exemple (planificació, disseny, desenvolupament...). També podries fer-ho en una subsecció, si vols.corregit

CAPÍTOL 4

Planificació

4.1 Introducció

El següent pla de planificació ofereix una visió clara del procés de desenvolupament del projecte, destacant les etapes clau, el temps assignat i els resultats esperats, així com l'aplicació de la metodologia Scrum per gestionar el projecte de manera àgil.

4.2 Desenvolupament

El procés de creació i construcció del producte de programari inclou les següents tasques:

1. Anàlisi de requisits:

- Revisar els requisits del projecte.
- Establir especificacions detallades per al desenvolupament.

2. Disseny:

- Crear l'arquitectura del sistema.
- Dissenyar les interfícies d'usuari i les interaccions.
- Definir les estructures de dades i les lògiques d'operació.

3. Codificació:

- Escriure el codi font segons les especificacions.
- Implementar les funcionalitats del sistema.
- Desenvolupar codi de *back-end* i *front-end*. *Teòricament encara no has decidit les eines: jo potser diria directament “codi de back-end i front-end, sense especificar.* *corregit*

4. Proves:

- Crear i executar proves per verificar la funcionalitat de cada component.
- Corregir errors detectats durant les proves.

5. Revisió de codi i millores:

- Realitzar revisions de codi per assegurar la qualitat i el compliment dels estàndards.

6. Documentació:

- Elaborar memòria sobre l'arquitectura, el disseny i el codi.

7. Manteniment i actualitzacions

- Realitzar manteniment regular del sistema per assegurar la seva funcionalitat.
- Implementar actualitzacions, millores i nous requisits.

Tal com es veu al diagrama Gantt 4.1 proporciona una visió detallada del calendari del projecte, desglossant les diferents fases i tasques previstes. Aquesta eina visual ofereix una visió general clara del progrés del projecte, facilitant tant la gestió del temps com l'assignació de recursos. Mitjançant el diagrama, és possible identificar les dates clau i controlar l'avanç de les tasques, ajudant així a assegurar que el projecte es mantingui en el camí correcte per a una finalització exitosa.

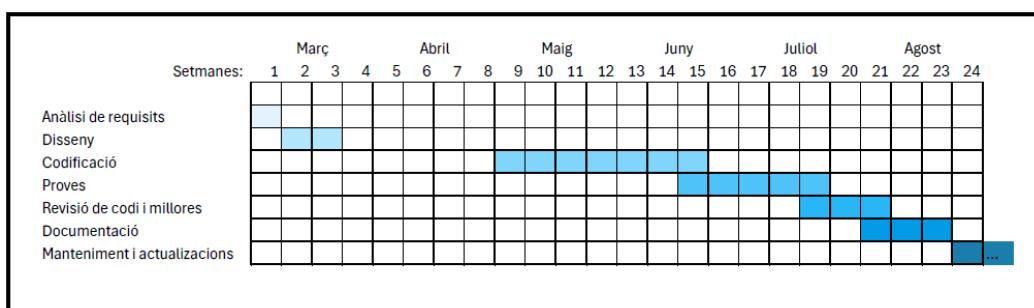


Figura 4.1: Diagrama de Gantt

Jo transformaria aquesta taula en un diagrama de Gantt, que és molt més visual (la columna “Resultats” no cal). Abans de posar el diagrama, introduceixo amb una mica de text.corregit

També faria que tot plegat fos una secció, i no una subsecció.corregit

CAPÍTOL 5

Marc de treball i conceptes previs

5.1 Introducció

En aquest capítol es tractaran els coneixements necessaris per desenvolupar correctament una aplicació bancària que utilitza codis QR per realitzar operacions. És important entendre com funciona el codi QR, especialment en relació amb com emmagatzemar el contingut. A més, cal disposar de coneixements sobre arquitectura i desenvolupament de programari per aportar valor a l'aplicació.

5.1.1 Propòsit

Potser afegiria “i conceptes clau” al títol de la subsecció.corregit

L'objectiu de l'aplicació és proporcionar una plataforma senzilla, escalable i potent per gestionar els pagaments i cobraments mitjançant codis QR. Tant els clients com els comerciants tindran accés a funcionalitats i permisos dins de l'aplicació, permetent-los configurar els seus sistemes de pagament o cobrament, establir opcions disponibles, ajustar els límits de transacció i altres configuracions relacionades. A més de facilitar les compres, l'aplicació permetrà als usuaris realitzar transferències entre amics i familiars, oferint així una solució versàtil per a diverses necessitats financeres quotidianes.

Per tant, en una sola aplicació, els usuaris podran controlar totes les transaccions que es realitzin, alhora que tindran la capacitat d'aplicar canvis i modificacions segons les seves necessitats i preferències.

5.1.2 Conceptes clau

A continuació, es presenta una llista amb les parts fonamentals que defineixen el funcionament de l'aplicació:

- **App:** L'aplicació es pot concebre com una entitat bancària en si mateixa, on els seus usuaris actuen com a clients que poden tant rebre com enviar pagaments.

- **Usuari Client:** L'usuari client és la persona que utilitza l'aplicació per realitzar pagaments o cobraments.
- **Usuari Administrador:** L'usuari administrador és qui gestiona l'aplicació, configurant els sistemes de pagament i supervisant les transaccions, comptes i també els usuaris.
- **Transacció:** Les transaccions es fan entre comptes bancaris, independentment dels usuaris. No es pot fer una transacció dins del mateix compte, però sí que poden ser entre comptes del mateix client.
- **QR de pagament:** Per a cada compte, només hi ha un codi QR actiu, ja que aquests estan xifrats per garantir la seguretat. Cada vegada que es genera un nou codi QR, la clau de xifratge s'actualitza. A més, els codis QR són d'ús únic i caduquen després d'un temps determinat. Tots aquests processos es duen a terme per reforçar la protecció.
- **QR de cobrament:** El codi QR de cobrament és més senzill que el de pagament, ja que presenta menys restriccions. A diferència del codi QR de pagament, el de cobrament no està xifrat ni caduca, i es pot utilitzar tantes vegades com es desitgi.
- **Activat/Desactivar Usuari client:** Un usuari pot canviar l'estat del seu compte per si mateix o mitjançant un administrador. Quan el usuari està desactivat, l'usuari no podrà realitzar cap acció ni visualitzar informació, encara que podrà rebre cobraments.
- **Activar/Desactivar Compte:** Un compte pot canviar el seu estat per si mateix o mitjançant un administrador. Quan el compte està desactivat, l'usuari no podrà generar codis QR de pagament ni fer pagaments de quansevol manera. Altres operacions seguiran igual.
- **Dictionary:** Cada cop que l'usuari genera un codi QR, tota la informació es guarda en la taula *Diccionary*. Quan el codi és escanejat, el sistema consulta internament aquest *Diccionary* per obtenir les dades necessàries, com el compte d'origen i l'import.
- **Codi QR:** El codi QR és un element clau de l'aplicació, utilitzat per comunicar informació entre clients.

Pots pensar altres conceptes clau com ara “transacció”, “QR de pagament”, “QR de cobrament”...corregit

5.1.3 Codi QR

Els codis QR (Quick Response) van ser desenvolupats al Japó a mitjans dels anys 90. És un tipus de codi de barres bidimensional que pot emmagatzemar una gran quantitat d'informació en un espai reduït. Aquest codi es pot escanejar fàcilment amb la càmera d'un telèfon mòbil o amb dispositius específics, permetent accedir a la informació que conté de manera ràpida i senzilla.

Tal com veu a la figura 5.1 mostra un codi QR. Es pot escanejar amb la càmera del mòbil i conté text emmagatzemat en el seu interior. El codi QR té tres quadrats grans en les cantonades, que ajuden a posicionar-lo correctament. La resta dels quadrats són en general de dos colors: negre (1) i blanc (0). Aquests colors formen un patró que representa informació en format binari, que posteriorment es converteix en text.



Figura 5.1: Imatge d'un codi QR.

Els codis QR s'han popularitzat per la seva versatilitat i eficiència. En aplicacions de pagaments i cobraments, els codis QR ofereixen una manera ràpida i segura de transferir informació, com ara detalls de pagament o d'una transacció específica. L'usuari simplement escaneja el codi amb el seu dispositiu, i la informació es transmet instantàniament, eliminant la necessitat d'introduir dades manualment, reduint errors i accelerant el procés de transacció. A més, aquesta manera de fer-ho protegeix les dades personals, ja que evita la necessitat de mostrar informació sensible a altres persones.

CAPÍTOL 6

Requisits del sistema

6.1 Introducció

Aquest capítol tractarà l'especificació de requisits del sistema per al programari que s'està desenvolupant. Es fixaran les bases del que es desenvoluparà, tenint en compte les necessitats dels usuaris, així com els requisits funcionals i no funcionals.

Els requisits funcionals es defineixen com les funcionalitats bàsiques del sistema. Aquests requisits especificen què ha de fer el sistema, descrivint les accions que ha de ser capaç de realitzar i les operacions que ha de suportar. Tots els requisits exposats en aquesta secció es consideren essencials i el sistema es consideraria incomplet si no complís aquests requisits.

Els requisits no funcionals, d'altra banda, defineixen les qualitats o atributs del sistema, com ara el seu rendiment, seguretat, escalabilitat i usabilitat.

6.1.1 Els requisits funcionals

La funcionalitat bàsica de l'aplicació que es presenta és fer transferències de diners entre dues usuaris. Seria una funció semblant al que permet la tecnologia “Bizum”, però incorporant el servei de pagament/cobrament mitjançant codis QR.

En l'apartat de bibliografia es detallen amb més precisió les informacions sobre BIZUM.

- <https://bizum.es/>

Aquí és una bona oportunitat per afegir un ítem a la bibliografia relacionat amb Bizum. Alguna cosa així??

Gestió d'usuaris clients:

- **RF-1.** Els usuaris clients han de poder registrar-se amb informació bàsica.
- **RF-2.** Els usuaris clients han de poder iniciar sessió a l'aplicació utilitzant les seves credencials.

- **RF-3.** Els usuaris clients han de poder modificar la contrasenya d'accés, proporcionant la contrasenya antiga.
- **RF-4.** Els usuaris clients han de poder donar-se de baixa ells mateixos.
- **RF-5.** Els usuaris clients han de poder configurar els seus comptes, incloent l'estat, la descripció i el límit màxim per transaccions, entre altres opcions.

Gestió d'usuaris administradors:

- **RF-6.** Els usuaris administradors han de poder registrar-se amb informació bàsica, però caldrà disposar de les credencials d'un altre administrador.
- **RF-7.** Els usuaris administradors han de poder iniciar sessió a l'aplicació utilitzant les seves credencials.
- **RF-8.** Els usuaris administradors han de poder modificar la contrasenya d'accés, proporcionant la seva contrasenya antiga.
- **RF-9.** Els usuaris administradors han de poder donar una nova contrasenya d'accés a l'usuari client, sense que calgui proporcionar la contrasenya antiga.
- **RF-10.** Els usuaris administradors han de poder donar-se de baixa ells mateixos (excepte l'usuari “admin”).
- **RF-11.** Els usuaris administradors han de poder visualitzar tots els usuaris clients i tots els moviments associats als seus comptes.
- **RF-12.** Els usuaris administradors han de poder veure tots els usuaris administradors i tots els moviments associats a cadascun d'ells.
- **RF-13.** Els usuaris administradors han de poder gestionar tant els usuaris clients com els seus comptes.

Operacions de transaccions d'usuari client:

- **RF-14.** Els usuaris clients han de poder generar un codi QR únic associat al seu compte per rebre pagaments i cobraments, amb o sense un preu definit.
- **RF-15.** Els usuaris clients han de poder escanejar codis QR generats per altres usuaris per fer pagaments i cobraments, introduint el preu si no està definit prèviament.
- **RF-16.** Els usuaris poden fer transaccions mitjançant l'entrada manual del número de compte del destinatari, en cas que la càmera no funcioni.

Funcionalitats addicionals:

- **RF-17.** S'ha de generar un *log* que registri tots els moviments tant de l'usuari client com de l'usuari administrador.

Integracions i serveis complementaris:

- **RF-18.** S'ha de poder canviar el color de fons de l'aplicació.

6.1.2 Els requisits no funcionals

Els requisits no funcionals són:

- **RNF-1.** L'aplicació ha de ser ràpida i eficient, amb temps de càrrega mínims i una resposta àgil a les interaccions de l'usuari.
- **RNF-2.** S'han d'implementar mesures de seguretat robustes, com l'autenticació i el xifrat de dades, per protegir la informació personal i finançera dels usuaris. **Això seria requisit no funcional!**corregit
- **RNF-3.** L'aplicació ha de poder gestionar un alt volum de transaccions i usuaris concurrents, amb la capacitat d'escalar horitzontalment segons sigui necessari.
- **RNF-4.** La interfície d'usuari ha de ser intuïtiva i fàcil d'utilitzar, amb un disseny net i clar que permeti als usuaris realitzar les seves tasques de manera eficient.
- **RNF-5.** L'aplicació ha de ser compatible amb una varietat de dispositius i sistemes operatius, incloent telèfons mòbils, tauletes. **Per a ordinadors també?**corregit
- **RNF-6.** S'han d'implementar pràctiques de seguretat sòlides per protegir la integritat i la confidencialitat de les dades dels usuaris, així com per prevenir frau i atacs cibernetícs.

6.1.3 Matriu de dependències de requisits

Seguint la taula 6.1 que il·lustra les relacions entre els diferents requisits del projecte. Aquesta matriu és fonamental per entendre com cada requisit es connecta i depèn d'altres, facilitant així la gestió de canvis i l'assegurament de la coherència durant tot el cicle de vida del projecte.

A través d'aquesta matriu, es poden identificar clarament les interdependències, cosa que permet una millor planificació i implementació dels requisits:

- Els requisits RF_1 i RF_6 són operacions de registre.
- Els requisits RF_3, RF_4, RF_5, RF_14, RF_15 i RF_16 són operacions de l'usuari client; per dur-les a terme, cal que l'usuari client estigui identificat.
- Els requisits RF_8, RF_9, RF_10, RF_11, RF_12 i RF_13 són operacions de l'usuari administrador; per dur-les a terme, cal que l'usuari administrador estigui identificat.
- Finalment, els requisits RF_17 i RF_18 són requisits complementaris, i requereixen que tant el client com l'administrador estiguin autenticats.

Aquí quedaria molt millor una matriu de veritat. Exemple ràpid trobat a internet: <http://wiki.doing-projects.org/images/4/4b/DSM.PNG>. Llavors pots complementar la matriu amb la taula que tens, on expliques la descripció de les dependències.corregit

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
RF- 1																		
RF- 2	X																	
RF- 3			X															
RF- 4		X																
RF- 5			X															
RF- 6																		
RF- 7							X											
RF- 8								X										
RF- 9									X									
RF- 10										X								
RF- 11											X							
RF- 12											X							
RF- 13												X						
RF- 14		X																
RF- 15			X															
RF- 16				X														
RF- 17					X													
RF- 18						X												

Figura 6.1: Matriu de dependències de requisits funcionals

CAPÍTOL 7

Estudi i decisions

[Tornat a fer](#)

7.1 Introducció

En aquest capítol es presenten els diferents aspectes que s'han considerat per escollir les tecnologies utilitzades en el desenvolupament del sistema. Les decisions sobre el maquinari, les llibreries i el programari s'han pres tenint en compte criteris de rendiment, compatibilitat, facilitat d'ús i mantenibilitat a llarg termini. Potser millor “... per escollir les tecnologies utilitzades per desenvolupar el sistema”.corregit

7.2 Maquinari

Per al desenvolupament del projecte, s'ha utilitzat un portàtil Lenovo ThinkPad amb les següents especificacions:

- Processador: Intel Core i7
- Memòria RAM: 16 GB DDR4
- Disc dur: SSD de 512 GB
- Sistema operatiu: Windows 10

Aquest maquinari s'ha seleccionat per la seva fiabilitat i capacitat de processament, que són essencials per gestionar l'execució simultània de diverses eines de desenvolupament.

7.3 Front-end

El *front-end* es refereix a la part del desenvolupament d'una aplicació o lloc web que interactua directament amb els usuaris. En termes simples, és tot el que els usuaris veuen i amb el que interactuen en una aplicació. Inclou el disseny i

la implementació de la interfície d'usuari (UI), que comprèn elements com ara botons, menús, pàgines i imatges.

Per al desenvolupament de *front-end* d'aplicacions mòbils, hi ha diverses tecnologies i eines que poden ser utilitzades, depenent de les necessitats específiques del projecte, com la plataforma objectiu, els requisits d'usuari i el rendiment desitjat.

7.3.1 Tecnologies més conegudes

A continuació, es detallen algunes de les tecnologies més conegudes i populars per al desenvolupament de front-end en aplicacions mòbils:

- **Flutter**

- **Descripció:** Flutter és un framework creat per Google per al desenvolupament d'interfícies d'usuari. Permet crear aplicacions mòbils nadiues per a iOS i Android utilitzant un sol codi font, escrit en el llenguatge Dart. Flutter proporciona una gran varietat de widgets personalitzables que faciliten el disseny d'interfícies d'usuari atractives i de rendiment alt.
- **Avantatges:** Disseny consistent entre plataformes, rendiment elevat gràcies a la compilació a codi nadiu, i un ampli conjunt de widgets personalitzables.

- **React Native**

- **Descripció:** React Native és un framework creat per Facebook que permet el desenvolupament d'aplicacions mòbils utilitzant JavaScript i React. Ofereix components nadius que es poden utilitzar per crear aplicacions per a ambdues plataformes, iOS i Android, amb un codi base compartit.
- **Avantatges:** Compartició de codi entre plataformes, gran comunitat i ecosistema amb moltes biblioteques i recursos, i capacitat d'integrar-se amb codi nadiu quan sigui necessari.

- **Xamarin**

- **Descripció:** Xamarin és una plataforma de Microsoft per al desenvolupament d'aplicacions mòbils utilitzant C# i .NET. Permet als desenvolupadors crear aplicacions nadiues per a iOS i Android amb un sol codi font, compartint gran part del codi entre plataformes.

- **Avantatges:** Integració amb l'ecosistema .NET i Visual Studio, reutilització significativa del codi entre plataformes, i accés a les API nadiues i funcions específiques del dispositiu.
- **Ionic**
 - **Descripció:** Ionic és un framework basat en HTML5 per al desenvolupament d'aplicacions mòbils híbrides. Utilitza tecnologies web com HTML, CSS i JavaScript, i es basa en Angular o Vue.js per gestionar l'arquitectura de l'aplicació. Ionic permet crear aplicacions que es poden executar en qualsevol dispositiu mitjançant un navegador web o com a aplicacions híbrides mitjançant Cordova o Capacitor.
 - **Avantatges:** Desenvolupament ràpid amb tecnologies web familiars, capacitat d'exportar a diverses plataformes amb un sol codi base, i gran biblioteca de components d'interfície d'usuari.

7.3.2 Tecnologia Escollida: Flutter

En aquest projecte, s'ha escollit **Flutter** com a framework per al desenvolupament del *front-end* de l'aplicació mòbil. A continuació, es detallen les raons per les quals Flutter ha estat seleccionat, així com els avantatges clau que ofereix aquest framework, especialment en comparació amb altres opcions comunes:

- **Rendiment Elevat:** Flutter ofereix un rendiment destacat gràcies a la seva arquitectura basada en la compilació a codi nadiu. A diferència de les solucions híbrides com React Native o Ionic, que utilitzen una capa intermitja per comunicar-se amb el sistema operatiu, Flutter compila directament a codi nadiu. Aquesta característica millora la velocitat d'execució i la resposta davant les accions dels usuaris.
- **Disseny Consistent:** Flutter proporciona una experiència visual i funcional uniforme tant en dispositius Android com iOS gràcies als seus widgets personalitzables. Això és una millora respecte a les tecnologies com React Native, que poden requerir ajustaments addicionals per garantir que l'aplicació sigui coherent entre plataformes. Flutter permet als desenvolupadors mantenir un disseny consistent sense necessitat de personalitzar excessivament el codi per a cada plataforma.
- **Productivitat Augmentada:** Flutter utilitza el llenguatge Dart, que facilita l'escriptura de codi de manera eficient. La funcionalitat de **hot reload** de Flutter permet als desenvolupadors veure els canvis en temps real sense haver de reiniciar l'aplicació completa. Aquesta característica és superior

a les eines de desenvolupament d'altres frameworks com Xamarin, on el procés de reinici i recompliació pot ser més llarg.

- **Suport Actiu i Comunitat:** Flutter compta amb un suport actiu per part de Google i una comunitat en creixement que proporciona recursos i solucions per a problemes comuns. Aquesta comunitat en creixement i el suport continuat asseguren que Flutter continui evolucionant i millorant. Comparat amb frameworks com Xamarin o Ionic, que poden tenir comunitats més petites o menys activament suportades, Flutter ofereix un ecosistema més robust i dinàmic.

Amb aquests avantatges, Flutter es presenta com una opció ideal per al desenvolupament d'aplicacions mòbils que necessiten un rendiment excel·lent, una interície d'usuari atractiva i una alta eficiència en el procés de desenvolupament. La seva arquitectura i funcionalitats el diferencien clarament d'altres tecnologies i el converteixen en una elecció destacada per a projectes amb requisits exigents.

Caldria explicar una mica els avantatges de cada tecnologia que es presenta respecte a altres alternatives, justificant-ne així l'elecció.corregit

7.3.3 IDE Escollida: Android Studio

Per al desenvolupament d'aplicacions amb Flutter, hi ha diversos entorns de desenvolupament integrats (IDE) disponibles. En aquest projecte, s'ha escollit **Android Studio** com a IDE per implementar la part client. Android Studio és l'entorn de desenvolupament integrat oficial per a aplicacions Android, creat per Google. Aquest IDE ofereix un conjunt complet d'eines dissenyades per facilitar la creació i gestió d'aplicacions destinades a dispositius Android.

Un aspecte destacat d'Android Studio és **l'emulador integrat**, que permet provar les aplicacions en dispositius simulats sense necessitat de tenir un dispositiu físic. L'emulador simula diverses configuracions de dispositius Android, com ara diferents mides de pantalla i versions del sistema operatiu. Això permet als desenvolupadors observar com es comporta l'aplicació en una gamma de condicions i configuracions diverses. L'ús de l'emulador facilita la depuració i les proves, oferint una manera ràpida i flexible de verificar el funcionament de l'aplicació abans del desplegament final.

7.4 Back-end

El *back-end* és la part de l'arquitectura d'una aplicació que opera al servidor, gestionant la lògica de negoci, el processament de dades i la comunicació amb altres serveis. La seva funció principal és assegurar que l'aplicació funcioni correctament, processant les sol·licituds dels usuaris i gestionant les operacions internes necessàries per proporcionar una resposta adequada.

7.4.1 Tecnologies més conegudes

Per al desenvolupament del backend d'una aplicació mòbil, hi ha diverses tecnologies àmpliament utilitzades que ofereixen diferents avantatges segons les necessitats específiques del projecte. A continuació es presenten algunes de les tecnologies més conegudes:

- **Node.js:** Node.js és una plataforma de servidor basada en el motor JavaScript V8 de Google Chrome. Permet als desenvolupadors escriure el backend de les aplicacions utilitzant JavaScript.
 - **Avantatges:**
 - * **Asíncron i No Bloquejant:** Permet manejar moltes connexions simultànies amb alta eficiència.
 - * **Ecosistema Gran:** Té un ecosistema extens amb nombroses llibreries i paquets disponibles a través de npm (Node Package Manager).
 - * **Full Stack JavaScript:** Permet usar JavaScript tant al frontend com al backend, facilitant la integració entre ambdós.
 - **Ruby on Rails:** Ruby on Rails, sovint anomenat Rails, és un framework de desenvolupament web en Ruby que utilitz a el patró Model-View-Controller (MVC).
 - **Avantatges:**
 - * **Desenvolupament Ràpid:** Facilita el desenvolupament ràpid gràcies a la seva estructura de convenció sobre configuració.
 - * **Facilitat d'ús:** Ofereix una sintaxi neta i llegible que simplifica el codi.
 - * **Comunitat Activa:** Té una comunitat activa amb moltes gems (llibreries) disponibles per afegir funcionalitats.

- **Spring Boot:** Spring Boot és un framework en Java que facilita el desenvolupament d'aplicacions empresarials basades en Spring, proporcionant una configuració mínima i opcions de desplegament senzilles.

– Avantatges:

- * **Configuració Automàtica:** Ofereix configuració automàtica basada en les dependències incloses al projecte.
- * **Escalabilitat:** És adequat per a aplicacions d'empresa amb grans requisits d'escalabilitat.
- * **Integració:** Facilita la integració amb altres tecnologies i serveis de Spring.

- **Express.js:** Express.js és un framework minimalist i flexible per a Node.js que proporciona un conjunt de característiques per al desenvolupament web i de backend.

– Avantatges:

- * **Simplicitat i Flexibilitat:** Permet construir aplicacions ràpidament amb una arquitectura senzilla i flexible.
- * **Middleware:** Ofereix suport per a middleware, facilitant la gestió de les sol·licituds HTTP.
- * **Popularitat:** Amplament utilitzat en combinació amb MongoDB i Angular/React per a desenvolupaments full-stack (coneugut com a MEAN/MERN stack).

- **GraphQL:** GraphQL és una consulta d'API i llenguatge de manipulació de dades desenvolupat per Facebook. Ofereix una alternativa a REST per a la consulta i manipulació de dades.

– Avantatges:

- * **Consultes Flexibles:** Permet als clients sol·licitar exactament les dades que necessiten, evitant la sobrecàrrega de dades.
- * **Tipat Fort:** Utilitza un sistema de tipus fort que facilita el desenvolupament i depuració.
- * **Evolució de l'API:** Permet evolucionar l'API sense afectar les consultes existents.

Igual que abans, caldria explicar una mica els avantatges de cada tecnologia que es presenta respecte a altres alternatives, justificant-ne així l'elecció.corregit

7.4.2 Tecnologia Escollida: GraphQL

En el desenvolupament del backend de l'aplicació mòbil, aqueste projecte utilitza **GraphQL** com a tecnologia principal. GraphQL és un llenguatge de consulta d'API desenvolupat per Facebook que ofereix diverses avantatges respecte a altres tecnologies de backend tradicionals com REST.

Una de les principals avantatges de GraphQL és la seva capacitat per proporcionar **consultes flexibles**. A diferència de les API REST, on cada endpoint retorna un conjunt fix de dades, GraphQL permet als clients especificar exactament quins camps volen recuperar. Això evita la sobrecàrrega de dades i optimitza la comunicació entre el client i el servidor, ja que només es transfereixen les dades necessàries per a cada sol·licitud. Aquesta característica és especialment valuosa en entorns mòbils, on l'ús eficient de l'ample de banda i la velocitat de resposta són essencials.

A més, GraphQL utilitza un sistema de tipus fort i un esquema ben definit que facilita tant el desenvolupament com la depuració. Els tipus de dades i les operacions disponibles estan clarament especificats en l'esquema, cosa que permet una millor validació i documentació automàtica de l'API. Això no només accelera el desenvolupament, sinó que també millora la qualitat i la mantenibilitat del codi.

En comparació amb tecnologies com REST, que requereixen múltiples endpoints per accedir a diferents recursos, GraphQL simplifica l'arquitectura de l'API. Amb un sol endpoint, es poden realitzar diverses consultes i mutacions, cosa que redueix la complexitat del sistema i millora la seva eficiència.

7.4.3 Codi obert per implementar servidors GraphQL: Apollo Server

Apollo Server és una biblioteca poderosa i versàtil que simplifica la creació i gestió de servidors GraphQL, oferint una interfície intuïtiva i un ecosistema integrat que facilita el desenvolupament.

Un dels seus punts forts més destacats és la seva completa compatibilitat amb els estàndards i especificacions de GraphQL, garantint que les aplicacions creades amb aquesta tecnologia segueixin les millors pràctiques i funcionin de manera òptima en qualsevol entorn GraphQL. Entre les seves característiques avançades, destaquen la gestió de subscripcions per a dades en temps real, el suport per a directives personalitzades i una gestió eficient de la memòria cau.

Distribuït a través de **npm** (Node Package Manager), Apollo Server és fàcil d'instal·lar i actualitzar, la qual cosa permet als desenvolupadors integrar-lo ràpidament en els seus projectes. Aquesta facilitat d'ús és un factor clau per al seu èxit en entorns de desenvolupament moderns.

7.4.4 Llenguatge Escollit: TS

El llenguatge de programació utilitzat en la part de *back-end* d'aquest projecte és **TypeScript** (TS). TypeScript és una extensió de JavaScript que incorpora tipatge estàtic al llenguatge. Això permet definir els tipus de dades per a variables i funcions, facilitant la detecció d'errors en temps de compilació abans que el codi s'executi. Aquesta funcionalitat no només millora la qualitat del codi, sinó que també enriqueix l'experiència de desenvolupament amb autocompleció i suggeriments intel·ligents en els editors de codi, augmentant així la productivitat dels desenvolupadors.

A més, TypeScript és totalment compatible amb el codi JavaScript existent, ja que és un superconjunt d'aquest llenguatge. Això facilita la integració progressiva de TypeScript en projectes que utilitzin JavaScript, permetent una transició suau. El codi TypeScript es compila a JavaScript, la qual cosa garanteix la seva compatibilitat amb qualsevol entorn que suporti JavaScript, com els navegadors web i entorns de servidor com Node.js.

7.5 Base de dades

Una base de dades és un sistema organitzat per emmagatzemar, gestionar i recuperar dades de manera eficient. Aquest sistema permet als usuaris emmagatzemar gran quantitat d'informació en un format estructurat, facilitant la consulta i actualització de les dades. Amb l'evolució de la tecnologia, les bases de dades en el núvol s'han convertit en una opció popular per a moltes organitzacions.

Les bases de dades en el núvol representen una solució moderna per a l'emmagatzematge i gestió de dades, proporcionades com a servei a través d'Internet. Aquest enfocament ofereix nombrosos avantatges en comparació amb les bases de dades tradicionals instal·lades localment.

- **Escalabilitat:** Les bases de dades en el núvol permeten ajustar fàcilment els recursos disponibles en funció de les necessitats de l'aplicació. Això

significa que pots augmentar o disminuir la capacitat de la base de dades sense haver de preocupar-te per la infraestructura física.

- **Accessibilitat Global:** Com que es proporcionen a través d'Internet, les bases de dades en el núvol són accessibles des de qualsevol lloc del món, permetent la col·laboració i accés remot a les dades, cosa que és ideal per a equips distribuïts i aplicacions amb usuaris globals.
- **Manteniment i Actualitzacions:** El proveïdor del servei de base de dades en el núvol es fa càrrec del manteniment, les actualitzacions de programari i les tasques de seguretat. Això redueix la càrrega de treball dels administradors de sistemes i garanteix que la base de dades es mantingui actualitzada i segura.
- **Cost Flexible:** Les bases de dades en el núvol sovint utilitzen un model de pagament basat en el consum, on pagues només pel que utilitzes. Això pot resultar més econòmic en comparació amb la compra i el manteniment de maquinari propi.
- **Seguretat:** Els proveïdors de serveis en el núvol sovint ofereixen mesures de seguretat avançades, com xifrat de dades, autenticació multifactor i controls d'accés, per protegir les dades emmagatzemades contra amenaçes i atacs.

7.5.1 Proveïdors Populars de Bases de Dades en el Núvol

Hi ha diversos proveïdors destacats que ofereixen serveis de bases de dades en el núvol, cadascun amb les seves pròpies característiques i avantatges. A continuació es presenten alguns dels més coneguts:

- **Amazon Web Services (AWS) - Amazon RDS:** AWS ofereix Amazon Relational Database Service (RDS), que suporta diverses bases de dades relacionals com MySQL, PostgreSQL, MariaDB, Oracle i SQL Server. Amazon RDS facilita la gestió de bases de dades amb funcions com el backup automàtic, la restauració, i l'escalabilitat.
- **Google Cloud Platform (GCP) - Cloud SQL:** Google Cloud SQL és un servei de base de dades gestionada que admet MySQL, PostgreSQL i SQL Server. Ofereix una gestió senzilla de les bases de dades amb funcions com el backup automàtic, l'escalabilitat i una integració profunda amb altres serveis de Google Cloud.

- **Microsoft Azure - Azure SQL Database:** Azure SQL Database és un servei de base de dades relacional completament gestionat per Microsoft. Supòrta la migració de bases de dades SQL Server al núvol i proporciona una alta disponibilitat, escalabilitat automàtica i seguretat integrada.
- **MongoDB Atlas:** MongoDB Atlas és el servei de base de dades en el núvol gestionat per MongoDB, Inc. Ofereix una solució completament gestionada per a MongoDB, amb funcions com el backup automàtic, l'escalabilitat dinàmica i l'anàlisi en temps real. És ideal per a aplicacions que utilitzen dades no relacionals.
- **Firebase Realtime Database:** Firebase, una plataforma de Google, ofereix Firebase Realtime Database com a servei de base de dades NoSQL en temps real. Aquesta base de dades permet la sincronització automàtica de dades en temps real entre clients i la base de dades, facilitant el desenvolupament d'aplicacions mòbils i web amb funcionalitats en temps real.

7.5.2 Base de Dades Escollida: MongoDB Atlas

En aquest projecte, la base de dades s'ha implementat utilitzant **MongoDB Atlas**.

MongoDB, a diferència de MySQL i PostgreSQL, és una base de dades NoSQL basada en documents. Aquesta característica li confereix una gran flexibilitat en l'estrucció de dades, ja que no requereix un esquema fix com les bases de dades relacionals. En contrast, MySQL i PostgreSQL utilitzen esquemes fixes que poden complicar la modificació de l'estrucció de les dades un cop establerta.

En comparació amb Firebase, una base de dades en temps real i basada en núvol, MongoDB ofereix un control més gran sobre la infraestructura i les opcións d'escalat. Firebase proporciona una solució gestionada que simplifica el procés de desenvolupament, mentre que MongoDB permet una personalització més gran i un control detallat en la configuració i el desplegament, especialment útil per a projectes que requereixen escalabilitat horitzontal o integració amb altres serveis de núvol.

Quant a escalabilitat, MongoDB destaca per la seva capacitat per gestionar grans volums de dades mitjançant el "sharding" (distribució de dades en múltiples servidors), facilitant l'escalat horitzontal. Aquesta característica és particularment avantatjosa en comparació amb MySQL i PostgreSQL, que tradicional-

ment escalen verticalment (és a dir, millorant el maquinari del servidor) i poden trobar limitacions quan es tracta d'escalar per a grans volums de dades.

7.6 Control de Versions

El control de versions és una pràctica essencial en el desenvolupament de software que permet gestionar i seguir els canvis realitzats en el codi font d'un projecte al llarg del temps. Utilitzant sistemes de control de versions, els desenvolupadors poden:

- **Rastrejar canvis:** Cada modificació al codi es regista amb informació sobre qui la va fer, quan i per què. Això facilita la revisió de l'historial de canvis i ajuda a identificar la causa de qualsevol problema que pugui sorgir.
- **Col·laborar:** Diversos desenvolupadors poden treballar simultàniament en diferents parts d'un projecte. El control de versions facilita la fusió de les seves contribucions i la resolució de conflictes que puguin sorgir.
- **Revertir canvis:** Si un canvi resulta ser problemàtic, és possible tornar a una versió anterior del codi sense perdre altres modificacions realitzades posteriorment.
- **Gestionar versions:** Els sistemes de control de versions permeten mantenir diverses versions del projecte, com ara versions de desenvolupament, versions estables i versions de producció.

7.6.1 Control de Versions Escollida: GitHub

GitHub és una plataforma de control de versions basada en Git que proporciona diversos avantatges significatius. Facilita una col·laboració eficaç entre desenvolupadors mitjançant el sistema de pull requests, que permet la revisió i fusió de canvis en el codi.

En un entorn amb múltiples desenvolupadors, cada un d'ells disposa d'una còpia completa del repositori, la qual cosa afavoreix un treball descentralitzat i una sincronització efectiva dels canvis.

En el cas d'aquest projecte, utilitzar branques ha facilitat la gestió de tasques i ha permès un control més acurat dels canvis, optimitzant el flux de treball i la integració de novetats.

A més, GitHub proporciona eines per gestionar projectes, com issues i taules Kanban, que ajuden a organitzar tasques i seguir l'evolució del projecte. La plataforma també suporta integracions amb eines d'automatització, com CI/CD, que milloren l'eficiència del procés de desenvolupament.

Amb la seva comunitat activa i la possibilitat de contribuir a projectes de codi obert, GitHub ofereix visibilitat i oportunitats per a la col·laboració amb altres desenvolupadors. Finalment, les funcionalitats de seguretat de GitHub permeten gestionar l'accés i protegir el codi, assegurant que només les persones autoritzades puguin realitzar canvis.

CAPÍTOL 8

Anàlisi i disseny del sistema

8.1 Introducció

L'anàlisi i el disseny són dues etapes clau en el desenvolupament de sistemes informàtics, cadascuna amb objectius específics.

L'anàlisi es centra en identificar i definir les necessitats del sistema. En aquesta fase, es creen models detallats que inclouen el model de dades, que defineix l'estructura i les relacions de les dades, i el model de processos, que descriu els fluxos de treball i les operacions necessàries.

El disseny, en canvi, ofereix solucions concretes basades en els resultats de l'anàlisi. Aquesta fase implica la creació d'interfícies d'usuari, models físics de dades i processos, i, si és necessari, models d'objectes. S'utilitzen patrons de disseny per aplicar solucions establertes a problemes comuns, facilitant la creació de sistemes que siguin eficaços i reutilitzables.

En les següents seccions, es presentarà el disseny del sistema mitjançant diversos diagrames.

Podria estar bé algun petit text introductorí, almenys per dir què s'explicarà a les següents seccions.corregit

8.2 Diagrama de casos d'ús

El diagrama de casos d'ús és una eina gràfica utilitzada en el desenvolupament de sistemes per identificar i representar les interaccions entre els usuaris (client i administrador) i el sistema en si. Aquest diagrama ajuda a entendre les funcionalitats que el sistema ha de proporcionar als usuaris i com els usuaris interactuen amb aquestes funcionalitats.

8.2.1 Diagrama de casos d'ús de l'usuari client

El diagrama de casos d'ús 8.1 il·lustra les diferents operacions que un **client** pot realitzar dins del sistema. Encara que hi ha dues instàncies de l'actor client,

aquestes representen el mateix actor i es mostren així per a millorar la llegibilitat del gràfic. (Aquest incís també s'hauria de posar a la llegenda de la figura)corregit. Cadascuna d'aquestes instàncies està connectada a diversos casos d'ús.

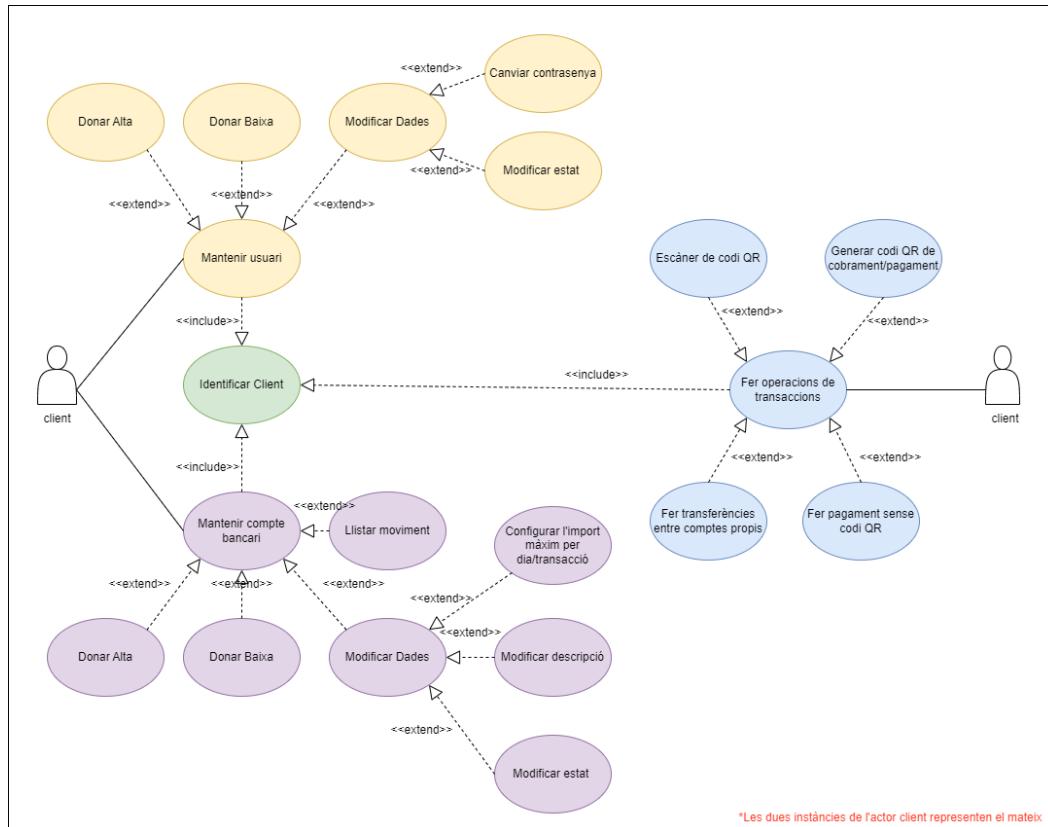


Figura 8.1: Diagrama de casos d'ús de context de client. “Cambiar” hauria de ser “Canviar”, i “tranferencies” hauria de ser “transferències”. Si la llegenda és massa llarga i no vols que a l’índex de figures del principi t’hi quedi una frase tan llarga, pots utilitzar els [], tal com t’ho he posat en aquest cas si mires el codi LaTeX de la figura.

corregit

Recorda citar la figura (per exemple: “El diagrama il·lustrat a la Figura 8.1 mostra les diferents operacions...”).corregit

L’actor client pot realitzar operacions relacionades amb la gestió del seu compte d’usuari, incloent-hi donar-se d’alta, donar-se de baixa, modificar les seves dades personals, canviar la contrasenya i modificar l’estat del seu compte. Totes aquestes operacions formen part d’un cas d’ús més general anomenat **Mantenir usuari**, que centralitza aquestes funcions. Per poder realitzar aquestes accions, el client ha de passar pel procés d’**Identificar usuari**, que és un pas

previ necessari per garantir la seguretat de les operacions.

A més de gestionar el compte d'usuari, el client també pot gestionar el seu compte bancari a través del cas d'ús **Mantenir compte bancari**, que inclou accions com donar d'alta un compte, donar-lo de baixa, modificar les dades del compte, llistar els moviments bancaris, i configurar l'import màxim per transacció o per dia. Aquestes funcions estan dissenyades per permetre al client controlar i supervisar les seves operacions bancàries dins del sistema.

Un altre grup important de casos d'ús està relacionat amb les **operacions de transaccions**. Aquí, el client pot escanejar codis QR per realitzar pagaments o cobraments, generar codis QR, fer transferències entre els seus propis comptes i realitzar pagaments sense necessitat d'utilitzar un codi QR. Aquestes operacions són essencials per a la funcionalitat de qualsevol sistema que impliqui gestió financer, ja que permeten al client executar transaccions de manera ràpida i segura.

8.2.2 Diagrama de casos d'ús de l'usuari administrador

Aquest segon diagrama de cas d'ús 8.2.2 mostra les diferents funcions que un **administrador** pot dur a terme dins d'un sistema. L'actor principal és l'administrador, qui interactua amb diversos casos d'ús que representen les operacions disponibles.

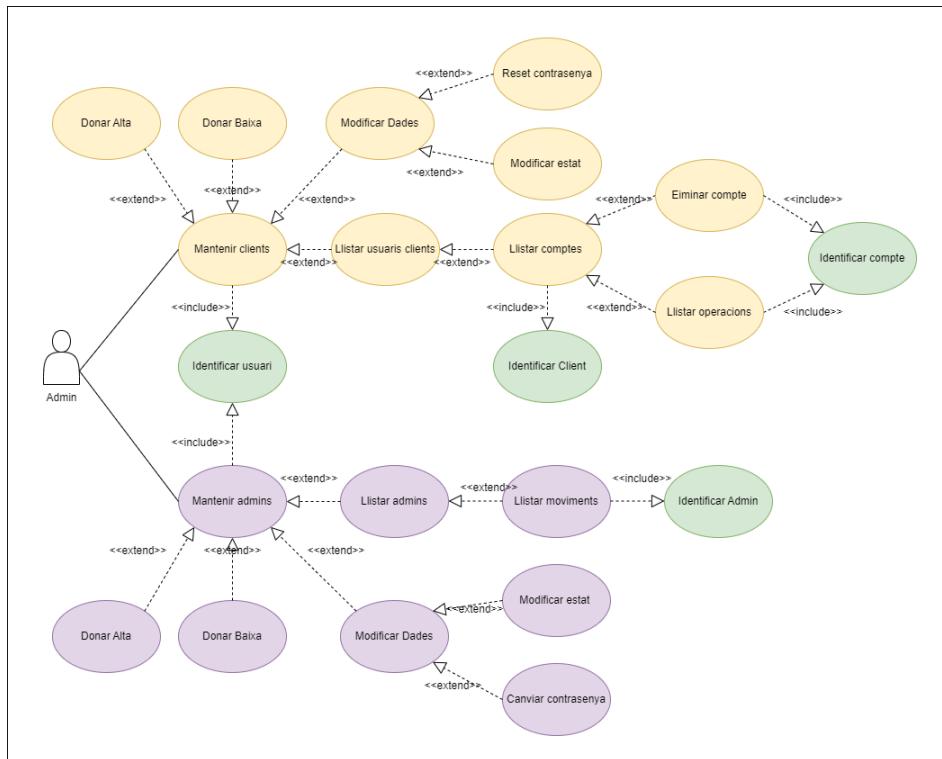


Figura 8.2: Diagrama de casos d'ús de context d'administrador. **“Llitar” hauria de ser “Llistar”, i “Cambiar” hauria de ser “Canviar”.**

corregit

Cal citar bé la figura.corregit

Un dels casos d'ús centrals és **Mantenir clients**, que inclou accions com donar d'alta nous clients, donar de baixa clients existents, modificar les dades dels clients, llistar els usuaris clients, restablir la contrasenya dels clients i modificar l'estat dels clients. Un altre cas d'ús important és **Mantenir admins**, que implica afegir nous administradors, eliminar administradors, modificar les dades dels administradors, llistar els administradors existents i modificar l'estat dels administradors.

A més, el diagrama mostra altres casos d'ús relacionats amb la gestió de comptes i operacions. **Llistar comptes** permet veure la llista de comptes de clients, mentre que **Eliminar compte** s'encarrega d'eliminar un compte de client. **Llistar operacions** permet revisar les operacions realitzades pels clients, i **Llistar moviments** té una funció similar però aplicada als comptes o administradors. També hi ha casos d'ús que fan referència a la identificació d'usuaris, comptes, clients i administradors, mostrant la interrelació entre aquestes operacions i la necessitat d'autenticació o identificació per dur-les a terme.

8.2.3 Diagrama de casos d'ús del sistema

Addicionalment, el sistema compta amb la funció de generar logs per a cada operació, tal com es mostra a la diagrama 8.3. Això vol dir que, fins i tot en cas d'eliminació d'un usuari o un compte, l'historial associat es manté sempre guardat.

Aquest enfocament assegura la conservació de dades importants per a possibles auditòries o revisions posteriors, garantint així la traçabilitat i la seguretat del sistema.

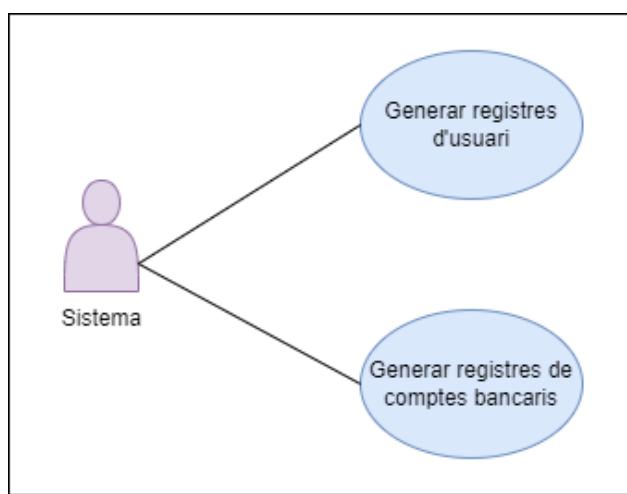


Figura 8.3: Diagrama de casos d'ús de context del sistema. “bancàries” hauria de ser “bancaris”.

corregit

Cal citar bé la figura.corregit

8.3 Fitxes de cas d'ús

Un fitxer de cas d'ús és un document que descriu detalladament les interaccions entre un usuari i un sistema per aconseguir un objectiu concret. Aquest fitxer és fonamental en el desenvolupament de sistemes informàtics, ja que proporciona una visió clara i comprensible dels requisits funcionals del sistema des de la perspectiva de l'usuari.

A continuació, es presenten cinc fitxers de cas d'ús que han estat creats per descriure les funcionalitats clau del sistema:

8.3.1 Generar codi QR de cobrament

Aquest primer fitxer de cas d'ús que mostra la figura 8.4 descriu el procés que un usuari segueix per generar un codi QR de cobrament. Inclou la identificació de l'usuari, la validació del compte bancari i la generació del codi QR amb les condicions específiques. Com que es tracta d'un cas de cobrament d'import, els missatges en el codi QR no estan xifrats i, a més, aquest codi QR no caduca mai.

CAS D'ÚS	Generar codi QR de cobrament
Versió	Visió usuari
Descripció	Un usuari vol generar un codi QR per a cobrament
Actors	Usuari client
Precondició	Usuari registrat al sistema i obté un compte bancari
Flux principal	<ol style="list-style-type: none"> 1. Identificar l'usuari 2. Comprovar si l'usuari està en estat actiu 3. Identificar el compte bancari 4. Introduir (o no) l'import que es desitja cobrar 5. Generar el codi QR de cobrament sense xifrat
Flux alternatiu	Si l'usuari està en estat desactivat, llavors aquesta operació es rebutja
Postcondició	Si tot està correcte, es genera el codi QR de cobrament sense xifrat; en cas contrari, no es realitza cap acció

Figura 8.4: Generar codi QR de cobrament

8.3.2 Generar codi QR de pagament

Aquest segon fitxer de cas d'ús que mostra la figura 8.5 és molt similar al primer. L'única diferència és que, en aquest cas, es tracta d'un pagament, cosa que implica més restriccions. En concret, el missatge del codi QR està xifrat, el codi QR caduca i, a més, un codi QR només pot ser escanejat una sola vegada.

CAS D'ÚS	Generar codi QR de pagament
Versió	Visió usuari
Descripció	Un usuari vol generar un codi QR per a pagament
Actors	Usuari client
Precondició	Usuari registrat al sistema i obté un compte bancari
Flux principal	<ol style="list-style-type: none"> 1. Identificar l'usuari 2. Comprovar si l'usuari està en estat actiu 3. Identificar el compte bancari 4. Comprovar si el compte està en estat actiu 5. Introduir (o no) l'import que es desitja pagar 6. Si s'ha introduït l'import: <ul style="list-style-type: none"> Comprovar que no s'ha superat el límit diari establert Verificar que hi hagi saldo suficient 7. Generar el codi QR de pagament xifrat, que caduca en 1 minut
Flux alternatiu	Si l'usuari o el compte bancari està en estat desactivat, o l'import no és acceptable, l'operació es rebutja
Postcondició	Si tot està correcte, es genera el codi QR de pagament xifrat; en cas contrari, no es realitza cap acció

Figura 8.5: Generar codi QR de pagament

8.3.3 Escanejar un codi QR de pagament

El quart fitxer de cas d'ús que mostra la figura 8.6, detalla els passos que un usuari ha de seguir per escanejar un codi QR de pagament i completar la transacció. Inclou la verificació de l'import i el compte bancari, així com el procés de cobrament.

CAS D'ÚS	Escanejar un codi QR de pagament
Versió	Visió usuari
Descripció	Un usuari vol escanejar un codi QR de pagament amb l'objectiu de cobrar un import
Actors	Usuari client
Precondició	Usuari registrat al sistema i obté un compte bancari
Flux principal	<ol style="list-style-type: none"> 1. Identificar l'usuari 2. Comprovar si l'usuari està en estat actiu 3. Identificar el compte bancari 4. Escanejar el codi QR <ul style="list-style-type: none"> Si l'import no està especificat, introduir l'import que es vol cobrar 5. Comprovar si l'import introduït és acceptable per al compte de pagament 6. Rebre l'import
Flux alternatiu	Si l'usuari està en estat desactivat o l'import de pagament no és acceptable per al compte de pagament, l'operació es rebutja.
Postcondició	Si tot està correcte, es genera el codi QR de pagament xifrat; en cas contrari, no es realitza cap acció

Figura 8.6: Generar codi QR de cobrament

8.3.4 Escanejar un codi QR de cobrament

Aquest quart fitxer de cas d'ús, que es mostra a la figura 8.7, detalla els passos que un usuari ha de seguir per escanejar un codi QR de cobrament i completar la transacció. Inclou la verificació de l'import i el compte bancari, així com el procés de cobrament.

CAS D'ÚS	Escanejar un codi QR de cobrament
Versió	Visió usuari
Descripció	Un usuari vol escanejar un codi QR de cobrament amb l'objectiu de pagar un import
Actors	Usuari client
Precondició	Usuari registrat al sistema i obté un compte bancari
Flux principal	<ol style="list-style-type: none"> 1. Identificar l'usuari 2. Comprovar si l'usuari està en estat actiu 3. Identificar el compte bancari 4. Comprovar si el compte està en estat actiu 4. Escanejar el codi QR Si l'import no està especificat, introduir l'import que es vol pagar 5. Comprovar si l'import introduït és acceptable per al compte de pagament 6. Pagar l'import
Flux alternatiu	Si l'usuari està en estat desactivat o l'import de pagament no és acceptable per al compte de pagament, l'operació es rebutja.
Postcondició	Si tot està correcte, es genera el codi QR de pagament xifrat; en cas contrari, no es realitza cap acció

Figura 8.7: Escanejar un codi QR de cobrament

8.3.5 Administrador modificar l'estat d'un compte bancari

El últim fitxer de cas d'ús que apareix a sota de la figura 8.8 explica el procés mitjançant el qual un administrador pot canviar l'estat d'un compte bancari associat a un client. Inclou la identificació del compte bancari, la selecció del compte a modificar, i la realització de la modificació.

CAS D'ÚS	Modificar l'estat d'un compte bancari
Versió	Visió administrador
Descripció	Un administrador vol modificar l'estat d'un compte bancari d'un client
Actors	Usuari administrador
Precondició	Usuari administrador i client registrats al sistema, i el compte bancari existeix
Flux principal	<ol style="list-style-type: none"> 1. Identificar l'usuari administrador 2. Llistar tots els clients 3. Trobar el propietari del compte bancari 4. Llistar tots els comptes bancaris d'aquest client 5. Escollir el compte bancari que es vol modificar 6. Modificar l'estat del compte bancari
Flux alternatiu	Si no s'ha pogut identificar l'usuari administrador, l'usuari client o el compte bancari, l'operació es rebutja.
Postcondició	Si tot està correcte, es modifica l'estat del compte bancari

Figura 8.8: Administrador modificar l'estat d'un compte bancari

Hauries de posar algunes fitxes de cas d'ús. Almenys unes 5 o 6 dels casos d'ús més importants.

8.4 Diagrama de la base de dades

Un diagrama de base de dades és una representació gràfica que mostra les taules d'una base de dades i les relacions entre elles. Ajuda a visualitzar com es dissenya i organitza la informació.

A continuació, es presenta el diagrama 8.9 amb una descripció detallada de les seves components principals:

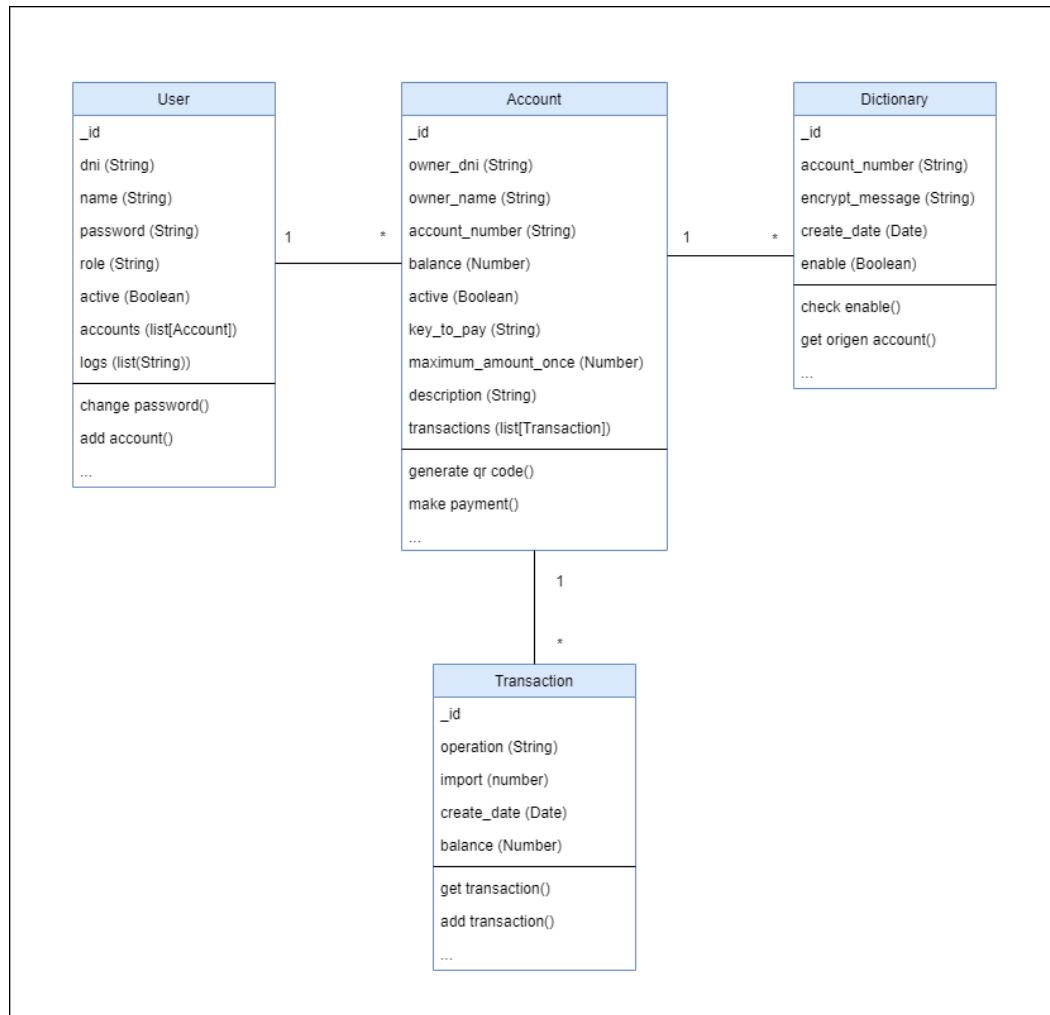


Figura 8.9: Diagrama de la base de dades

Cal citar bé la figura.corregit

El diagrama representa un model de dades que inclou quatre entitats principals: **User**, **Account**, **Transaction** i **Dictionary**. Aquestes entitats estan interrelacionades per gestionar informació d'usuaris, comptes, transaccions i altres dades associades. A continuació, es detalla cadascun d'ells.

8.4.1 User

L'entitat **User** defineix els usuaris del sistema amb atributs bàsics com el DNI, el nom, la contrasenya, el rol dins del sistema (administrador o client), així com altres camps com l'estat actiu, la llista de comptes associats i els registres d'activitat (*logs*). Aquesta entitat ofereix funcionalitats per canviar la contrasenya, afegir comptes nous i altres operacions relacionades amb l'usuari.

Aquesta entitat conté les dades obligatòries que cada usuari ha de tenir:

- **_id** (Object_id) [Clau primària]: Generat automàticament per MongoDB.
- **dni** (String) [NOT NULL]: Emmagatzema el número d'identificació de l'usuari, no repetible.
- **name** (String) [NOT NULL]: Emmagatzema el nom de l'usuari.
- **password** (String) [NOT NULL]: Emmagatzema la contrasenya de l'usuari, que s'encripta abans d'inserir-la.
- **role** (String) [NOT NULL]: Emmagatzema el rol de l'usuari, que pot ser administrador o client.
- **active** (Boolean) [DEFAULT: “true”]: Indica si l'usuari està disponible per fer operacions de transacció.
- **accounts** (Account List) [DEFAULT: []]: Emmagatzema tots els ID de comptes que pertanyen a l'usuari.
- **logs** (String List) [DEFAULT: []]: Emmagatzema els moviments de l'usuari.

8.4.2 Account

L'entitat **Account** representa els comptes bancaris associats als usuaris. Cada compte té atributs per identificar el propietari, el saldo disponible, una clau per generar codis QR de pagament, límits de transacció, i una descripció del compte. Els comptes poden generar codis QR per facilitar pagaments o cobraments i tenen associades llistes de transaccions.

Cada vegada que es genera un codi QR de pagament, la clau s'actualitza i es guarda la informació necessària a la taula **Dictionary**. Quan un client escaneja el codi QR, a partir de la taula Dictionary es pot identificar el propietari i obtenir

la clau per desxifrar el missatge.

Aquesta entitat conté les dades obligatòries que cada compte ha de tenir:

- **_id** (Object_id) [Clau primària]: Generat automàticament per MongoDB.
- **owner_dni** (String) [NOT NULL]: Emmagatzema el DNI del propietari del compte.
- **owner_name** (String) [NOT NULL]: Emmagatzema el nom del propietari del compte.
- **account_number** (String) [NOT NULL]: Emmagatzema el número de compte, que ha de ser únic.
- **balance** (Number) [NOT NULL]: Emmagatzema el saldo disponible en el compte.
- **active** (Boolean) [DEFAULT: “true”]: Indica si el compte està actiu i pot realitzar transaccions.
- **key_to_pay** (String) [NOT NULL]: Emmagatzema una clau per generar codis QR de pagament.
- **maximum_amount_once** (Number) [DEFAULT: 0]: Indica la quantitat màxima que es pot pagar en una sola transacció.
- **description** (String) [OPTIONAL]: Conté una descripció addicional del compte.
- **transactions** (Transaction List) [DEFAULT: []]: Emmagatzema la llista de transaccions associades al compte.

8.4.3 Dictionary

L'entitat **Dictionary** que mostra la diagrama 8.10 de sota emmagatzema informació complementària relacionada amb els comptes i les transaccions. Aquesta entitat es fa servir per guardar dades com missatges encriptats, dates de creació, i estats d'activació, permetent verificar i processar correctament les operacions financeres dins del sistema.

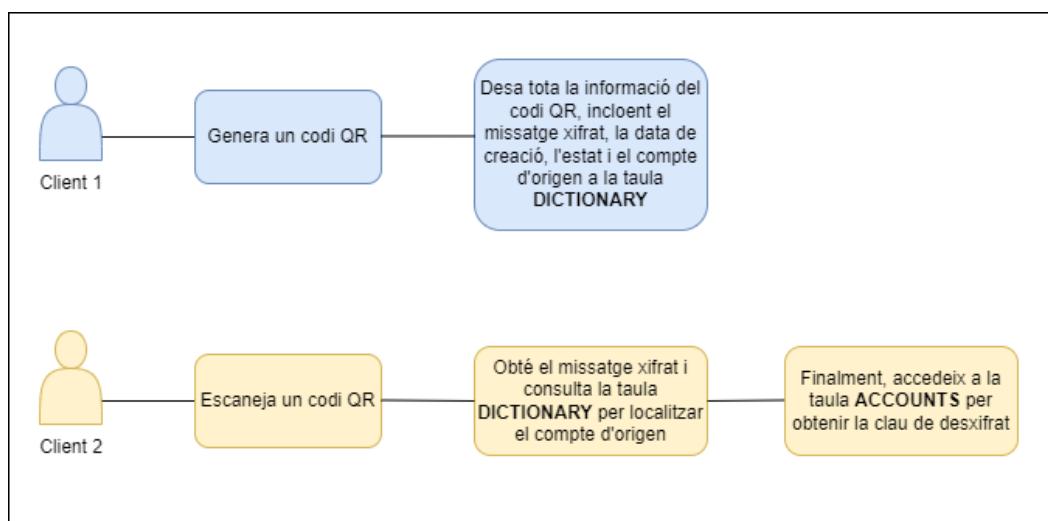


Figura 8.10: Ús de Dictionary. Canviar “informaciones” per “informacions”, “incloent, missatge” per “incloent missatge”, “en DICTIONARY” per “a la taula Dictionary”, “a DICTIONARY per trobat” per “a la taula Dictionary per trobar”, i “a ACCOUNTS” per “a la taula Account”.

corregit

Cal citar bé la figura.

Aquesta entitat conté les dades obligatòries que cada entrada ha de tenir:

- **_id** (Object_id) [Clau primària]: Generat automàticament per MongoDB.
- **account_number** (String) [NOT NULL]: Emmagatzema el número de compte associat.
- **encrypt_message** (String) [NOT NULL]: Conté el missatge encriptat relacionat amb la transacció.
- **create_date** (Date) [NOT NULL]: Indica la data de creació de l'entrada al diccionari.
- **enable** (Boolean) [DEFAULT: “true”]: Indica si l'entrada està activa i disponible per ser utilitzada.

8.4.4 Transaction

L'entitat **Transaction** descriu les operacions financeres realitzades en un compte. Aquesta entitat emmagatzema informació detallada sobre cada transacció, com ara el tipus d'operació, l'import, i el saldo resultant. Cada transacció genera dues instàncies: una per al compte d'origen (qui paga) i una altra per al compte de destí (qui cobra). Aquest diagrama reflecteix com les entitats es connecten per permetre la gestió d'usuaris, comptes i transaccions dins del sistema.

Aquesta entitat conté les dades obligatòries que cada transacció ha de tenir:

- **_id** (Object_id) [Clau primària]: Generat automàticament per MongoDB.
- **operation** (String) [NOT NULL]: Indica el tipus d'operació realitzada (per exemple, “pagament” o “cobrament”).
- **amount** (Number) [NOT NULL]: Emmagatzema l'import de la transacció.
- **create_date** (Date) [NOT NULL]: Indica la data de creació de la transacció.
- **balance** (Number) [NOT NULL]: Reflecteix el saldo resultant en el compte després de la transacció.
- **origin_account** (String) [NOT NULL]: Emmagatzema el número de compte d'origen de la transacció.
- **destination_account** (String) [NOT NULL]: Emmagatzema el número de compte de destí de la transacció.

8.5 Diagrama de processos

Aquesta diagrama de flux 8.11 representa els processos de gestió d'usuaris i transaccions de pagament mitjançant codis QR. Mostra com les diferents decisions i funcions es ramifiquen en funció de les característiques i accions de l'usuari, garantint que cada procés es gestiona adequadament segons les condicions del sistema.

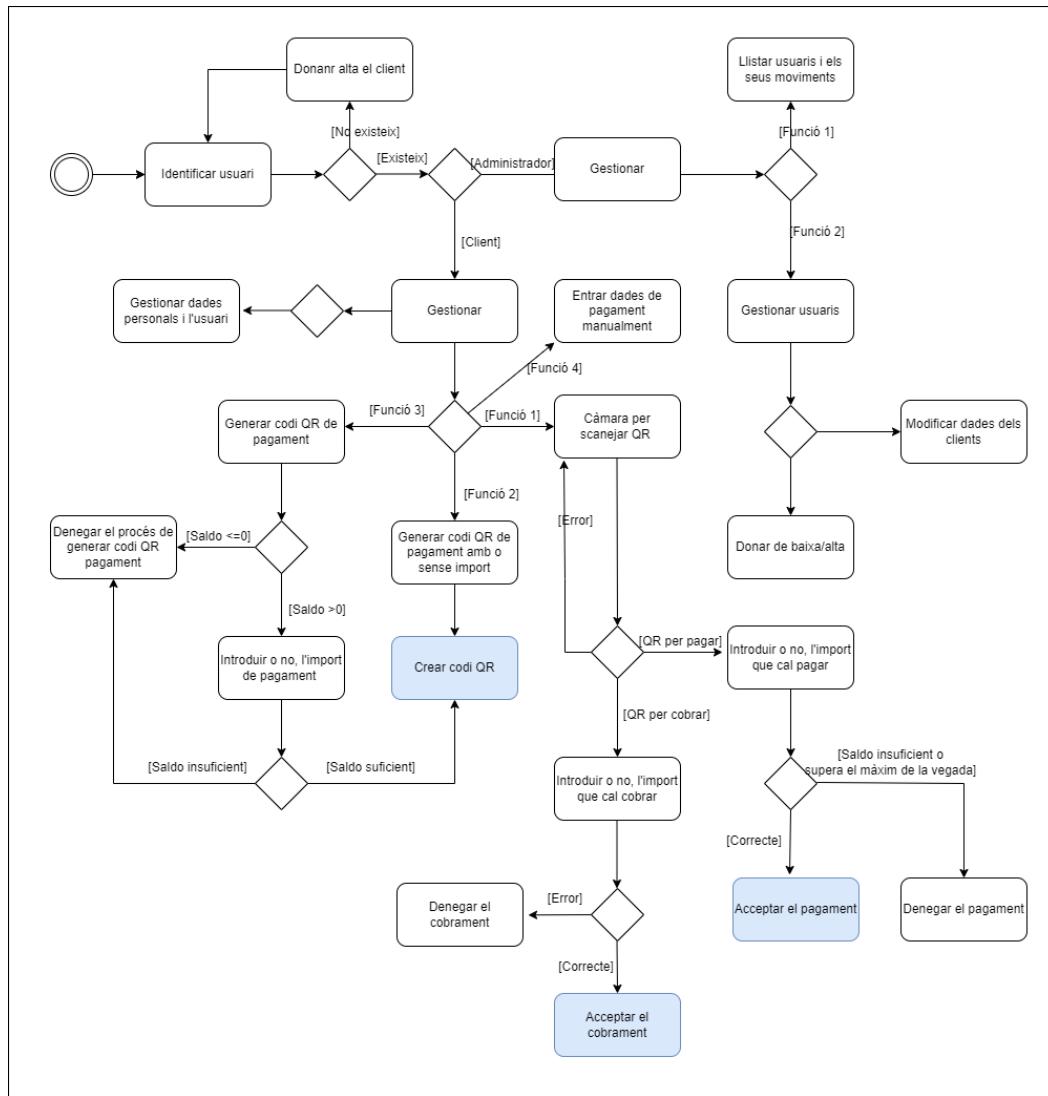


Figura 8.11: Diagrama de processos.

Cal citar bé la figura.corregit

8.5.1 Descripció del procés

- **Identificació de l'usuari:** El procés comença amb la identificació de l'usuari. Si l'usuari no existeix, es procedeix a **donar d'alta el client**. Si l'usuari existeix, es determina si és un **administrador** o un **client**.
- **Administradors:**
 - **Funció 1:** Permet llistar usuaris i els seus moviments.

- **Funció 2:** Permet gestionar usuaris, amb accions com **modificar les dades dels clients o donar-los d'alta o de baixa.**
- **Clients:**
 - Si l'usuari és un client, té l'opció de **gestionar les seves dades personals i d'usuari.**
 - També pot optar per **entrar les dades de pagament manualment**, una solució en cas que la càmera no funcioni.
 - **Funcions 2 i 3:** S'utilitzen per generar el codi QR de cobrament i de pagament. En cas de cobrament, no cal fixar-se en res (**no cal fixar res, o no cal fixar-se en res?**)**corregit**, però en cas de pagament, cal verificar si el compte disposa de saldo suficient. Un cop generat el codi QR, la càmera entra en joc per escanejar-lo.
 - **Funció 1:** La càmera s'encarrega de llegir el text que hi ha darrere del codi QR i, si el missatge és correcte, processa l'operació.
 - Cal destacar que es pot introduir l'import tant en el moment de generar el codi QR com després d'escanejar-lo.

8.6 Interfícies d'usuari

El disseny de les interfícies d'usuari ha estat una prioritat abans d'endinsarnos en el desenvolupament del *front-end*. Aquestes interfícies proporcionen una visió preliminar de l'aspecte i la sensació que tindrà l'aplicació. Les figures següents mostren els dissenys de les interfícies d'usuari.

Dissenyar tots els possibles estats de l'aplicació en un temps tan limitat és inviable. No obstant això, atès que l'aplicació principal i la dels usuaris comparteixen gran part de la seva interfície, només s'han dissenyat les vistes principals. En desenvolupar l'aplicació de l'usuari, la majoria dels components s'han reutilitzat, modificant-ne el comportament segons les necessitats, ja que les interaccions dels usuaris són limitades.

És important destacar que algunes vistes no corresponen exactament a com són a l'aplicació final, principalment perquè s'han modificat o afegit funcionalitats segons les necessitats durant el desenvolupament.

Per cada secció caldria explicar una mica la interfície (quines opcions té, per a què serveix cada botó... una explicació molt senzilla), i citar correctament la figura que s'està descriuint.

8.6.1 Client

L'usuari client és qui utilitza les funcionalitats de l'aplicació per gestionar les seves pròpies operacions i accions.

8.6.1.1 Primeres pàgines

En entrar a l'aplicació, es mostra una pàgina de benvinguda que es manté durant 3 segons. Després, la pàgina redirigeix automàticament a la pàgina de **Login**. Si ho desitja, l'usuari pot saltar els 3 segons clicant el botó *Login*.

Llavors, si el client no estigui registrat, cal anar primer a la pàgina de registre per donar-se d'alta. Amb el botó de *Register* s'anirà a la pàgina de **registre**.

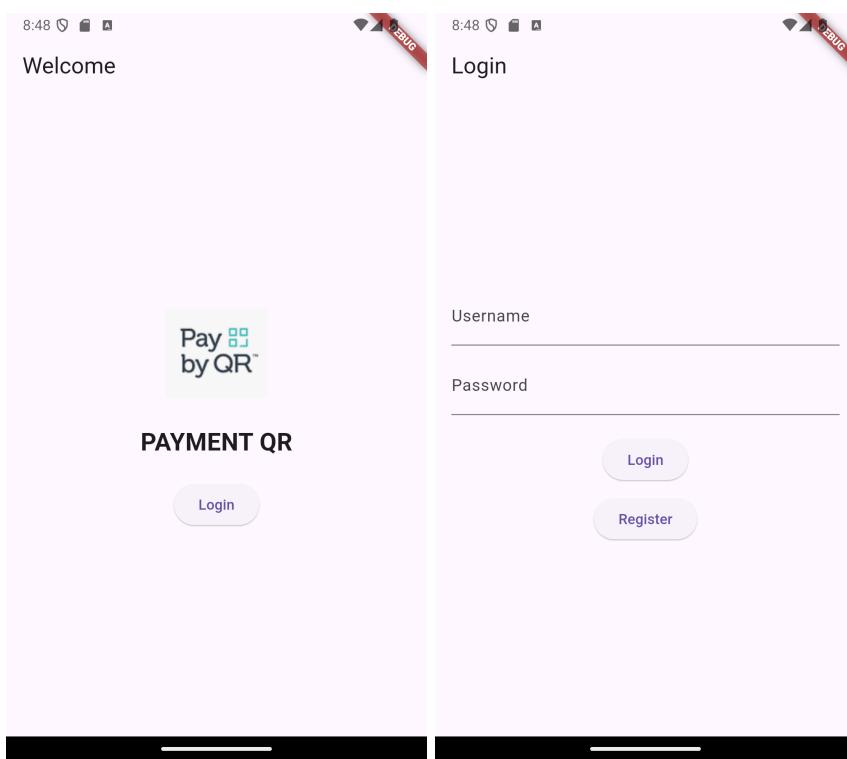


Figura 8.12: Primeres pàgines

8.6.1.2 Registrar un nou usuari

En aquesta pàgina es demanen dades personals, el sistema fa una comprovació per verificar si les dades introduïdes corresponen al format esperat.

Si les dades són correctes, envia la informació al servidor per iniciar el procés de registre. Si no ho són, el sistema indicarà que no s'ha pogut completar l'operació de registre. Al final del procés, es redirigirà automàticament a la pàgina de login.

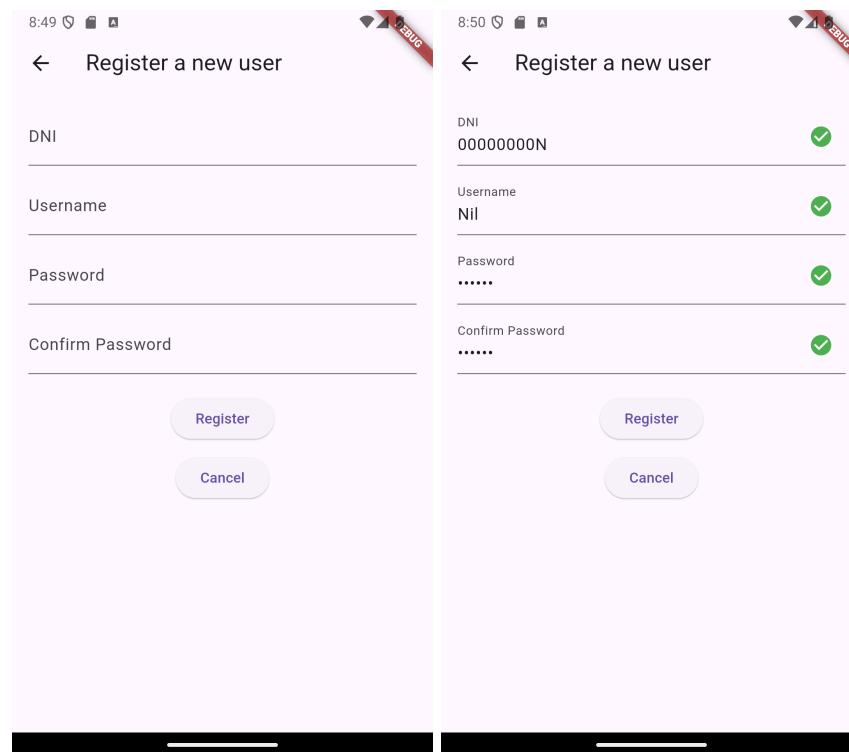


Figura 8.13: Registrar nou usuari client

8.6.1.3 Login

Una vegada l'usuari estigui registrat, podrà accedir a l'aplicació.

La figura 8.14 de la dreta mostra la pàgina principal de l'aplicació, on apareixen diversos botons. Al mig apareix un compte bancari que s'ha creat per defecte. Per poder utilitzar les funcionalitats de sota, cal que tingui un compte bancari seleccionat.

En les següents seccions es detallarà el significat de cadascun d'aquests elements.

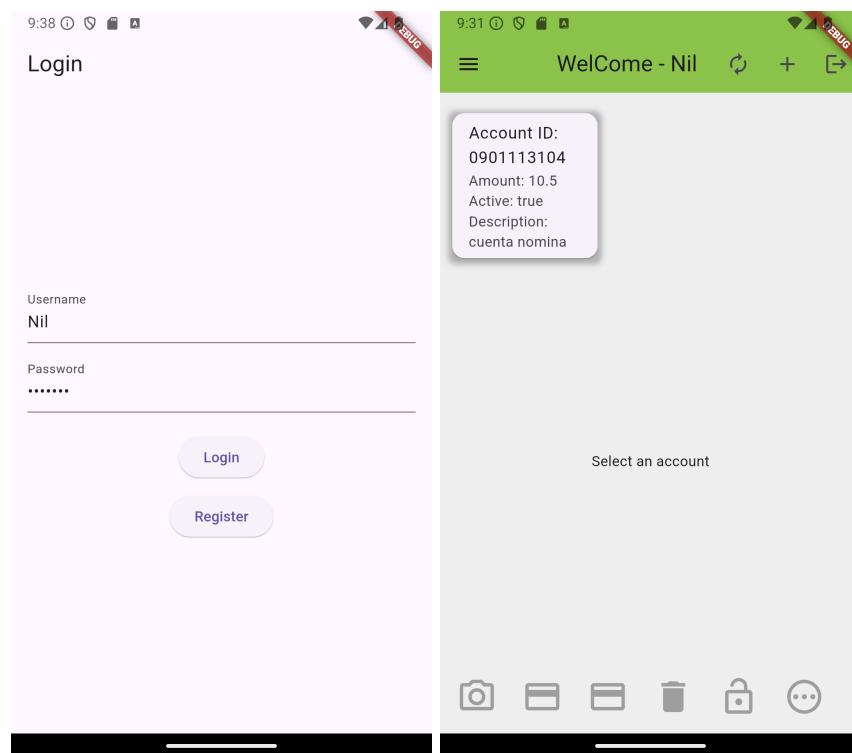


Figura 8.14: Pàgina de *login* - Client.

8.6.1.4 Càmera per escanejar codi QR

La càmera s'utilitza per escanejar un codi QR generat prèviament. Un cop escanejat el codi, si les dades són correctes, es processarà l'operació. Si tot va bé, es mostrarà un missatge de 'OK' amb els detalls de l'operació, i finalment es proporcionarà un botó per tornar a la pàgina principal.

En tornar a la pàgina principal, es mostrarà la transacció que s'ha fet just abans.

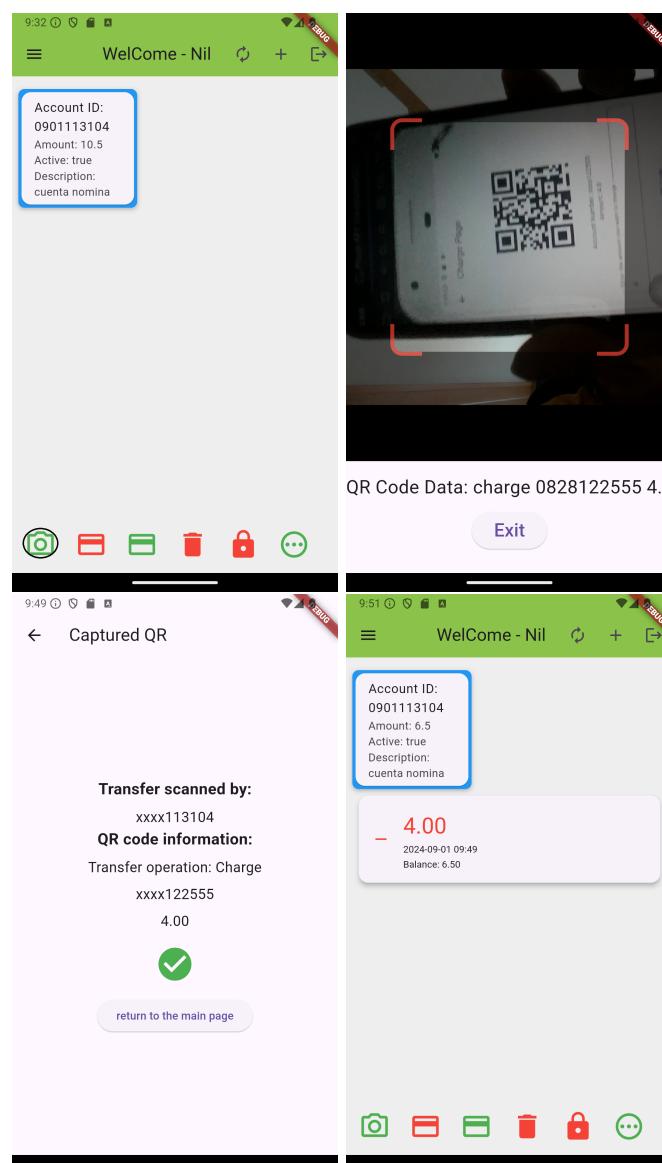


Figura 8.15: Escanejar codi QR

8.6.1.5 Página per generar codi QR de pagament

En entrar a la pàgina per generar un codi QR de pagament, hi haurà dues opcions:

- Amb import introduït: el sistema fa una comprovació per assegurar-se que l'import és acceptable.
- Sense import introduït: l'import serà introduït per la persona que escaneja, i després el sistema farà la comprovació.

De totes maneres, el codi QR caducarà després d'un temps; només serà vàlid l'últim QR generat i només es permetrà escanejar-lo una sola vegada.

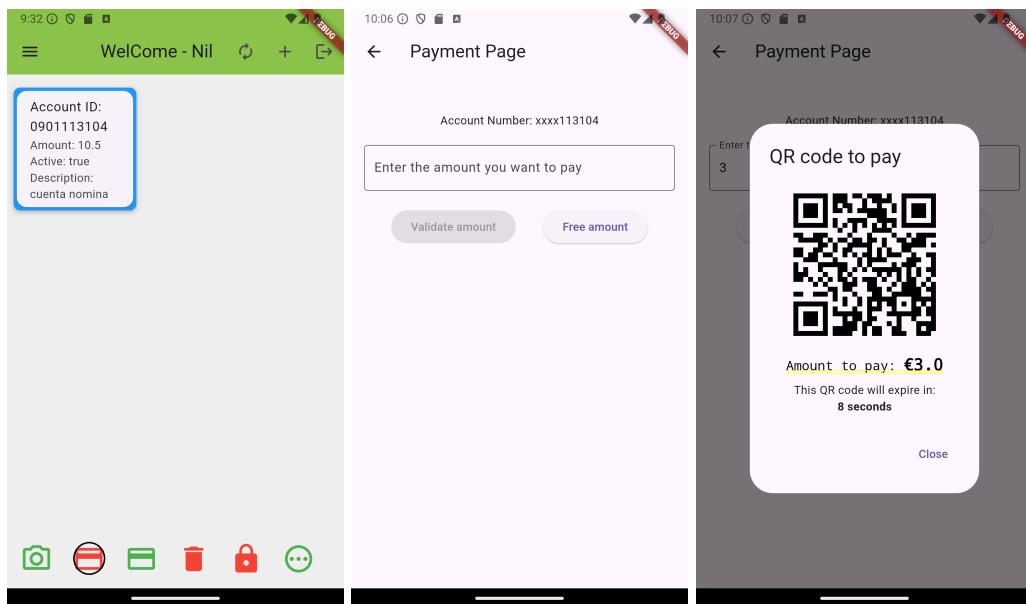


Figura 8.16: Generar un codi QR de pagament.

8.6.1.6 Generar codi QR de cobrament

És molt similar a la funcionalitat anterior 8.6.1.5. Per defecte, en entrar a la pàgina, ja es mostra el codi QR de cobrament sense un import definit.

La diferència és que aquest codi QR de cobrament no caduca mai i es pot fer servir tantes vegades com es vulgui.

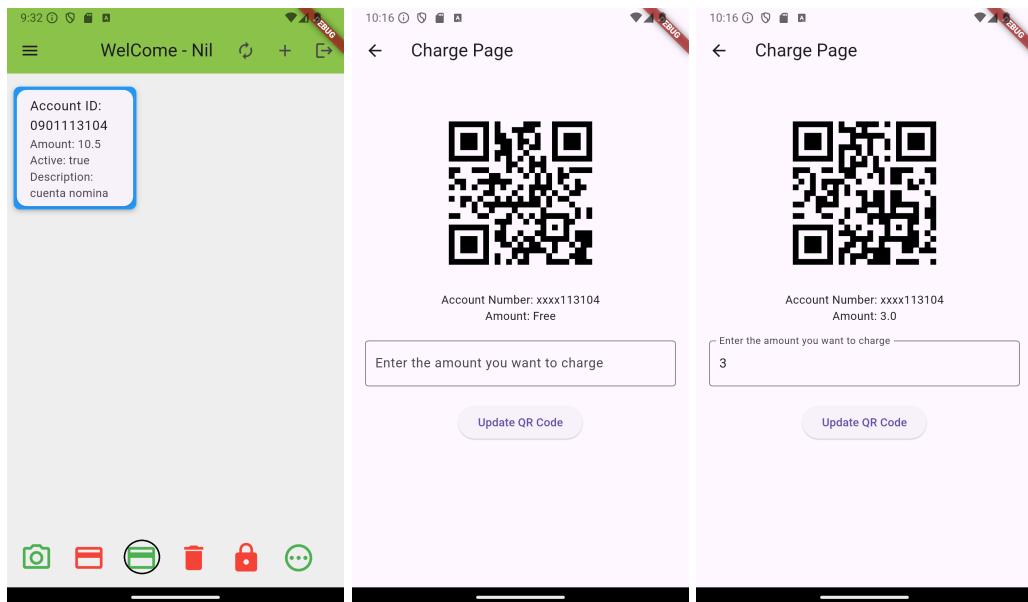


Figura 8.17: Pàgina per generar un codi QR de cobrament.

8.6.1.7 Eliminar un compte bancari

Per eliminar un compte bancari, cal que el saldo sigui 0. Si no és així, caldrà fer una transferència per deixar-lo a zero. Per facilitar-ho, l'app ofereix l'opció de fer transferències entre comptes bancaris del mateix client.

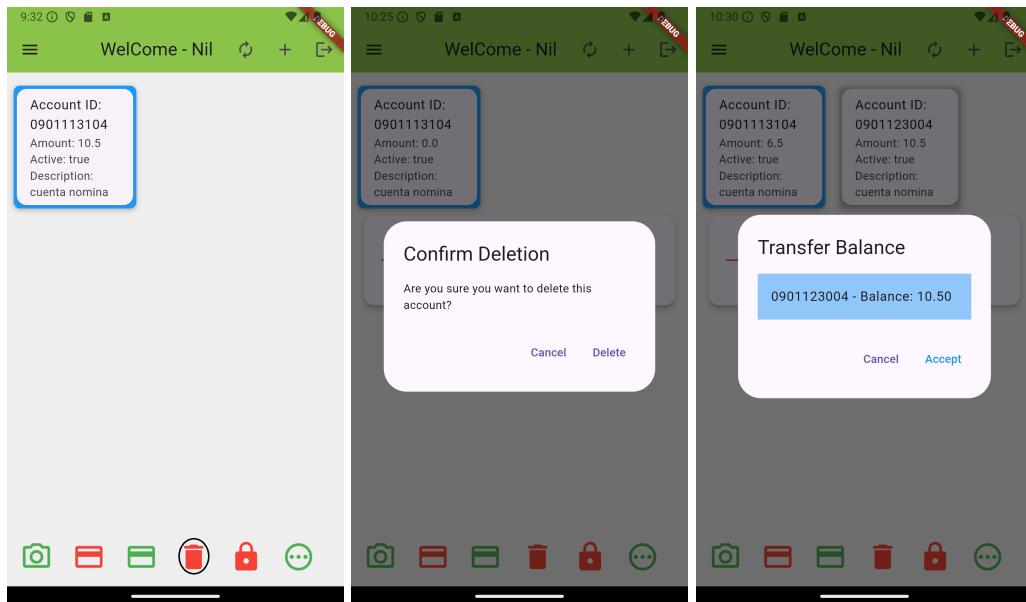


Figura 8.18: Pàgina per generar un codi QR de cobrament.

8.6.1.8 Modificar l'estat del compte bancari

Per activar o desactivar un compte bancari, es simplement clicant en l'icana del cadenat, es canviarà l'estat del compte bancari. Si està en estat de desactivat, no es podrà generar un QR de pagament ni efectuar cap pagament.

El cadenat es tanca o obre segons l'estat.

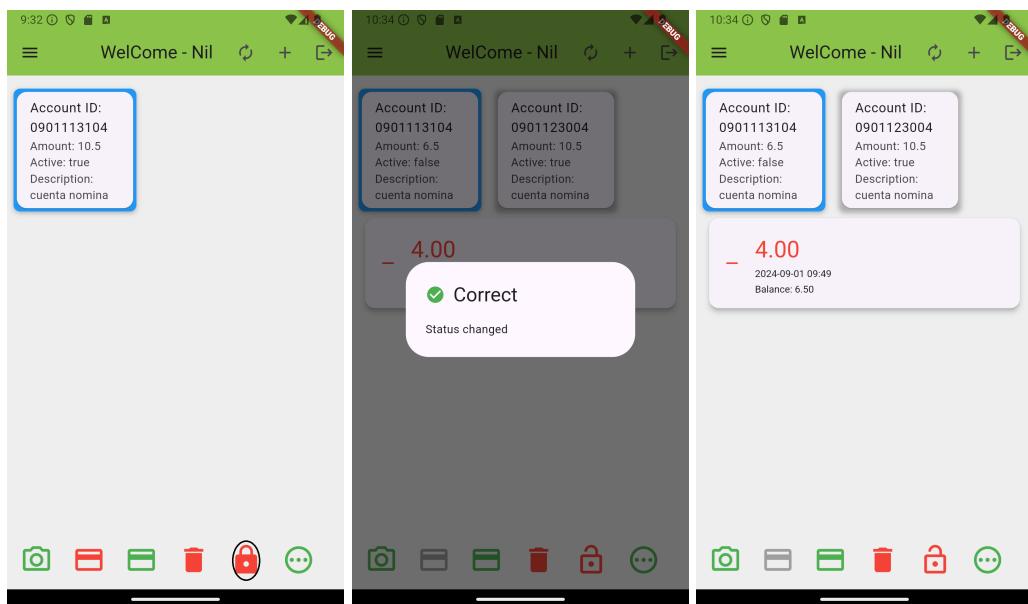


Figura 8.19: Modificar l'estat del compte bancari

8.6.1.9 Altres operacions sobre el compte bancari

A més de les funcionalitats bàsiques, també es poden realitzar les següents operacions sobre el compte bancari:

- Fer transaccions manualment sense generar ni escanejar un codi QR.
- Definir el nou límit de pagament per transacció.
- Definir una nova descripció per al compte bancari.

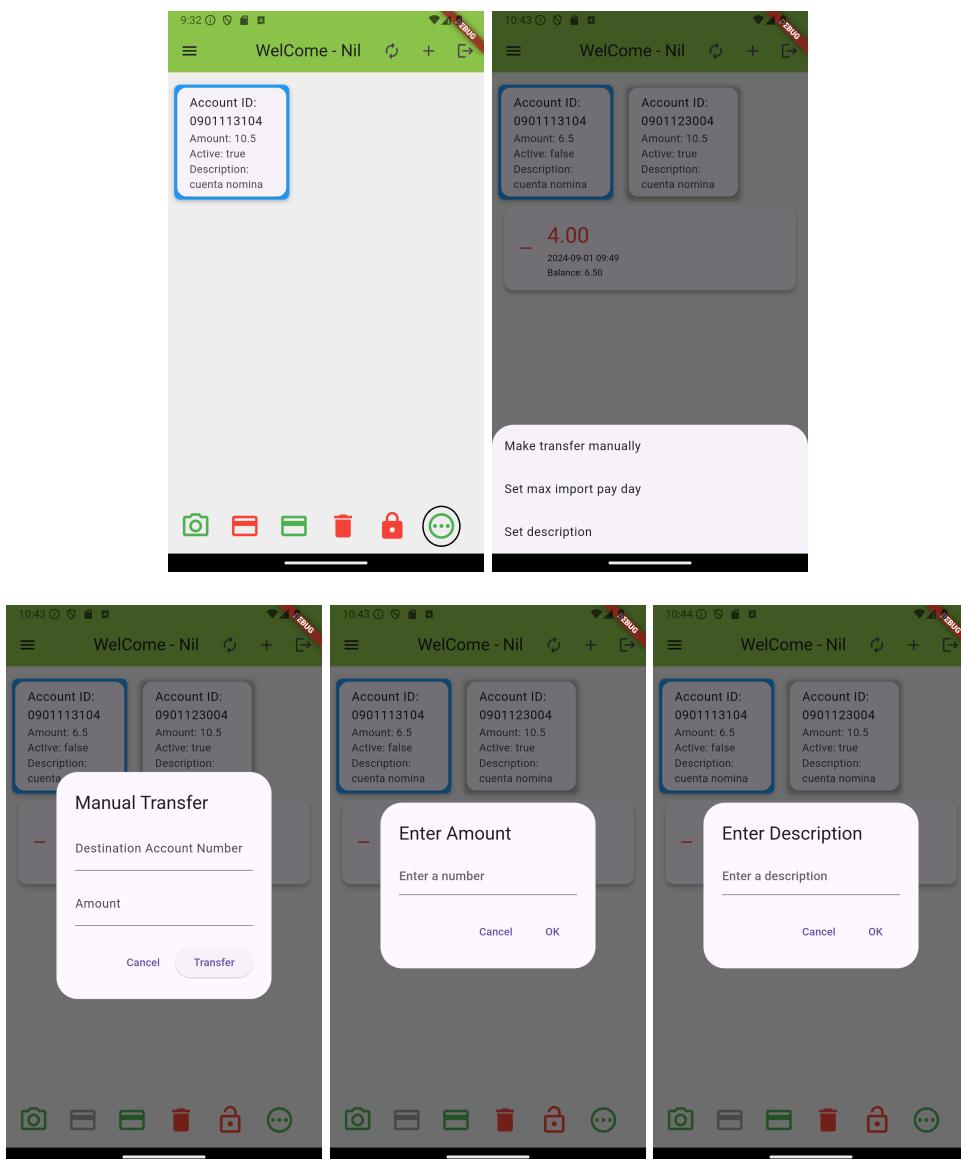


Figura 8.20: Modificar l'estat del compte bancari

8.6.1.10 Afegir un compte bancari per a l'usuari

Per afegir un nou compte bancari per a l'usuari, només cal clicar a l'ícone de "+". Es generarà i es mostrerà el nou compte creat, amb un import per defecte de 10,50 €.

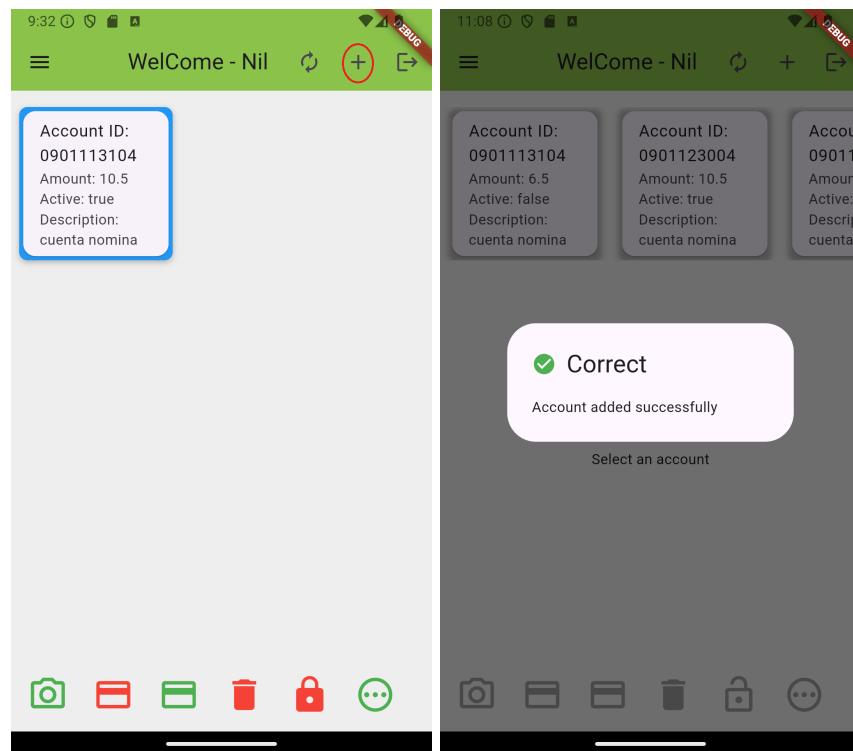


Figura 8.21: Afegir un compte bancari per a l'usuari

8.6.1.11 Refrescar la pàgina

Per refrescar la pàgina, simplement clica al botó de refresh. Això actualitzarà el contingut i mostrerà qualsevol canvi recent.

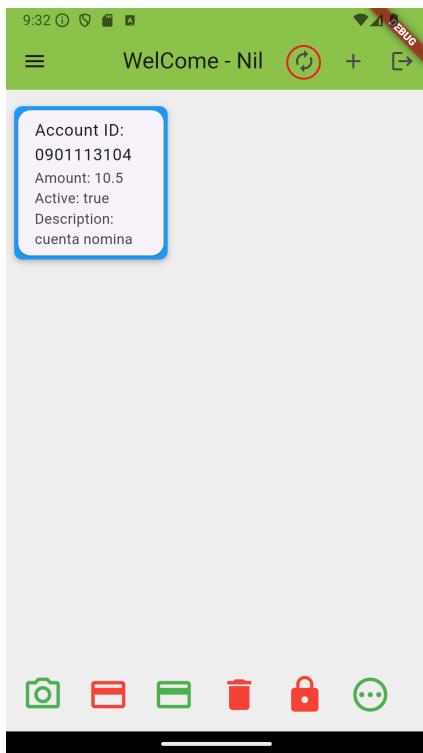


Figura 8.22: Refrescar la pàgina

8.6.1.12 Sortir de la secció

Per sortir de la secció, se't preguntarà si estàs segur que vols sortir. Un cop confirmada la teva decisió, es eliminaran les credencials i seràs redirigit a la pàgina de login.

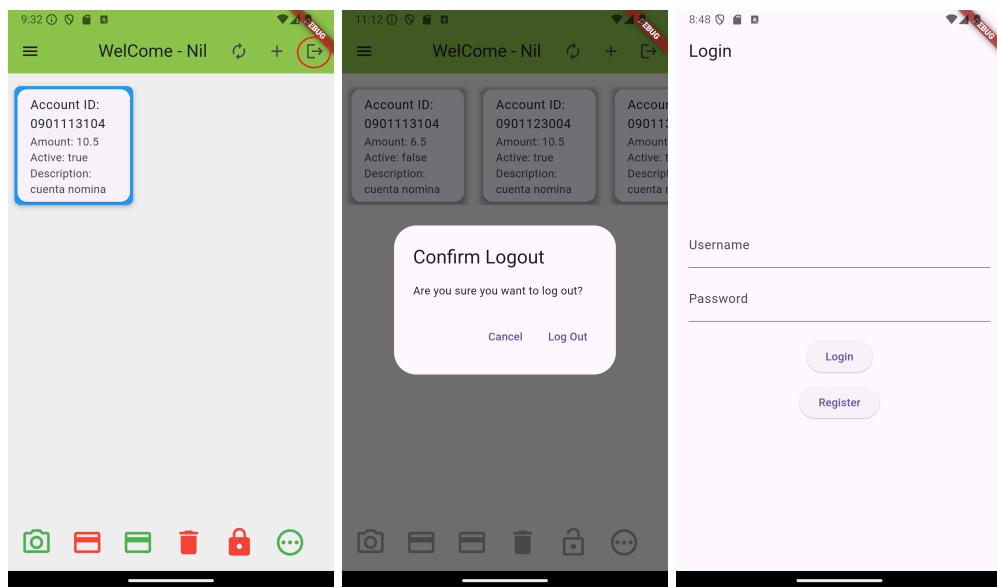


Figura 8.23: Sortir de la secció

8.6.1.13 Altres gestions de l'usuari

- **Canviar la contrasenya:** Per canviar la contrasenya, cal que introdueixis la contrasenya antiga i la nova dues vegades.
- **Canviar fons de color:** Per canviar el color de fons, cal que seleccionis el color que més et conveni.
- **Canviar l'estat:** Per modificar l'estat de l'usuari, si l'usuari està desactivat no podrà realitzar cap operació amb el compte bancari.
- **Eliminar usuari:** Per eliminar l'usuari, cal que no hi hagi import en els seus comptes bancaris.

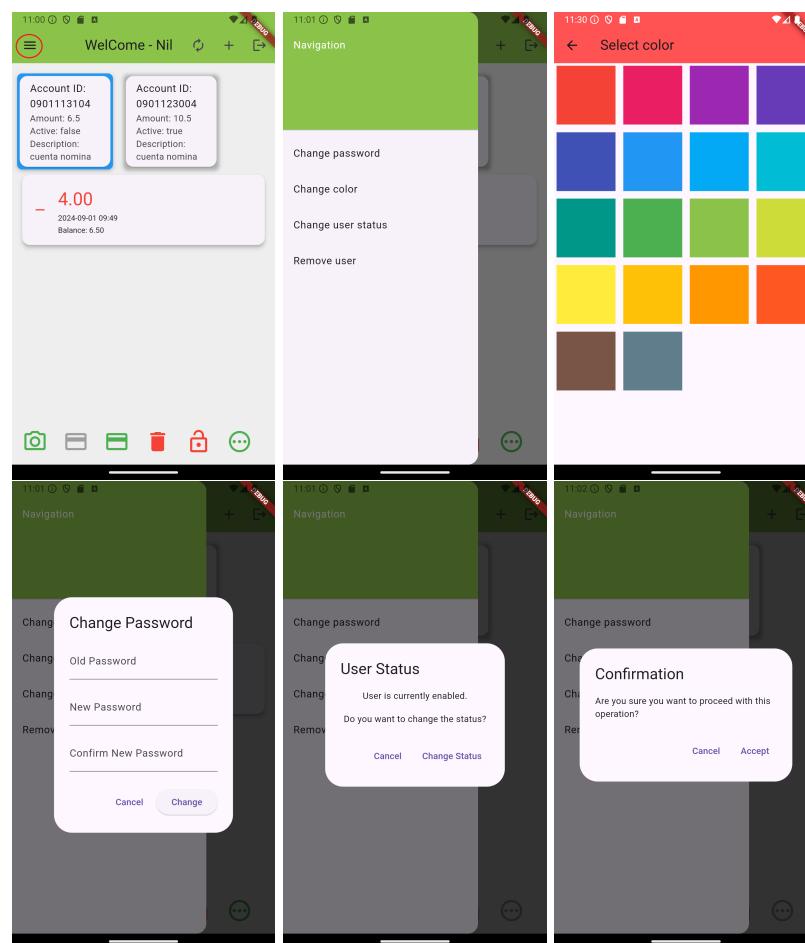


Figura 8.24: Altres gestions de l'usuari

8.6.2 Administrador

L'usuari administrador és qui té permisos especials per gestionar i configurar l'aplicació, incloent la creació i eliminació d'altres usuaris clients, així com la gestió de configuracions i opcions avançades.

8.6.2.1 Login d'un administrador

Un administrador accedeix de la mateixa manera que un client. Per defecte, hi ha un administrador predefinit amb el nom d'usuari "*Admin*" i la contrasenya "*Admin123*". Aquest administrador no es eliminable. Cal que s'identifiqui l'administrador per poder donar d'alta altres administradors.

La funcionalitat que hi apareix a sota s'activa només quan s'ha seleccionat un usuari.

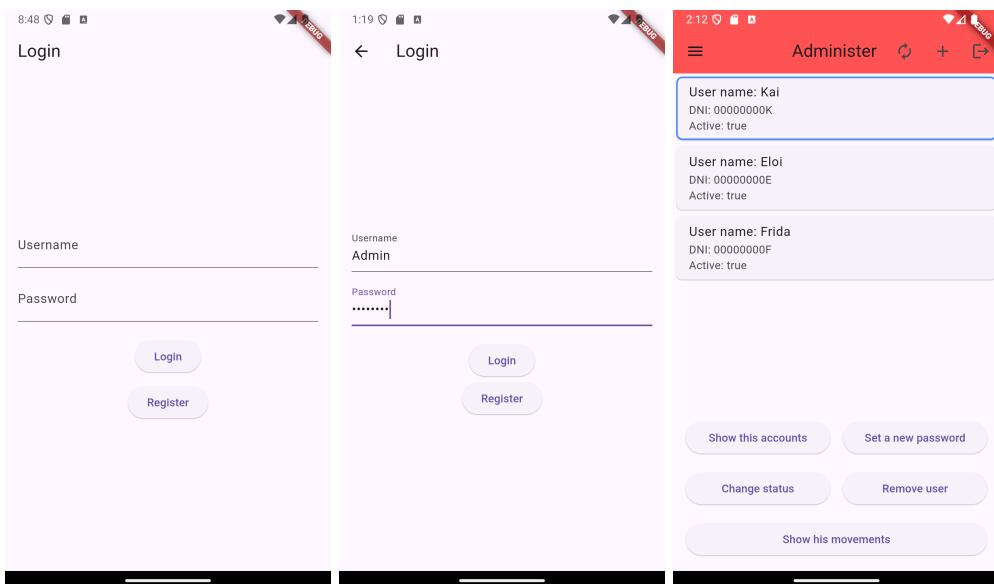


Figura 8.25: Pàgina de login - Administrador

8.6.2.2 Llistar els comptes dels clients i modificar-ne l'estat

Un administrador pot visualitzar tots els usuaris clients i els seus comptes bancaris. També pot modificar l'estat dels comptes bancaris.

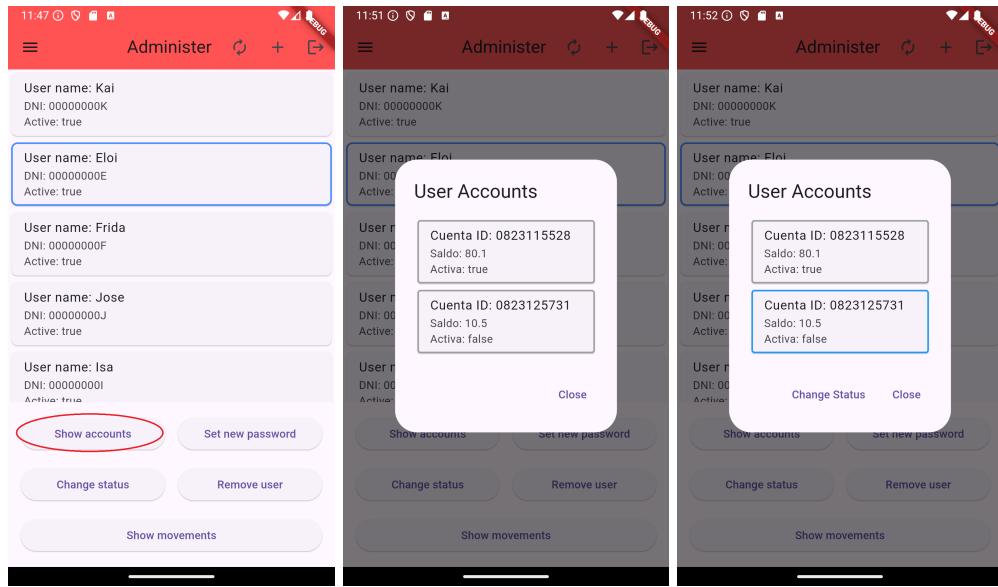


Figura 8.26: Llistar els comptes dels clients i modificar-ne l'estat

8.6.2.3 Crear una nova contrasenya per client

Un administrador pot canviar la contrasenya d'un usuari client sense haver de proporcionar la contrasenya antiga. La nova contrasenya té restriccions com en el moment de registrar-se: cal proporcionar una contrasenya d'almenys 3 caràcters, i el canvi es pot efectuar només si les dues contrasenyes coincideixen.

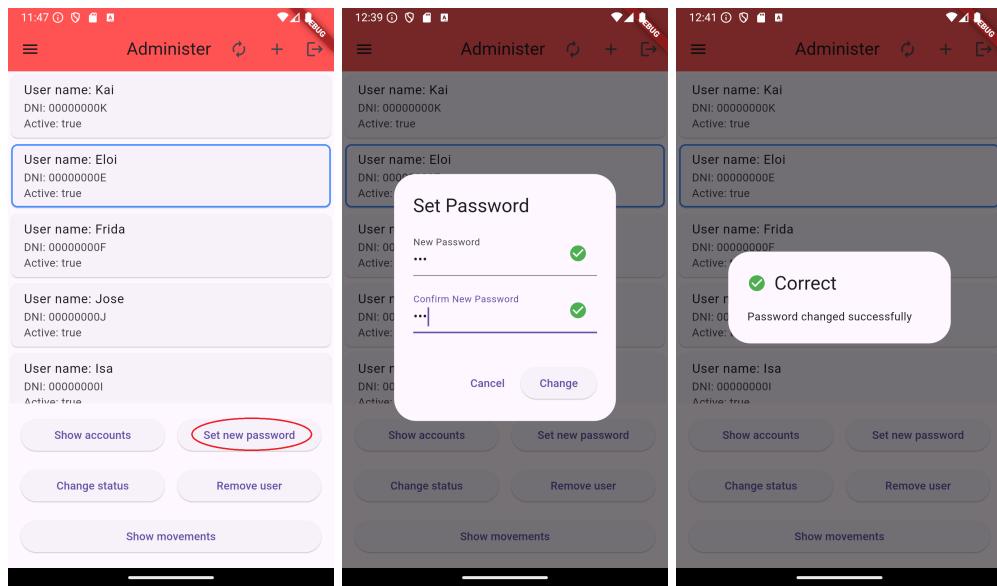


Figura 8.27: Crear una nova contrasenya per client

8.6.2.4 Modificar l'estat de l'usuari client

Un administrador té el privilegi de modificar l'estat d'un usuari. Si un usuari té l'estat desactivat, no podrà realitzar cap operació sobre els seus comptes bancaris.

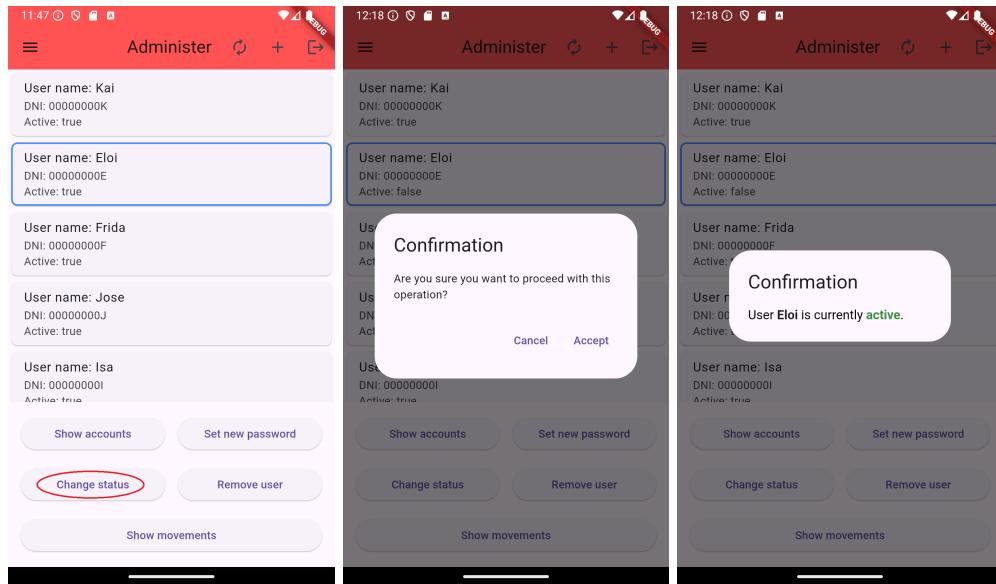


Figura 8.28: Modificar l'estat de l'usuari client

8.6.2.5 Eliminar un usuari client

Només es pot eliminar un usuari client quan el client no disposa de cap compte bancari. Primer es fa una comprovació, i després es mostra un diàleg que demana confirmació.

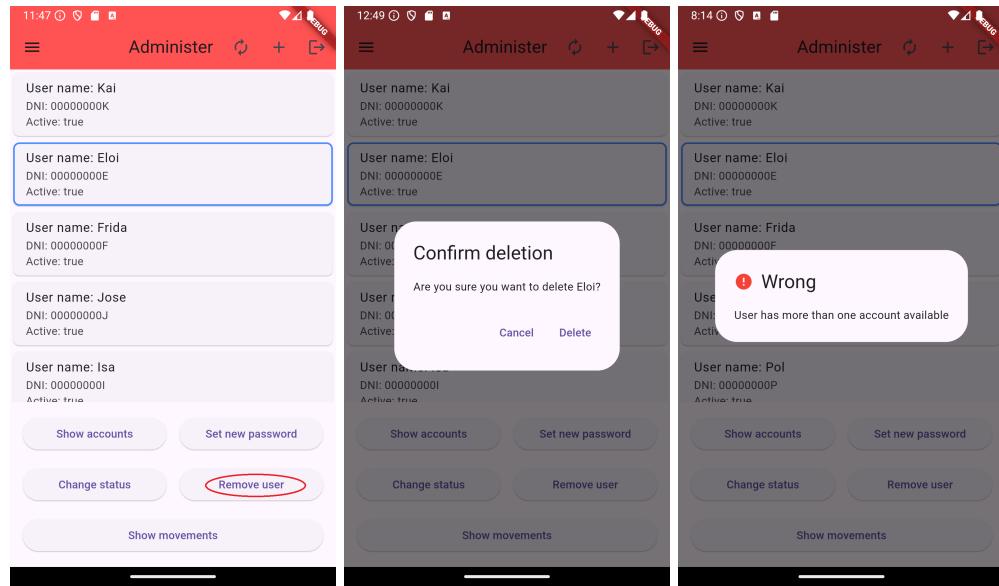


Figura 8.29: Eliminar un usuari client

8.6.2.6 Visualitzar els moviments d'un usuari

Un administrador pot llistar tots els moviments d'un usuari client.

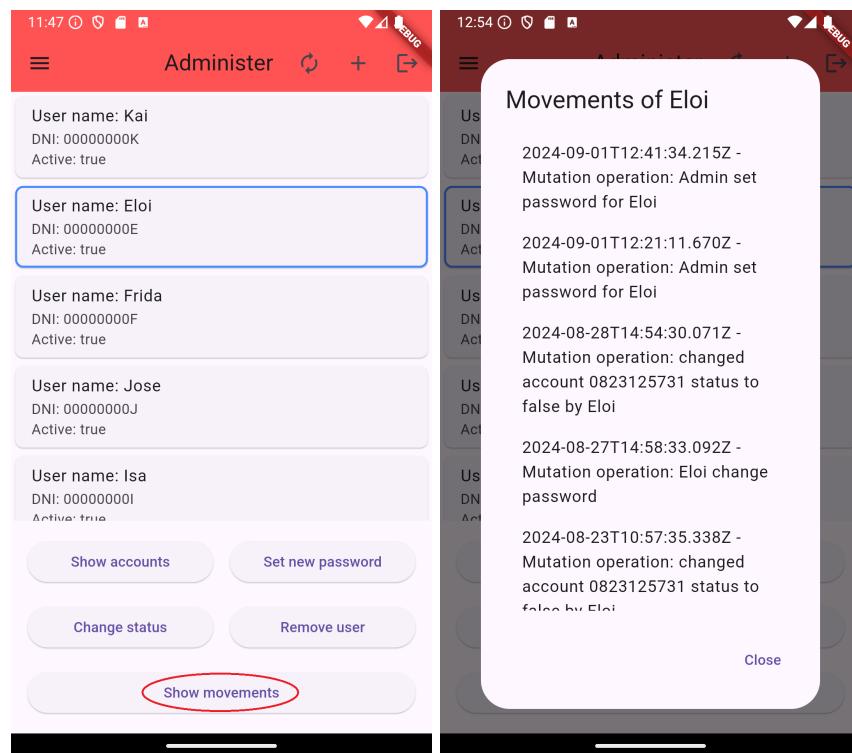


Figura 8.30: Visualitzar els moviments d'un usuari

8.6.2.7 Registrar un nou usuari

Un administrador pot donar d'alta tant un usuari administrador com un usuari client.

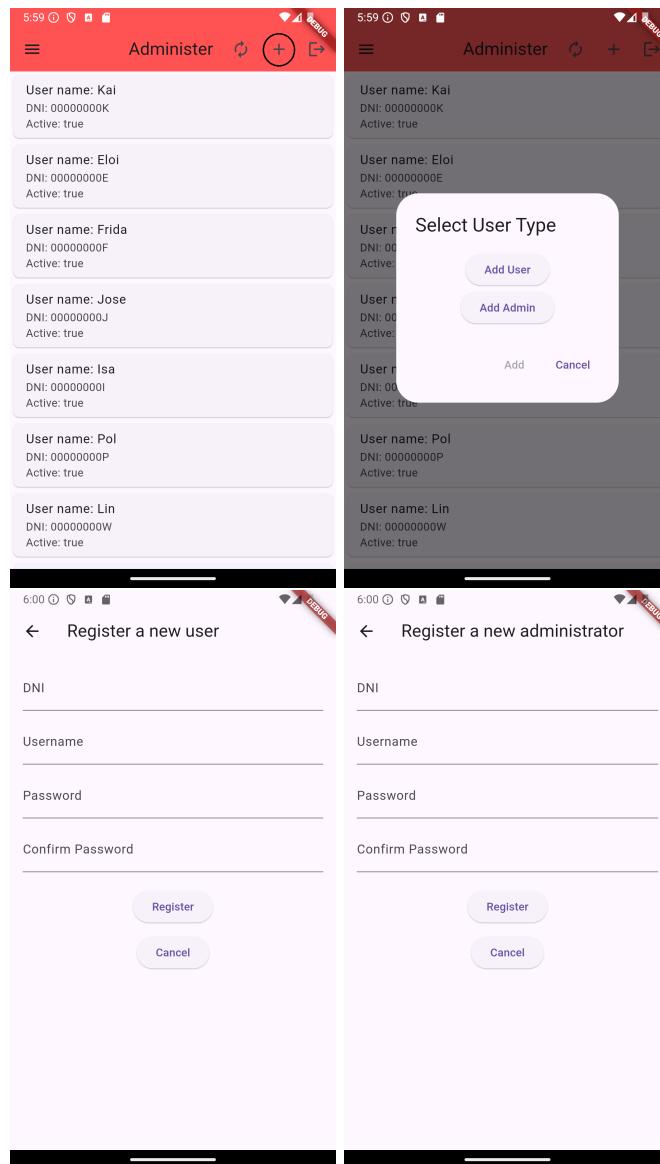


Figura 8.31: Registrar un nou usuari

8.6.2.8 Llistar tots els administradors i veure els seus moviments

Els administradors poden veure les operacions realitzades entre ells.

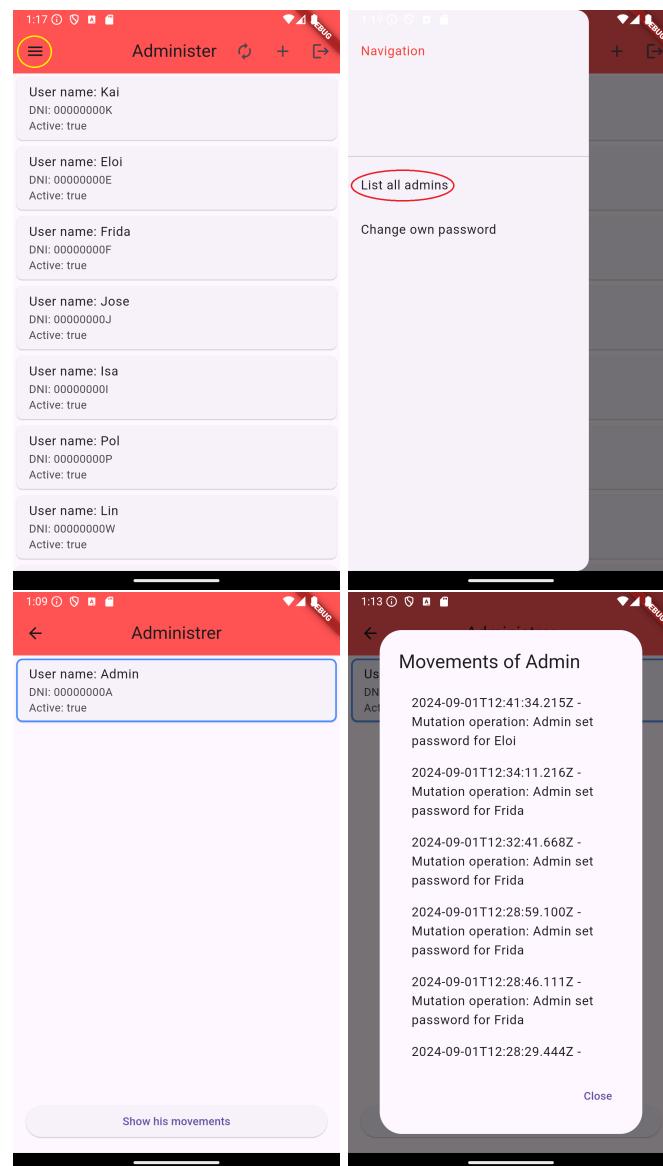


Figura 8.32: Llistar tots els administradors i veure els seus moviments

8.6.2.9 Canviar la pròpia contrasenya

L'usuari administrador pot modificar la seva pròpia contrasenya, però cal introduir la contrasenya antiga. A més, la nova contrasenya ha de complir les normes estableties.

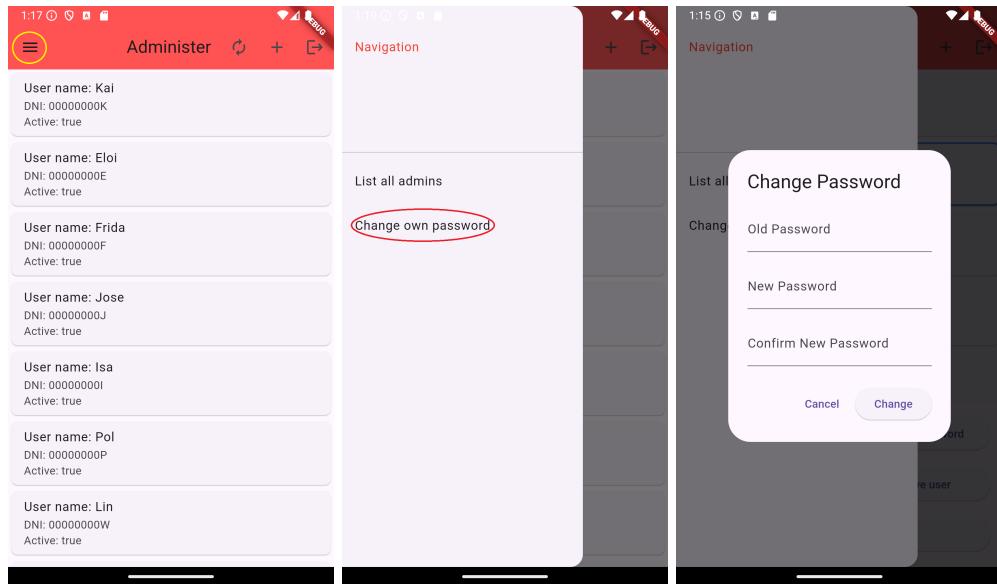


Figura 8.33: Canviar la pròpia contrasenya

CAPÍTOL 9

Implementació i proves

9.1 Introducció

L'objectiu d'aquest capítol és explicar el procés de desenvolupament, des de la idea inicial de l'aplicació i la seva estructura fins al desenvolupament actual de la mateixa. També s'inclourà una explicació sobre com s'ha provat constantment l'aplicació. A més, es detallaran les diferents proves que s'han realitzat per verificar el funcionament de l'aplicació.
(potser millor “També s'explicaran les diferents proves que s'han dut a terme per verificar el funcionament de l'aplicació”).corregit

9.2 Idea inicial (potser millor “Idea inicial”)

corregit

Quan se'm va ocórrer la idea de l'aplicació, vaig acabar amb una barreja de pensaments i conceptes que necessitaven ser organitzats i connectats de manera coherent. El primer pas en aquest procés va ser redactar una explicació detallada de l'aplicació, incloent l'objectiu general, les funcionalitats principals que hauria d'ofrir, així com les limitacions i allò que no inclouria. Va ser essencial intentar cobrir tants detalls com fos possible per assegurar una comprensió clara del projecte.

Durant aquesta fase, vaig analitzar com es relacionava cada component de l'aplicació amb els altres i com s'integraria dins del sistema global. Vaig considerar les possibles interaccions entre les diferents funcionalitats i com aquestes podrien impactar en l'experiència de l'usuari. A més, vaig revisar diversos enfocaments arquitectònics per determinar el més adequat per aconseguir els objectius establerts.

Tot i que probablement canviaria l'estructura en el futur en resposta a noves necessitats o oportunitats de millora, la idea inicial de l'arquitectura va ser útil per establir una base sólida. A partir d'aquesta base, vaig poder començar a especificar els requisits funcionals i no funcionals de manera més detallada. Aquesta etapa va ser probablement la tasca més exigent del projecte, ja que va

requerir una atenció meticulosa a cada aspecte del disseny i l'estructura, a més de l'organització dels paquets de treball. Després que aquestes parts estiguessin definides i estructurades, vaig poder visualitzar un full de ruta clar i detallat per al desenvolupament del sistema, el que em va permetre avançar amb una estratègia ben definida i metòdica.

9.3 Biblioteques utilitzades

Per a aquest projecte, ha estat necessari utilitzar diverses biblioteques que han facilitat el desenvolupament i la implementació de les funcionalitats previstes. Aquestes biblioteques han proporcionat les eines i els recursos necessaris per gestionar diferents aspectes del projecte.

A continuació, es detallen les biblioteques específiques que s'han utilitzat i la seva utilitat en el context del projecte, com ara generar codis QR, llegir codis QR, encriptar missatges, gestionar contrasenyes i administrar credencials.

9.3.1 Generar codi QR

Per generar codis QR en Flutter, una biblioteca popular és [qr_flutter](#). Aquesta biblioteca facilita la creació de codis QR a partir de text i la seva renderització com a widgets dins de l'aplicació Flutter. A més de la seva funcionalitat bàsica, qr_flutter ofereix diverses opcions de personalització, com ara ajustar la mida del codi QR, configurar el color del codi QR i afegir imatges al centre del codi QR.

El problema que vaig trobar en la generació del codi QR es deu a l'actualització de la biblioteca qr_flutter. En les versions 2.0.0 fins a 3.0.0, utilitzàvem la nomenclatura *QrImage*. No obstant això, en la versió actual 4.1.0, s'ha actualitzat a *QrImageView*. Inicialment, això causava errors de compilació perquè el sistema no reconeixia *QrImage*. Em va sorprendre molt, però finalment vaig descobrir que el problema es devia a aquesta actualització.

Tal com es representa a la figura 9.1 es mostren dues imatges que il·lustren el codi per a les dues versions:

<pre>child: QrImage(data: "Aquest és el text que es codificarà en el QR", version: QrVersions.auto, size: 200.0, gapless: false,),</pre>	<pre>child: QrImageView(data: "Aquest és el text que es codificarà en el QR", version: QrVersions.auto, size: 200.0, gapless: false,),</pre>
--	--

Figura 9.1: Versió antiga vs. versió nova

9.3.2 Llegir codi QR

Per llegir codis QR en Flutter, aquest projecte utilitza la biblioteca **qr_code_scanner**. Aquesta biblioteca facilita la lectura de codis QR mitjançant la càmera del dispositiu i permet integrar fàcilment la funcionalitat d'escanneig de codis QR en aplicacions Flutter. Ofereix una solució senzilla i eficient per capturar i processar dades codificades en codis QR.

Vaig trobar un problema, encara que vaig sortir de la pantalla de la càmera, el webcam del portàtil seguia encès. La solució consisteix a utilitzar el controlador de la biblioteca **qr_code_scanner**. Cal controlar el controlador de la càmera i assegurar-se de cridar el mètode *dispose()* quan ja no es necessiti utilitzar la càmera.

9.3.3 Encriptar Missatge

Per encriptar els missatges dels codis QR en flutter, utilitza la biblioteca **encrypt 5.0.3**. Aquesta biblioteca ofereix una àmplia gamma de mètodes d'encriptació, i per a aquesta pràctica s'ha escollit l'algorisme AES.

La biblioteca encrypt proporciona eines robustes per gestionar tant l'encriptació com la desencriptació de dades, facilitant així la protecció de la informació codificada en els codis QR. L'algorisme AES (Advanced Encryption Standard) és un mètode de xifrat simètric que utilitza la mateixa clau per a l'encriptació i la desencriptació, garantint així la seguretat de les dades. Utilitzant AES, podem assegurar que els missatges emmagatzemats en els codis QR siguin segurs i només accessibles per aquells que disposin de la clau correcta per desencriptar-los.

Inicialment, vaig implementar una clau única per a tots els usuaris de l'aplicació. No obstant això, vaig adonar-me que aquesta solució podia ser molt arriscada en cas de pèrdua de la clau. Per garantir la màxima seguretat, he optat per utilitzar una clau independent per a cada compte bancari. A més, a me-sa que es generen nous codis QR de pagament, la clau associada s'actualitza automàticament.

9.3.4 Encriptar la contrasenya

Per guardar les contrasenyes dels usuaris a la base de dades, és molt perillós emmagatzemar-les en text clar. Per garantir la seguretat, aquest projecte utilitza la biblioteca crypto de Node.js. El mòdul crypto de Node.js proporciona eines essencials per a la seguretat de les aplicacions, incloent funcionalitats com

encriptació, hashing, signatura digital i més. En aquest projecte, s'empla la funcionalitat de hashing.

El hashing és una tècnica criptogràfica clau que transforma les dades en una representació fixa i, idealment, irreversible. Aquesta tècnica permet emmagatzemar les contrasenyes de manera segura, verificar la integritat de les dades i assegurar l'autenticitat dels missatges gràcies a les seves propietats úniques.

9.3.5 Gestió de les credencials

En aquest projecte, gestionem les credencials d'usuari utilitzant la biblioteca jsonwebtoken. Aquesta biblioteca proporciona les eines necessàries per treballar amb tokens JWT, que són essencials per a l'autenticació i l'autorització d'usuaris. Les funcions inclouen:

- **Generació de Tokens d'Accés:** La funció `getAccessToken` crea un token JWT signat amb una clau secreta a partir d'un payload proporcionat. Aquest token es configura amb una durada de 1 dia, la qual cosa significa que l'usuari haurà de tornar a autenticar-se després d'aquest període per obtenir un nou token.
- **Verificació de Tokens:** La funció `getUserId` extreu i verifica el token JWT des de l'encapçalament d'autorització de la sol·licitud. Si el token és vàlid, retorna l'identificador de l'usuari emmagatzemat en el token, permetent accedir a recursos protegits.

Aquí es podria explicar també les biblioteques que ha calgut utilitzar per generar i llegir codis QR, com les has integrat a la teva aplicació, problemes que t'has trobat i com els has solucionat, etc. I el mateix per biblioteques o mòduls utilitzats per encriptar missatges.corregit

9.4 Desenvolupament d'aplicació

Vaig començar a desenvolupar el servidor ([vols dir l'API de l'aplicació?](#))corregit, ja que això ajudava a definir la base de dades. La meva estratègia consistia a implementar primer les funcionalitats més crítiques i bàsiques, assegurant-me que cada component essencial funcionés correctament abans de passar a les següents etapes. Per exemple, vaig començar amb l'operació de transacció entre dos comptes bancaris, assegurant-me que aquest procés fonamental fos fiable. Un cop aquesta funcionalitat va estar operativa, vaig afegir seguretat amb l'encriptació de dades. Posteriorment, vaig implementar controls addicionals, com

ara verificar si un compte disposava de saldo suficient i comprovar si el compte estava actiu.

Un cop vaig tenir una base sòlida al servidor, vaig començar a dissenyar la interfície d'usuari. Durant aquest procés, vaig adaptar el disseny en funció dels punts finals disponibles i les funcionalitats implementades en el servidor, mentre identificava les possibles mancances en l'API. Així, després de completar el disseny de la interfície, vaig tornar a l'API per afegir les funcionalitats que faltaven i per netejar el codi existent.

A mesura que avançava en el desenvolupament, vaig començar a treballar en les interfícies d'usuari (**potser millor “vaig començar a treballar en les interfícies d'usuari”**)corregit. Vaig desenvolupar primer l'esquelet bàsica (**aplicació?**)corregit per esquelet, que contenia la major part de la lògica i complexitat. Aquesta fase va ser crucial, ja que vaig afegir diversos components i vistes que posteriorment es reutilitzarien a l'aplicació client. Vaig adoptar un enfocament iteratiu: cada vegada que afegia o ajustava una funcionalitat en el servidor, feia el mateix en el client, variant les funcionalitats de manera sincronitzada. Aquesta estratègia em va permetre assegurar-me que ambdues parts de l'aplicació estiguessin alineades i funcionessin conjuntament.

Tot i que no vaig utilitzar eines específiques com “Nx”, vaig intentar fer abstractació i reutilitzar el codi de manera eficaç per simplificar el desenvolupament. Si en el futur es necessita afegir un altre projecte (**aplicació?**)corregit, el procés serà més senzill, ja que molts components i funcions podrien ser reutilitzats.

9.5 Problemes i Solucions en la Implementació

Durant el desenvolupament del projecte, s'han presentat diversos reptes i obstacles que han afectat la implementació de les funcionalitats previstes. A mesura que avançava en el procés, vaig identificar diversos problemes que requereixen una atenció detallada per garantir el bon funcionament de l'aplicació.

En aquest apartat, es descriuen els problemes més significatius que vaig trobar, juntament amb les estratègies i solucions que vaig aplicar per superar-los. Cada problema s'explica en detall, incloent el context en què va sorgir, l'impacte que va tenir sobre el projecte, i les solucions implementades per resoldre'l. Aquest enfocament proporciona una visió clara dels desafiaments afrontats i de com es va assegurar el bon desenvolupament i la funcionalitat final del projecte.

9.5.1 Android Studio

Vaig enfocar diversos problemes amb Android Studio. Amb el mateix codi, l'aplicació fallava al dia següent. Inicialment, no estava clar si el problema es devia a Android Studio o al codi mateix. Vaig intentar revisar les versions anteriors del codi utilitzant el control de versions, però això no va resoldre el problema.

En la figura 9.2 es mostra un dels errors que he trobat.

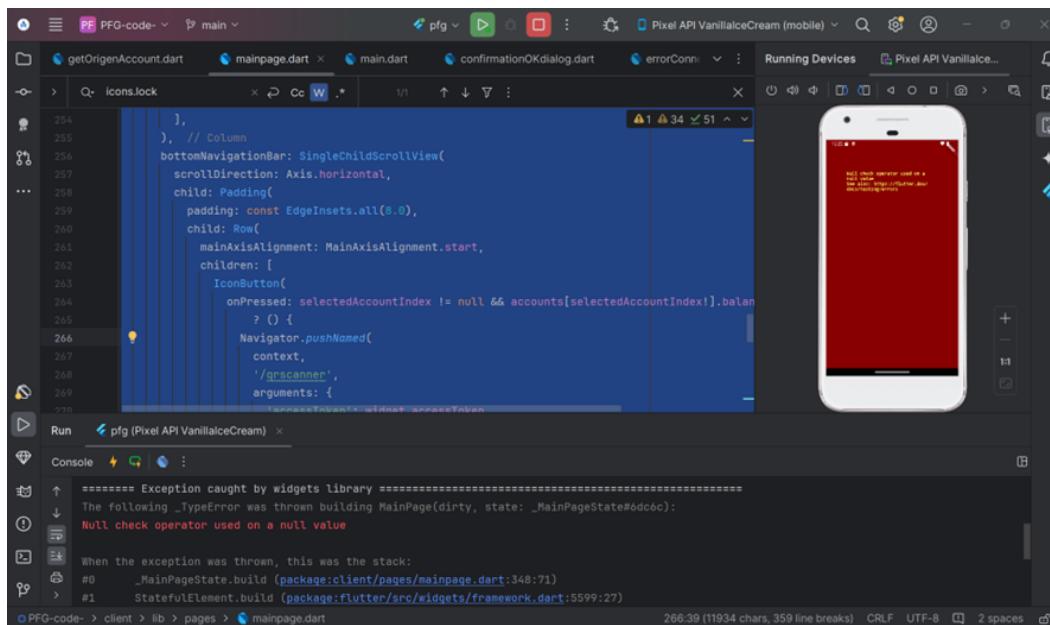


Figura 9.2: Problemes d'implementació amb Android Studio

Més endavant, vaig descobrir que el simulador sovint es pot omplir de memòria, cosa que pot provocar que no aranqui correctament. El simulador té un límit de memòria, i gestionar-ho adequadament és important per garantir que funcioni de manera fluida. Si experimentes problemes de rendiment, cal ajustar la configuració del simulador i fer una neteja periòdica de dades. La solució que faig servir és realitzar un **wipe data** del simulador, el qual neteja les dades emmagatzemades i resol els problemes de càrrega.

Una altra opció que es pot combinar amb el wipe data és crear una pàgina simple. Aquesta mesura permet comprovar si el problema es troba en el simulador o en el codi. Un altre avantatge d'aquesta pàgina simple és que permet realitzar implementacions complexes de manera independent del codi global, cosa que facilita les proves en casos en què no es té la seguretat total sobre la funcionalitat.

La figura 9.3 mostrada a continuació presenta el codi de la pàgina simple (mutation.dart). Aquesta pàgina es carrega canviant la ruta del main.

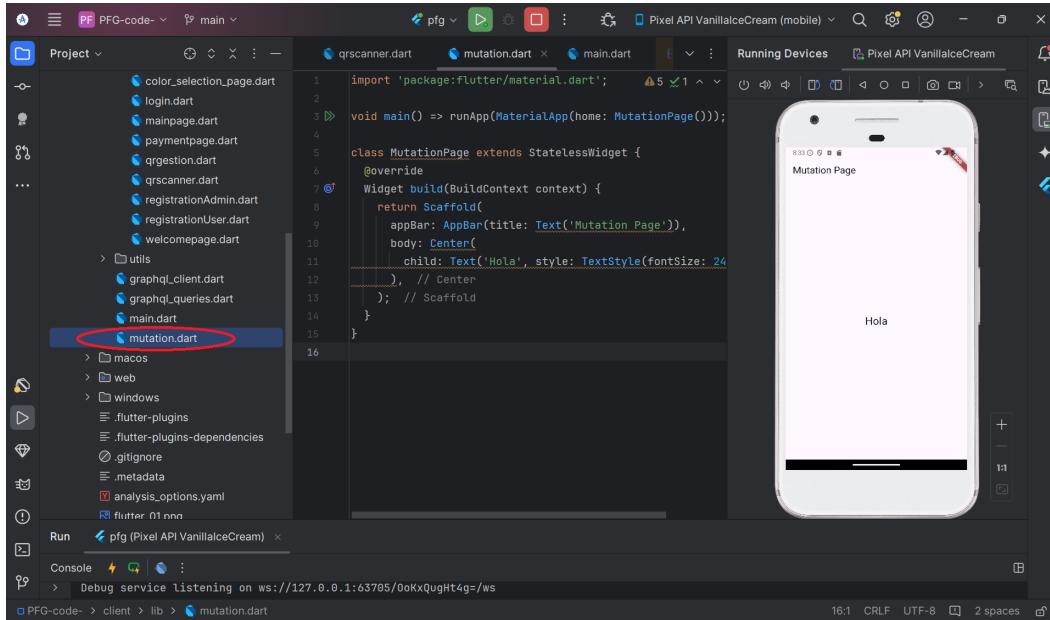


Figura 9.3: Pàgina de prova

9.5.2 Flutter

Si l'error de compilació no es pot resoldre amb les mesures esmentades anteriorment, és probable que el problema estigui relacionat amb el compilador de Flutter.

La comanda **flutter clean** és utilitzat per netejar el directori de construcció i altres fitxers generats que poden causar problemes. Posteriorment cal fer servir **flutter pub get** per tornar a descarregar les dependències. Algunes situacions en què és necessari utilitzar aquesta comanda inclouen:

- Problemes de Compilació
- Actualitzacions de Dependències
- Problemes Amb el Simulador o Dispositiu
- Actualització de Flutter o el SDK

També hi ha la comanda **flutter doctor**, que és utilitzat per verificar la configuració de l'entorn de desenvolupament i solucionar problemes de configuració. Algunes situacions en què és necessari utilitzar aquesta comanda inclouen:

- Verificació de l'Entorn de Desenvolupament
- Solució de Problemes de Configuració
- Abans de Reportar Problemes

Aquesta imatge 9.4 mostra l'execució de flutter doctor, indicant que l'entorn de desenvolupament està configurat correctament.

```

Símbolo del sistema - flutter
Microsoft Windows [Versión 10.0.22631.4112]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\sudan.wu>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.22.3, on Microsoft Windows [Versión 10.0.22631.4112], locale es-ES)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
    X cmdline-tools component is missing
        Run 'path/to/sdkmanager --install "cmdline-tools;latest"'
        See https://developer.android.com/studio/command-line for more details.
    X Android license status unknown.
        Run 'flutter doctor --android-licenses' to accept the SDK licenses.
        See https://flutter.dev/docs/get-started/install/windows#android-setup for more details.
[✓] Chrome - develop for the web
[✗] Visual Studio - develop Windows apps
    X Visual Studio not installed; this is necessary to develop Windows apps.
        Download at https://visualstudio.microsoft.com/downloads/.
        Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2023.3)
[✓] VS Code, 64-bit edition (version 1.92.1)
[✓] Connected device (4 available)
[✓] Network resources

! Doctor found issues in 2 categories.

C:\Users\sudan.wu>

```

Figura 9.4: Comanda flutter doctor

9.5.3 GitHub

Un error greu que vaig cometre en aquest projecte va ser que, quan el codi no arrencava, la primera cosa que feia era anar al repositori i retirar els commits, fent còpies del projecte localment per seguretat. Si, després de retirar un commit, el codi seguia sense funcionar, procedia a retirar-ne un altre, i així successivament.

Més endavant, em vaig adonar que aquest enfocament no era efectiu i em vaig adonar que "això no és possible; el codi de fa una setmana funcionava correctament..."

En aquest punt, em vaig adonar que la millor solució era recuperar els commits, però sovint em trobava bloquejat, ja que havia eliminat commits del repositori local sense sincronitzar-los amb el repositori remot. Això em va portar a perdre temps i a confondre'm en diverses ocasions.

Per solucionar aquest problema, és necessari estudiar a fons les comandes de Git:

9.5.3.1 Eliminar Commits

- *git reset –soft HEAD 1*: Elimina el commit més recent però manté els canvis en l'índex.
- *git reset –hard HEAD 1*: Elimina el commit més recent i descarta els canvis.
- *git revert <commit>*: Desfà els canvis d'un commit específic creant un nou commit que els elimina.
- *git rebase -i HEAD n*: Permet modificar, combinar o eliminar múltiples commits en l'historial.

9.5.3.2 Recuperar Commits

- *git log*: Mostra l'historial de commits i identifica el commit que vols recuperar.
- *git checkout <commit>*: Permet revisar l'estat de l'arbre de treball en el moment d'un commit específic.
- *git cherry-pick <commit>*: Aplica els canvis d'un commit específic a la branca actual.
- *git reflog*: Mostra l'historial de referències i permet recuperar commits que no són visibles en l'historial normal.

9.5.4 MongoDB

Vaig passar més d'un dia barallant-me amb un error aparentment simple. El problema era el següent: tenia creada la taula **users**, però no contenia tots els atributs necessaris. Un dels atributs era el compte bancari, i volia substituir-lo per una llista de comptes bancaris.

Vaig revisar tots els atributs de l'esquema i vaig revisar el codi del resolver una i altra vegada. Vaig anar a MongoDB per esborrar totes les dades anteriors de la taula (sense eliminar la taula mateixa). No hi havia manera que funcionés correctament; només es podia inserir un sol compte bancari, tot i que el servidor indicava que la inserció es realitzava correctament.

Finalment, vaig descobrir que havia d'eliminar la taula sencera per poder efectuar l'actualització, com indica la figura 9.5.

The screenshot shows the Compass MongoDB interface. On the left, there's a sidebar with a '+ Create Database' button and a search bar for namespaces. Below that, the 'pfg' database is expanded, showing collections: accounts, dictionaries, people, transactions, users (which is selected and highlighted in green), sample_mflix, and test. To the right of the sidebar, the main area is titled 'pfg.users'. It displays storage information: STORAGE SIZE: 96KB, LOGICAL DATA SIZE: 37.19KB, TOTAL DOCUMENTS: 11, INDEXES TOTAL S. Below this, there are tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, and Settings. A search bar says 'Generate queries from natural language in Compass'. Under 'Find', there's a 'Filter' section with a placeholder 'Type a query: { field: 'value' }' and a 'Reset' button. At the bottom, it says 'QUERY RESULTS: 1-11 OF 11' and lists one document:

```
_id: ObjectId('66bda455bed284dedd687863')
dni : "00000000A"
name : "Admin"
password : "cadc77d11cdafc5f435cf4b065ff95ea226e2cd17e0a9a5bf154dd"
role : "admin"
```

Figura 9.5: Eliminar la taula de mongoDB

Faltaria indicar quins problemes t'has trobat durant la implementació (sigui el que sigui), i com els has acabat solucionant. corregit

9.6 Proves

Aquesta secció descriu diverses categories de proves que es duen a terme durant el desenvolupament de programari per assegurar la seva qualitat i el correcte funcionament. A continuació, es presenta un breu resum de cada tipus de prova, juntament amb un exemple del projecte:

9.6.1 Proves d'unitat

Aquestes proves se centren a verificar el correcte funcionament de les parts més petites del codi, com ara funcions o mètodes, de manera independent.

Un exemple és la generació de contrasenyes, on es realitza el procés de hashing i es guarden les contrasenyes a la base de dades de manera codificada. La figura 9.6 de sota mostra el resultat.



Figura 9.6: Les contrasenyes codificades

9.6.2 Proves d'integració

L'objectiu d'aquestes proves és assegurar que les diferents unitats de codi treballin bé conjuntament quan es combinen, garantint així el bon funcionament del sistema en el seu conjunt.

Escanejar un codi QR de cobrament seria un cas de prova d'integració. Per part del client, es poden obtenir les dades necessàries per fer la transacció, que després es transmeten al servidor. El servidor processa la transacció i retorna una resposta que el client ha de mostrar.

La figura 9.7 mostra com apareix el client i els logs del servidor en el cas que tot hagi anat bé.

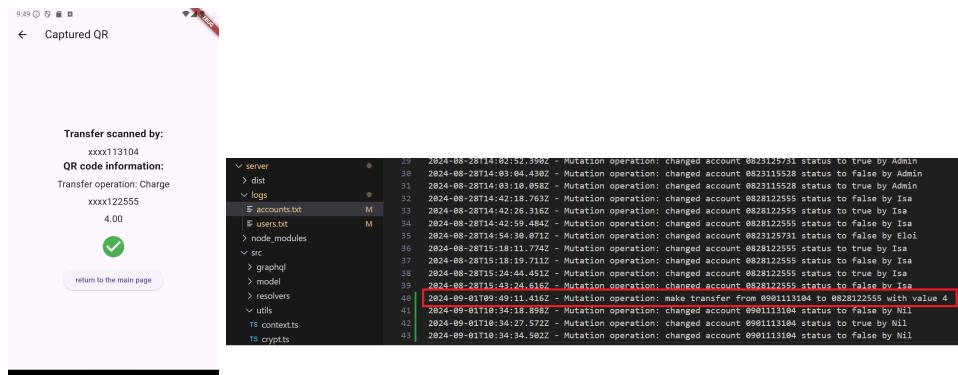


Figura 9.7: Resolució d'una transferència QR

9.6.3 Proves de regressió

Aquestes proves es realitzen per garantir que les noves funcionalitats o canvis no han introduït errors en altres parts del sistema. Es tornen a executar proves prèvies per assegurar que el comportament del sistema no ha canviat de manera inesperada.

En el cas d'aquest projecte, per implementar el codi QR de pagament o cobrament, primer treballo amb missatges en clar. Un cop verificat que tot funciona correctament, incorporo el missatge xifrat només per al codi QR de pagament. Així, comprovo que el procés de xifrat i desxifrat no afecta el procés d'operació.

La figura 9.8 mostra un codi QR de pagament i el missatge integrat.

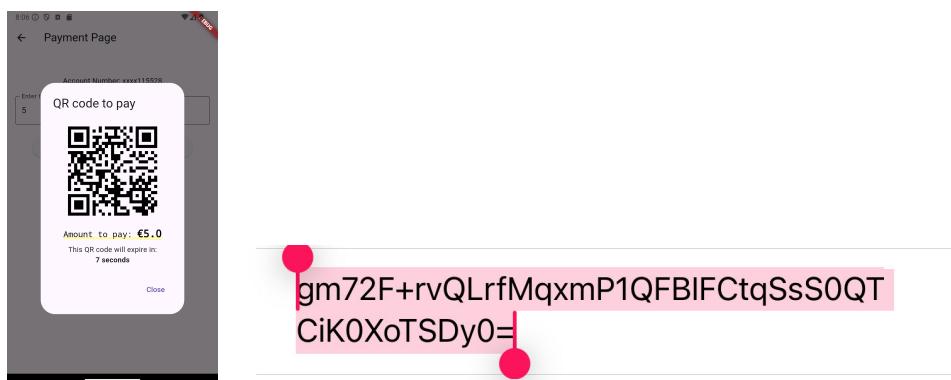


Figura 9.8: Xifratge del codi QR de pagament

9.6.4 Proves de sistema

Aquestes proves verifiquen que tot el sistema compleixi amb els requisits establerts, utilitzant escenaris d'ús que reflecteixen situacions reals.

9.6.5 Altres proves

No se que puc explicar d'aquí?

- **Proves d'acceptació de l'usuari:** Aquestes proves són realitzades per usuaris finals en un entorn que simula el real, per obtenir el seu *feedback* sobre la usabilitat i l'adequació de les noves funcionalitats.

Aquests 3 de sota no he estudiat

- **Proves de càrrega i rendiment:** Es realitzen per avaluar com es comporta el sistema sota condicions de càrrega elevades i diferents situacions de rendiment, ajudant a identificar possibles problemes i millores necessàries.
- **Proves de seguretat:** Aquestes proves tenen l'objectiu de verificar que el sistema és segur i està protegit contra vulnerabilitats i atacs potencials.
- **Proves de compatibilitat:** S'utilitzen per assegurar que el sistema funciona correctament en diverses plataformes, navegadors, dispositius i sistemes operatius.

L'aplicació de la interfície d'usuari (IU) s'ha provat manualment per trobar errors. En el cas de l'API, provar els punts finals ha estat una mica més difícil, però s'ha utilitzat programari especialitzat per fer-ho. Pel que fa a les aplicacions d'interfície d'usuari, s'han provat casos avançats executant l'aplicació. Quan s'ha trobat un error, primer he revisat les proves per veure si el cas no estava cobert o si va ser un error en la prova, ja que és una situació que pot passar. Si es troava un error en les proves, actualitzava les proves i després corregia el codi real. Si no es troava un error en les proves, buscava l'error en el codi i després el cobria amb les proves.

Seria interessant posar un exemple del teu propi projecte per cada tipus de proves llistat abans.

CAPÍTOL 10

Implantació i resultats

10.1 Implantació

La implantació de l'aplicació es realitza en diverses fases, amb un enfocament gradual per assegurar la qualitat i funcionalitat del sistema. La fase inicial del projecte va consistir en desenvolupar l'aplicació de pagament mitjançant codi QR. Aquesta primera etapa es va centrar en establir l'infraestructura bàsica necessària per generar i escanejar codis QR associats a transaccions de pagament.

Un cop l'aplicació va estar operativa amb les funcionalitats bàsiques, la següent etapa va ser la incorporació d'una passarel·la de pagament. Aquesta passarel·la és crucial per processar les transaccions de pagament de manera segura i eficient. Actualment, el desenvolupament de la passarel·la està en curs, i la seva integració és el següent pas important per completar la implantació de l'aplicació.

10.1.1 Opcions per a la Implantació Futura

El projecte presenta dues vies potencials per a la seva implantació futura:

- 1. Integració amb Sistemes Bancaris Estatals:** La primera opció és integrar l'aplicació amb els sistemes bancaris estatals, seguint un model similar al de Bizum. Aquesta integració permetria que l'aplicació ofereixi funcionalitats comparables a les d'un servei de pagament basat en la infraestructura bancària existent. Aquest enfocament facilitarià la interoperabilitat amb els sistemes de pagament establerts i podria beneficiar-se de la confiança ja existent en el sistema bancari.
- 2. Desenvolupament d'una Passarel·la de Pagament Propria:** La segona opció és crear una passarel·la de pagament pròpia, desenvolupant una solució independent similar a PayPal. Aquesta opció permetria una major autonomia en la gestió de les transaccions, oferint una flexibilitat més gran en la configuració i adaptació dels serveis de pagament. A més, operant com una empresa de pagaments independent, es podria innovar i adaptar els serveis segons les necessitats específiques del mercat.

10.1.2 Mesures de Protecció de Dades i Compliment Normatiu

En el moment de la implantació, és essencial garantir la protecció de dades dels usuaris i complir amb les normatives vigents. Per aquest motiu, s'han establert les següents mesures clau:

- **Cumpliment amb el Reglament General de Protecció de Dades (GDPR):**
 - *Seguretat de Dades Personals:* Es garantirà que les dades personals dels usuaris estiguin protegides mitjançant xifratge en trànsit i en repòs.
 - *Consentiment i Transparència:* S'obtenirà el consentiment explícit dels usuaris abans de recollir dades, oferint informació clara sobre l'ús que se'n farà.
 - *Drets dels Usuaris:* Els usuaris podran accedir, modificar o eliminar les seves dades a través d'una interfície fàcil d'usar.
- **Seguretat en el Processament de Pagaments:**
 - *Protecció de Transaccions:* Les dades de les transaccions seran xifrades per prevenir interceptacions o manipulacions.
 - *Compliment amb PCI-DSS:* La passarel·la de pagament complirà amb els estàndards PCI-DSS per assegurar la protecció de les dades de targeta de crèdit.
- **Gestió de Riscos i Auditories:**
 - *Auditories de Seguretat:* Es realitzaran auditories periòdiques per detectar i corregir vulnerabilitats.
 - *Plans de Resposta a Incidents:* S'implementaran plans per gestionar eficaçment qualsevol incidència de seguretat.
- **Compliment Normatiu Local i Internacional:**
 - *Normatives Locals:* Es complirà amb les lleis locals relatives a la protecció de dades i transaccions financeres.
 - *Regulacions Internacionals:* Si l'aplicació opera internacionalment, s'adaptarà a les regulacions de protecció de dades dels diferents països.

També hauries de parlar de quines mesures has pres o prendries en el moment de la implantació pel que fa a garantir la protecció de dades (i, per tant, la legislació i normatives vigents).corregit

10.2 Resultats

En aquesta secció es presenta l'aplicació final amb detall, mitjançant captures de pantalla que il·lustren les diferents funcionalitats. Es descriu com cada una d'aquestes pantalles contribueix a assolir els objectius plantejats inicialment, proporcionant una visió clara de l'eficàcia de l'aplicació en complir amb els requisits establerts.

10.2.1 La pàgina principal de l'usuari client

La pàgina principal del client s'ha dissenyat per oferir una experiència d'usuari intuïtiva i eficaç, amb un accés ràpid a les funcions més importants. A continuació es detallen les funcionalitats clau de la pàgina i com cada una d'elles contribueix als objectius del projecte:

- **Accés Ràpid a Funcions Clau:** A la part inferior de la pàgina, es presenten botons d'accés ràpid a les funcions essencials, com la càmera escanejadora, el generador de codi QR de pagament i cobrament, així com altres opcions rellevants. Aquest disseny permet als usuaris accedir fàcilment a les funcionalitats més utilitzades, millorant l'eficiència i reduint el temps necessari per completar les tasques habituals.
- **Botó per Sortir:** El botó per sortir de l'aplicació està ubicat en una posició destacada per garantir que els usuaris puguin tancar la sessió de manera segura i ràpida quan acabin d'utilitzar l'aplicació. Això ajuda a protegir la informació personal i garantir la seguretat de les dades.
- **Botó per Afegir Comptes Bancaris:** Aquesta funcionalitat permet als usuaris afegir i gestionar els seus comptes bancaris directament des de la pàgina principal. Amb un accés fàcil a aquesta opció, els usuaris poden vincular els seus comptes per realitzar pagaments o rebre transferències sense complicacions.
- **Navegador per Gestió del Propi Usuari:** Un navegador dedicat facilita la gestió de les dades personals i preferències de l'usuari. Aquest navegador permet accedir a opcions com la modificació de dades personals, proporcionant una manera eficient de gestionar la informació de l'usuari.

Cada una d'aquestes funcionalitats està dissenyada per contribuir a una experiència d'usuari fluida i eficaç, assegurant que els clients puguin gestionar les seves transaccions i dades personals amb facilitat. Aquest enfocament ajuda a aconseguir els objectius del projecte i a maximitzar la satisfacció dels usuaris.

El disseny final es mostra en la següent figura 10.2:

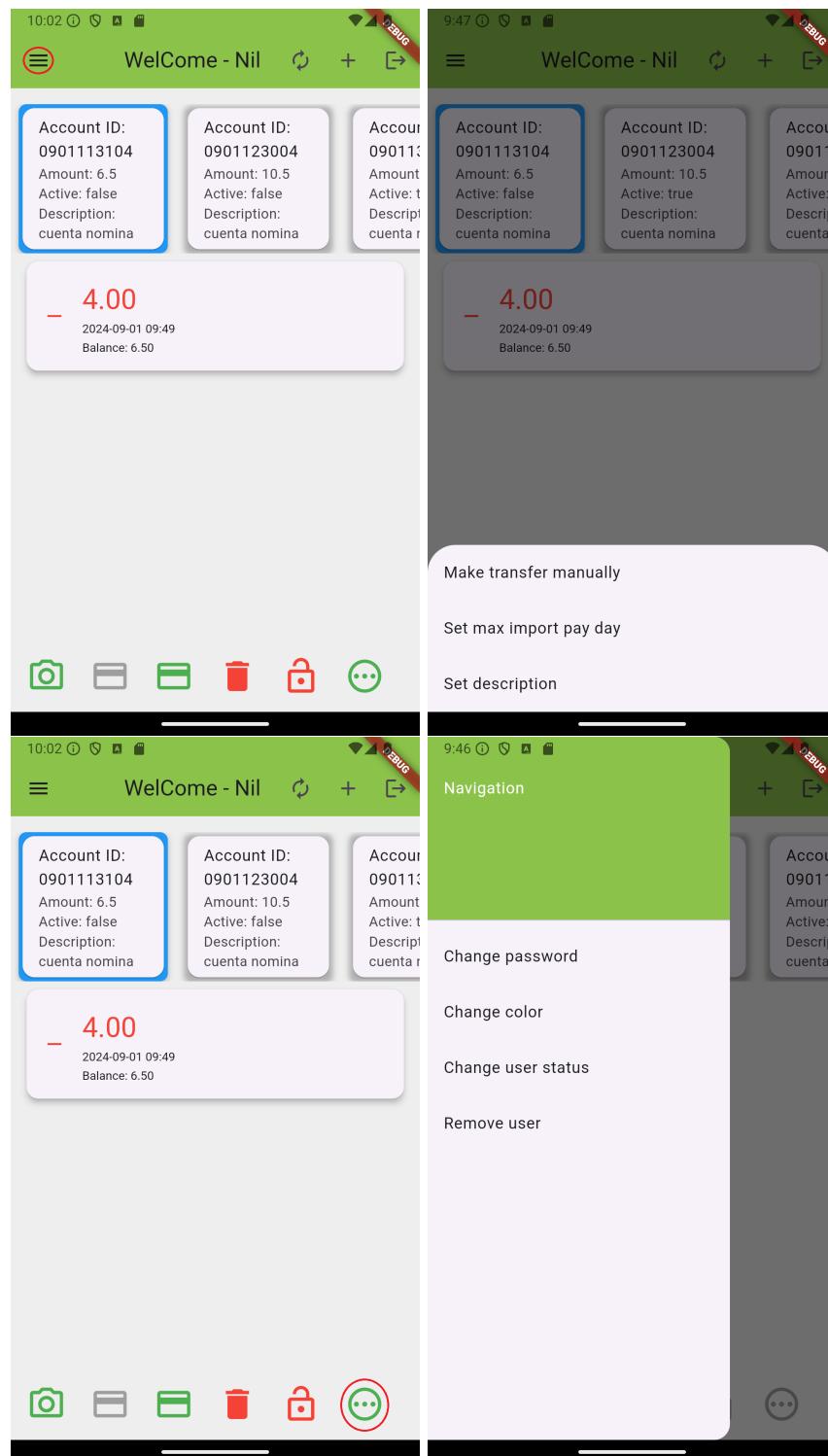


Figura 10.1: La pàgina principal de l'usuari client

10.2.2 La pàgina principal de l'usuari administrador

En aquesta secció es descriu la pàgina principal de l'administrador, amb un enfoquament especial en les funcionalitats que permeten la gestió eficient del sistema. A continuació es detallen les opcions disponibles i com aquestes contribueixen a la supervisió i administració del sistema:

- **Accés Ràpid a Funcions Clau:** Un cop seleccionat el usuari client, a la part inferior de la pàgina, es presenten botons d'accés ràpid a les funcions essencials, com gestionar estat del usuari, modificar contrasenya, veure els seus moviments, així com altres opcions rellevants. Aquest disseny permet als usuaris accedir fàcilment a les funcionalitats més utilitzades, millorant l'eficiència i reduint el temps necessari per completar les tasques habituals. Els administradors poden afegir, editar o eliminar comptes d'usuaris des de la pàgina principal.
- **Botó per Sortir:** El botó per sortir de l'aplicació està ubicat en una posició destacada per garantir que els usuaris puguin tancar la sessió de manera segura i ràpida quan acabin d'utilitzar l'aplicació. Això ajuda a protegir la informació personal i garantir la seguretat de les dades.
- **Navegador per Gestió dels administradors:** Un navegador dedicat facilita la gestió de les dades personals i veure tots els administradors. Aquest navegador permet accedir a opcions com la modificació de dades personals, proporcionant una manera eficient de gestionar la informació de l'usuari.

Cada una d'aquestes funcionalitats ha estat dissenyada per ajudar els administradors a gestionar eficaçment el sistema i a assegurar-se que els objectius operatius es compleixin. La pàgina principal proporciona una interfície centralitzada i organitzada, facilitant la tasca de supervisió i administració del sistema.

El disseny final es mostra en la següent figura 10.2:

Tot això que has posat ja ho has mig explicat a la secció anterior i, en tot cas, seria més pertinent posar-ho a "treball futur". Aquí a resultats has de mostrar com ha quedat l'aplicació final, amb captures de les diferents pantalles, i explicant com cadascuna assoleix els objectius plantejats inicialment. Les captures han de ser les finals, mentre que al Capítol 8 hi ha d'anar només el disseny inicial de les pantalles (esbossos).en les captures del Capítol 8 jo ho són finals...

Tot això de les proves no hi ha d'anar! Ja les expliques al capítol anterior.corregit

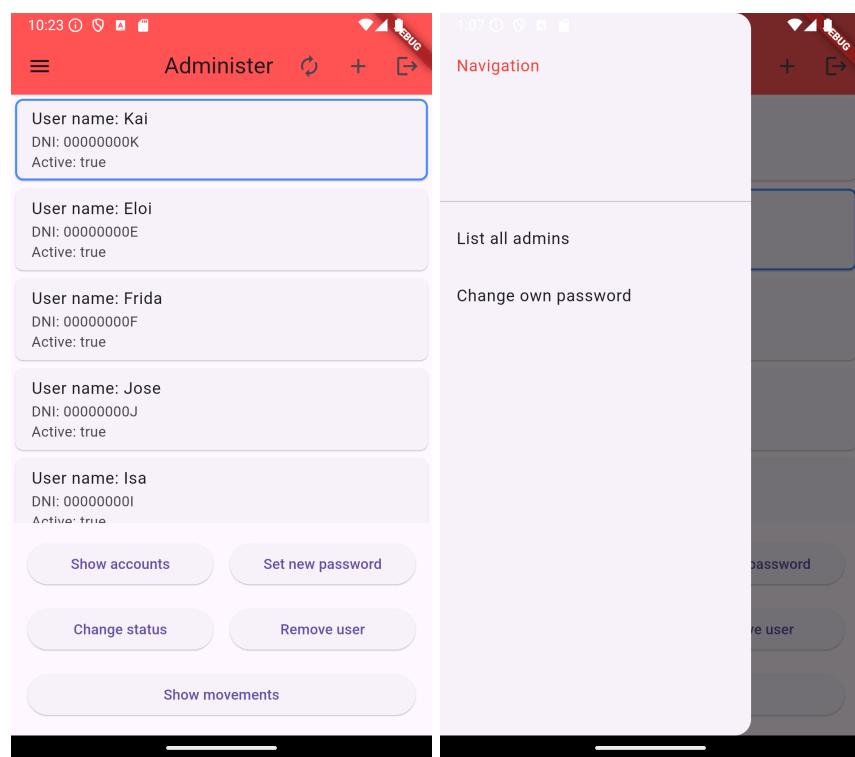


Figura 10.2: La pàgina principal de l'usuari client

CAPÍTOL 11

Conclusions

L'objectiu d'aquest projecte és permetre que les persones puguin realitzar operacions bancàries amb la màxima flexibilitat i seguretat possible.

Recentment vaig visitar una sala de jocs on vaig observar una millora en el sistema de pagament: les antigues monedes han estat substituïdes per mini TPV (Terminals Punt de Venda) a cada màquina. No obstant això, aquest canvi implica una inversió en infraestructura. Si pogués utilitzant codis QR, cada màquina podria mostrar l'import a pagar amb un simple codi QR i els clients podrien realitzar el pagament escanejant el codi QR amb el seu mòbil. Aquesta solució seria més flexible i eficient tant per a les empreses com per als usuaris. Això redueix la necessitat d'infraestructura física com ara TPVs i minimitza els problemes associats amb el seu manteniment, com ara problemes de bateria, cobertura o maquinari defectuós.

Tanmateix, per adoptar amb èxit aquesta tecnologia, cal avançar en diversos aspectes. En primer lloc, és essencial garantir la seguretat de les transaccions. El procés de pagament amb codi QR ha d'incloure mesures de seguretat robustes, com ara l'encriptació de dades i l'autenticació de dos factors, per protegir les dades dels usuaris i prevenir frauds.

A més, la implementació d'una passarel·la de pagament fiable i eficient és crucial. Això pot implicar la integració amb sistemes de pagament establerts, com ara Bizum, o el desenvolupament d'una solució pròpia, similar a PayPal. La passarel·la de pagament ha de ser capaç de gestionar les transaccions de manera fluïda i assegurar la compatibilitat amb diversos dispositius i aplicacions.

L'aplicació presentada en aquest projecte aprofita els avantatges dels codis QR per oferir una solució de pagament innovadora i eficient. A continuació, es detallen els principals beneficis i les funcionalitats clau que aporta l'aplicació:

- **Facilitat d'Ús:** Els codis QR permeten als usuaris realitzar pagaments simplement escanejant el codi amb el seu dispositiu mòbil. Aquesta funcionalitat simplifica el procés de pagament, reduint la necessitat d'introduir

manualment dades com números de targeta o informació de compte bancari.

- **Seguretat:** L'aplicació utilitza xifratge per garantir que les dades transmeses a través dels codis QR estiguin protegides. Això redueix el risc de frau i garanteix que les transaccions siguin segures.
- **Eficiència en les Transaccions:** Mitjançant l'ús de codis QR, les transaccions es poden completar ràpidament, reduint els temps d'espera tant per als clients com per als comerciants. Això millora l'experiència de l'usuari i agilita el procés de pagament.
- **Integració Simple:** L'aplicació està dissenyada per integrar-se fàcilment amb sistemes de pagament existents i amb les passarel·les de pagament. Aquesta flexibilitat facilita la seva adopció tant per a petits comerciants com per a grans empreses.
- **Adaptabilitat i Flexibilitat:** Els codis QR poden ser generats i utilitzats per una àmplia gamma d'aplicacions, des de pagaments en comerços fins a transaccions en línia. Això ofereix una gran flexibilitat en la manera en què els clients poden utilitzar l'aplicació.
- **Reducció d'Errors Humans:** La digitalització del procés de pagament mitjançant codis QR minimitza els errors que poden ocórrer amb els pagaments manuals. Això ajuda a garantir que les transaccions es registrin correctament i es processin sense problemes.
- **Sostenibilitat:** L'ús de codis QR per a pagaments contribueix a una menor necessitat de paper, ja que les transaccions es poden realitzar de manera completament digital. Aquesta característica és especialment beneficiosa per a la sostenibilitat ambiental.

En resum, els pagaments mitjançant codi QR poden revolucionar la forma en què es realitzen les transaccions comercials, oferint una solució més flexible i eficient. L'aplicació aprofita aquestes capacitats per oferir un procés de pagament senzill i segur, millorant l'experiència tant per a usuaris com per a comerciants. Encara que presenten nous reptes, especialment en termes de seguretat, la innovació tecnològica continuada ajudarà a superar aquests obstacles. A més, aquest enfocament proporciona beneficis pràctics i estratègics, adaptant-se a les necessitats de totes les parts implicades i millorant l'experiència de pagament en general.

Faltaria dir què aporta l'aplicació presentada en aquest projecte. Es diu que l'ús de codis QR és beneficiós, però no s'acaba de dir que l'aplicació que es presenta implementa tots aquests avantatges. corregit??

CAPÍTOL 12

Treball futur

12.1 Introducció

Fins al moment, l'aplicació ha aconseguit implementar les funcionalitats bàsiques necessàries per generar i escanejar codis QR. No obstant això, encara falta integrar la passarel·la de pagament per completar el procés de pagament. La principal dificultat actual és la integració amb la passarel·la, que requereix una coordinació complexa amb els proveïdors de serveis de pagament i la implementació de mesures de seguretat addicionals.

Els beneficis esperats de l'aplicació són significatius. Un cop completada, l'aplicació permetrà als usuaris realitzar pagaments de manera senzilla i segura utilitzant codis QR, amb opcions tant per integrar-se amb bancs com per utilitzar una passarel·la de pagament pròpia. Aquesta flexibilitat en els mètodes de pagament oferirà una alternativa valiosa a serveis establerts com PayPal i proporcionarà als usuaris una solució innovadora i eficient per als seus pagaments.

Els pròxims passos en el desenvolupament inclouen la finalització de la integració de la passarel·la de pagament i la implementació de les funcionalitats avançades.

Estaria bé un petit text introductorí dient que s'ha dividit el treball futur en tres seccions.
corregit

He dividit el treball futur en tres seccions: Funcionalitats de l'aplicació, Seguretat de dades i Millora del codi. Es explicarà més detalladament cada secció.

12.2 Funcionalitats de l'aplicació

Com a funcionalitats futures, tot i que no són prioritàries en aquesta fase inicial del projecte, el sistema podria incloure:

- Una pàgina d'anàlisi que proporcioni informació detallada sobre les transaccions realitzades, ajudant els comerciants a gestionar millor les seves vendes.

- Un sistema de subscripcions que permeti als clients realitzar pagaments recurrents de manera automàtica.
- Millores en els aspectes visuals per a una millor experiència d'usuari.
- Incorporar notificacions per veu quan es realitzi una transacció bancària, millorant així la facilitat d'ús i el seguiment en temps real.

Aquestes funcionalitats addicionals estan pensades per ampliar les capacitats del sistema i oferir un valor afegit als usuaris, contribuint així a una experiència de pagament més fluïda i moderna.

12.3 Seguretat de dades

Un aspecte clau a destacar és el sistema d'encriptació. Actualment he utilitzat l'encriptació AES (Advanced Encryption Standard), que és un algorisme de xi-frat simètric àmpliament utilitzat per la seva eficàcia i seguretat. No obstant això, considero que seria molt més avantatjós implementar l'encriptació RSA (Rivest-Shamir-Adleman), que és un sistema d'encriptació asimètric. L'encriptació RSA utilitza un parell de claus, una pública i una privada, per assegurar les dades, oferint un nivell de seguretat addicional en comparació amb AES. Això permet una gestió més segura de les claus i millora la protecció de les transaccions en línia.

12.4 Millora del codi

En el codi actual hi ha molts aspectes que es poden millorar, així com nombrosos fragments de codi repetits que es podrien optimitzar. En el futur, es recomana refactoritzar el codi per millorar la seva eficiència i manteniment. També seria molt important afegir tests per validar el codi freqüentment, utilitzant *frameworks* com Nx.

El capítol de Bibliografia ja surt sol a sota, no cal posar-lo explícitament. Parlant de la bibliografia, n'hauries de posar una mica: el més habitual són pàgines web, ja sigui d'alternatives que has comparat com Bizum o PayPal, d'eines que has utilitzat per desenvolupar l'aplicació, o altres fonts d'informació. De totes maneres, també pots citar llibres i articles. Pots veure com fer-ho a la plantilla de LaTeX que et vaig passar al principi. Si no, m'ho demanes!

CAPÍTOL 13

Bibliografia

[1] *Overleaf*, <https://www.overleaf.com> (No citat.)

no en se treure el (no citat).

ANNEX A

Pressupost

Curabitur ut sodales sapien. Etiam eget ultrices risus, in dignissim nunc. Quisque quis tortor in nunc posuere lacinia ut sed dui. Praesent ut sollicitudin diam, ut mattis magna. Morbi porttitor fermentum magna a pharetra. Nullam at magna diam. Suspendisse vehicula tellus eget ligula aliquam semper. Integer sed ullamcorper felis, ac imperdiet elit. Praesent eu suscipit ligula, sit amet vulputate erat. Etiam eget tempor est, vitae aliquet tortor. Nam efficitur tristique ligula. Aliquam blandit leo non ante suscipit, vitae mattis diam rhoncus. Ut ac dui sit amet dui venenatis suscipit.

Nunc sollicitudin hendrerit risus, quis ultricies orci elementum non. Aliquam erat volutpat. Mauris neque turpis, molestie in tellus id, pharetra gravida mi. Suspendisse potenti. Donec aliquam dolor eu pellentesque auctor. Proin eget sapien ut tellus maximus lacinia. Praesent blandit pretium mi, suscipit sollicitudin eros iaculis in. Nunc a justo sit amet mauris auctor posuere sit amet vel erat. Etiam in maximus nisi. Pellentesque blandit pharetra lectus nec efficitur. Praesent tristique vel arcu id bibendum. In eget eros fringilla magna hendrerit facilisis ac vitae urna. Donec malesuada fermentum dictum. Pellentesque aliquet tortor vitae suscipit tempus. Phasellus ornare risus mi, vel placerat justo efficitur eu. Phasellus quis tincidunt enim.

Etiam a elementum lorem. Integer ac lectus hendrerit, venenatis ante vitae, accumsan eros. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur non varius dolor. Donec suscipit metus vitae ultrices sagittis. Praesent ac dapibus justo, vel mollis sapien. Pellentesque at iaculis neque. Donec ac tellus orci. Pellentesque eleifend fringilla massa, in lobortis nibh finibus in. Morbi ut neque vitae est congue tempor efficitur imperdiet

dolor. Pellentesque nec nibh nec nibh fringilla laoreet. Duis fermentum ornare sollicitudin. Maecenas finibus tincidunt justo commodo commodo. Suspendisse venenatis odio dignissim auctor elementum. Duis non mauris orci.

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc congue mattis tellus, in rhoncus nisl laoreet vel. Sed nulla libero, tincidunt id ligula congue, hendrerit maximus mi. Morbi nec metus in urna imperdiet mattis pretium a libero. Fusce est mauris, convallis id facilisis faucibus, consequat aliquet eros. Vivamus et dui pulvinar, rutrum velit at, volutpat tellus. Nulla vehicula ullamcorper justo, blandit interdum leo sagittis quis. Quisque convallis vel ante ac rutrum. Morbi et varius sem, sed tristique elit. Vestibulum aliquam facilisis pellentesque. Suspendisse consequat commodo eros, sit amet tincidunt sapien semper in. Nunc ut magna ac quam tempor malesuada et at ex. Donec mattis mauris ante, id condimentum elit dapibus at.

 Curabitur ut sodales sapien. Etiam eget ultrices risus, in dignissim nunc. Quisque quis tortor in nunc posuere lacinia ut sed dui. Praesent ut sollicitudin diam, ut mattis magna. Morbi porttitor fermentum magna a pharetra. Nullam at magna diam. Suspendisse vehicula tellus eget ligula aliquam semper. Integer sed ullamcorper felis, ac imperdiet elit. Praesent eu suscipit ligula, sit amet vulputate erat. Etiam eget tempor est, vitae aliquet tortor. Nam efficitur tristique ligula. Aliquam blandit leo non ante suscipit, vitae mattis diam rhoncus. Ut ac dui sit amet dui venenatis suscipit.

 Nunc sollicitudin hendrerit risus, quis ultricies orci elementum non. Aliquam erat volutpat. Mauris neque turpis, molestie in tellus id, pharetra gravida mi. Suspendisse potenti. Donec aliquam dolor eu pellentesque auctor. Proin eget sapien ut tellus maximus lacinia. Praesent blandit pretium mi, suscipit sollicitudin eros iaculis in. Nunc a justo sit amet mauris auctor posuere sit amet vel erat. Etiam in maximus nisi. Pellentesque blandit pharetra lectus nec efficitur. Praesent tristique vel arcu id bibendum. In eget eros fringilla magna hendrerit facilisis ac vitae urna. Donec malesuada fermentum dictum. Pellentesque aliquet tortor vitae suscipit tempus. Phasellus ornare risus mi, vel placerat justo efficitur eu. Phasellus quis tincidunt enim.

 Etiam a elementum lorem. Integer ac lectus hendrerit, venenatis ante vitae, accumsan eros. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur non varius dolor. Donec suscipit metus vitae ultrices sagittis. Praesent ac dapibus justo, vel mollis sapien. Pellentesque at iaculis neque. Donec ac tellus orci. Pellentesque eleifend fringilla massa, in lobortis nibh finibus in. Morbi ut neque vitae est congue tempor efficitur imperdiet dolor. Pellentesque nec nibh nec nibh fringilla laoreet. Duis fermentum ornare sollicitudin. Maecenas finibus tincidunt justo commodo commodo. Suspendisse venenatis odio dignissim auctor elementum. Duis non mauris orci.

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc congue mattis

tellus, in rhoncus nisl laoreet vel. Sed nulla libero, tincidunt id ligula congue, hendrerit maximus mi. Morbi nec metus in urna imperdiet mattis pretium a libero. Fusce est mauris, convallis id facilisis faucibus, consequat aliquet eros. Vivamus et dui pulvinar, rutrum velit at, volutpat tellus. Nulla vehicula ullamcorper justo, blandit interdum leo sagittis quis. Quisque convallis vel ante ac rutrum. Morbi et varius sem, sed tristique elit. Vestibulum aliquam facilisis pellentesque. Suspendisse consequat commodo eros, sit amet tincidunt sapien semper in. Nunc ut magna ac quam tempor malesuada et at ex. Donec mattis mauris ante, id condimentum elit dapibus at.

Curabitur ut sodales sapien. Etiam eget ultrices risus, in dignissim nunc. Quisque quis tortor in nunc posuere lacinia ut sed dui. Praesent ut sollicitudin diam, ut mattis magna. Morbi porttitor fermentum magna a pharetra. Nullam at magna diam. Suspendisse vehicula tellus eget ligula aliquam semper. Integer sed ullamcorper felis, ac imperdiet elit. Praesent eu suscipit ligula, sit amet vulputate erat. Etiam eget tempor est, vitae aliquet tortor. Nam efficitur tristique ligula. Aliquam blandit leo non ante suscipit, vitae mattis diam rhoncus. Ut ac dui sit amet dui venenatis suscipit.

Nunc sollicitudin hendrerit risus, quis ultricies orci elementum non. Aliquam erat volutpat. Mauris neque turpis, molestie in tellus id, pharetra gravida mi. Suspendisse potenti. Donec aliquam dolor eu pellentesque auctor. Proin eget sapien ut tellus maximus lacinia. Praesent blandit pretium mi, suscipit sollicitudin eros iaculis in. Nunc a justo sit amet mauris auctor posuere sit amet vel erat. Etiam in maximus nisi. Pellentesque blandit pharetra lectus nec efficitur. Praesent tristique vel arcu id bibendum. In eget eros fringilla magna hendrerit facilisis ac vitae urna. Donec malesuada fermentum dictum. Pellentesque aliquet tortor vitae suscipit tempus. Phasellus ornare risus mi, vel placerat justo efficitur eu. Phasellus quis tincidunt enim.

Etiam a elementum lorem. Integer ac lectus hendrerit, venenatis ante vitae, accumsan eros. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur non varius dolor. Donec suscipit metus vitae ultrices sagittis. Praesent ac dapibus justo, vel mollis sapien. Pellentesque at iaculis neque. Donec ac tellus orci. Pellentesque eleifend fringilla massa, in lobortis nibh finibus in. Morbi ut neque vitae est congue tempor efficitur imperdiet dolor. Pellentesque nec nibh nec nibh fringilla laoreet. Duis fermentum ornare sollicitudin. Maecenas finibus tincidunt justo commodo commodo. Suspendisse venenatis odio dignissim auctor elementum. Duis non mauris orci.

ANNEX B

Planificació
