

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 meua 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, notifikacions 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 meua 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

<b>1</b>	<b>Introducció</b>	<b>1</b>
1.1	Motivació . . . . .	1
1.2	Finalitat i objectius del projecte . . . . .	2
<b>2</b>	<b>Estudi de viabilitat</b>	<b>3</b>
2.1	Introducció . . . . .	3
2.2	Viabilitat tècnica . . . . .	3
2.3	Viabilitat econòmica . . . . .	4
2.4	Viabilitat de mercat . . . . .	6
<b>3</b>	<b>Metodologia</b>	<b>7</b>
3.1	Introducció . . . . .	7
3.2	Scrum . . . . .	7
3.3	Adaptació del scrum . . . . .	9
<b>4</b>	<b>Planificació</b>	<b>11</b>
4.1	Introducció . . . . .	11
4.2	Desenvolupament . . . . .	11
<b>5</b>	<b>Marc de treball i conceptes previs</b>	<b>13</b>
5.1	Introducció . . . . .	13
5.1.1	Propòsit . . . . .	13
5.1.2	Conceptes clau . . . . .	13
5.1.3	Codi QR . . . . .	15
<b>6</b>	<b>Requisits del sistema</b>	<b>17</b>
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 . . . . .	20
<b>7</b>	<b>Estudi i decisions</b>	<b>23</b>
7.1	Introducció . . . . .	23
7.2	Maquinari . . . . .	23
7.3	Front-end . . . . .	23
7.3.1	Tecnologies més conegudes . . . . .	24
7.3.2	Tecnologia Escollida: Flutter . . . . .	25
7.3.3	IDE Escollida: Android Studio . . . . .	26

7.4	Back-end	27
7.4.1	Tecnologies més conegudes	27
7.4.2	Tecnologia Escollida: GraphQL	29
7.4.3	Codi obert per implementar servidors GraphQL: Apollo Server	29
7.4.4	Llenguatge: TS	30
7.5	Base de dades	31
7.5.1	Opcions	31
<b>8</b>	<b>Anàlisi i disseny del sistema</b>	<b>33</b>
8.1	Diagrama de casos d'ús	33
8.2	Diagrama de la base de dades	36
8.2.1	User	36
8.2.2	Account	38
8.2.3	Dictionary	39
8.2.4	Transaction	40
8.3	Interfícies d'usuari	40
8.3.1	Client	41
8.3.2	Administrador	41
8.4	Diagrama de processos	45
8.4.1	Descripció del procés	46
<b>9</b>	<b>Implementació i proves</b>	<b>47</b>
9.1	Introducció	47
9.2	Organitza la idea (potser millor "Idea inicial")	47
9.3	Desenvolupament d'aplicació	48
9.4	Proves	49
<b>10</b>	<b>Implantació i resultats</b>	<b>51</b>
10.1	Implantació	51
10.2	Resultats	51
<b>11</b>	<b>Conclusions</b>	<b>53</b>
<b>12</b>	<b>Treball futur</b>	<b>55</b>
12.1	Funcionalitats de l'aplicació	55
12.2	Seguretat de dades	55
12.3	Millora del codi	56
<b>13</b>	<b>Bibliografiaaaaa</b>	<b>57</b>
<b>A</b>	<b>Pressupost</b>	<b>59</b>





# Índex de figures

2.1	El preu del servei MongoDB Atlas. . . . .	5
2.2	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 . . . . .	21
8.1	Diagrama de casos d'ús de context de client. . . . .	33
8.2	Diagrama de casos d'ús de context d'administrador. "Llitar" hauria de ser "Llistar", i "Cambiar" hauria de ser "Canviar". . . . .	35
8.3	Diagrama de casos d'ús de context del sistema. "bancàries" hauria de ser "bancaris". . . . .	36
8.4	Diagrama de la base de dades. . . . .	37
8.5	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". . . . .	39
8.6	Registrar nou usuari. . . . .	41
8.7	Pàgina de <i>login</i> - Client. . . . .	42
8.8	Pàgina per generar un codi QR de pagament. . . . .	42
8.9	Pàgina per generar un codi QR de cobrament. . . . .	43
8.10	Pàgina per escanejar un codi QR. . . . .	43
8.11	Pàgina de <i>login</i> - Administrador . . . . .	44
8.12	Diagrama de processos. . . . .	45



# Índex de taules

2.1	Cost estimat del desenvolupament de l'aplicació . . . . .	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.

# 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	40 h	600 €
Disseny de l'Aplicació	60 h	900 €
Desenvolupament del Front End	120 h	1.800 €
Desenvolupament del Back End	150 h	2.250 €
Proves i Depuració	50 h	750 €
Implementació i Lliurament	30 h	450 €
<b>Total</b>	<b>450 h</b>	<b>6.750 €</b>

Taula 2.1: Cost estimat del desenvolupament de l'aplicació

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 6.750 € 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

Tal com es detallarà al Capítol 7, aquest projecte utilitzarà MongoDB Atlas per gestionar el servei de bases de dades, proporcionant una solució escalable i segura per a l'emmagatzematge i la gestió de dades. A la següent figura 2.1 es detalla el cost del servei MongoDB.

En algun lloc del text hauries de citar la figura, com per exemple “tal com es veu a la Figura 2.1, ...”. 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...”).

Free M0			Starter			Dedicated M10+ <small>RECOMMENDED</small>		
\$0 /hour <small>Free forever</small>			\$0.007 /hour <small>Up to \$26/month</small>			\$0.08 /hour <small>Pay as you go</small>		
For learning and exploring MongoDB in a cloud environment.			For application development and testing; resources and costs scale to your needs.			For production applications with sophisticated workload requirements.		
STORAGE	RAM	vCPU	STORAGE	RAM	vCPU	STORAGE	RAM	vCPU
612 MB	Shared	Shared	Up to 5GB	Shared	Shared	10 GB	2 GB	2vCPUs
Try Free			Get Started			Get Started		
<a href="#">View shared tier pricing &gt;</a>			<a href="#">View Starter pricing &gt;</a>			<a href="#">View dedicated pricing &gt;</a>		

Figura 2.1: El preu del servei MongoDB Atlas.

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 6.750 € per al desenvolupament de l'aplicació i 5.000 € per al manteniment anual, sumant un total de 11.750 €.

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 11.750 € invertits es podria aconseguir en aproximadament 0,6 anys, equivalent a uns 7 mesos. 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 2.2; 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 2.2, on es mostra un codi QR de PayPal”.corregit

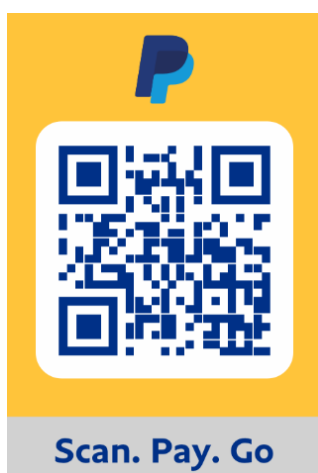


Figura 2.2: Un codi QR de PayPal.

La imatge mostrada a la Figura 2.2 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. Aquest procés de pagament és extremadament còmode i simplificat per a l'usuari.

## 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:** Les metodologies àgils 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:** Amb àgil, es treballa en iteracions curtes i incrementals, proporcionant resultats parcials i millorant contínuament el producte. 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, cadascun 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, prioritzant les més crítiques i assegurant que estan alineades amb els objectius generals del projecte.

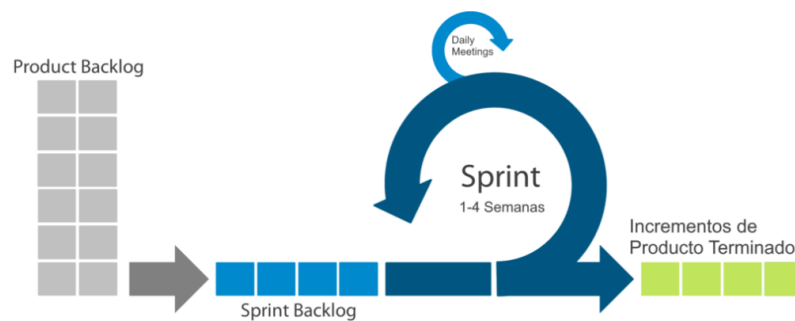


Figura 3.1: Sistema de treball Scrum.

Citar figura al text.corregit

Tal com es veu en 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 completés 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 prioritzades 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

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 i millores basades en *feedback* i nous requisits.

El Diagrama de 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.

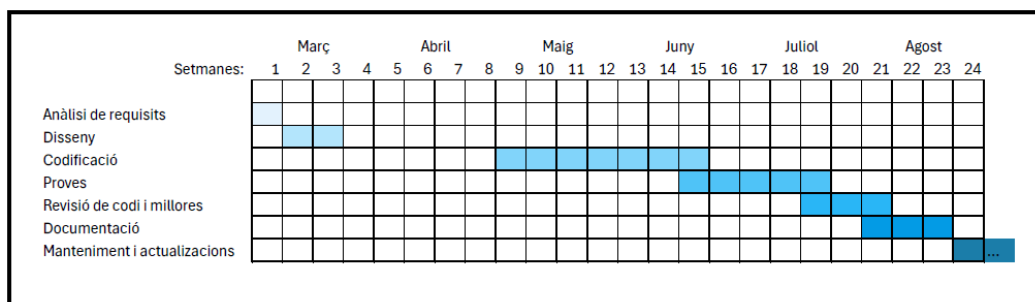


Figura 4.1: Diagrama de Gantt

Jo transformaria aquesta taula en un diagrama de Gantt, que és molt més visual (la columna "Resultats" no cal). Abans de posar el diagrama, introdueix-lo amb una mica de text.[corregit](#)

També faria que tot plegat fos una secció, i no una subsecció.[corregit](#)

# 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. En resum, 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 desxifratge 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/DesactivarUsuari 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 amb el compte desactivat. Altres operacions seguiran igual.
- **Dictionary:** Cada cop que l'usuari genera un codi QR, tota la informació es guarda en un Dictionary. Quan el codi és escanejat, el sistema consulta internament aquest Dictionary 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.

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 per exemple 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'entrar dades manualment, reduint errors i accelerant el procés de transacció.





# 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 especifiquen 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??

D'altra banda, el Bizum està vinculat a un número de telèfon, mentre que l'aplicació que es presenta es vincularà a un número de compte bancari. Cada compte bancari serà capaç de generar codis QR per a cobraments o, a la inversa, serà capaç de llegir codis QR per a efectuar pagaments.

**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").

**Operacions d'usuari administrador:**

- **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 financera 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 cibernètics.

### 6.1.3 Matriu de dependències de requisits

A continuació presenta una 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					x											
RF- 18		x					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 o pàgina web. 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 recompilació 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 continuï 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 interfície 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.**

### 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 utilitza 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 minimalista 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:** Amplement utilitzat en combinació amb MongoDB i Angular/React per a desenvolupaments full-stack (conegut 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.

La creació d'APIs (Interfícies de Programació d'Aplicacions) és fonamental per permetre la comunicació entre el servidor i els clients.

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

### 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.

Un altre avantatge important de GraphQL és la seva capacitat per evolucionar l'API de manera flexible. Amb GraphQL, es poden afegir nous camps i tipus de dades a l'API sense afectar les consultes existents. Això facilita l'evolució contínua de l'API sense interrupcions per als clients actuals, permetent un desplegament més suau de noves funcionalitats.

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 que permet construir un servidor GraphQL de manera senzilla i eficaç. S'integra fàcilment amb diverses tecnologies i entorns, i és compatible amb els estàndards i especificacions de GraphQL. Es dissenya

per funcionar amb qualsevol servidor HTTP i és flexible en la configuració i integració.

Apollo Server és una de les solucions més destacades per a la creació de servidors GraphQL. Els següents punts ressalten els avantatges més importants:

- **Ecosistema Integrat:** Apollo Server forma part d'un ecosistema complet que inclou **Apollo Client**, **Apollo Studio**, i altres eines. Aquesta integració facilita la creació d'una solució completa de GraphQL, amb una gestió unificada tant del client com del servidor.
- **Suport per a Subscripcions:** Ofereix suport per a **subscriptions**, permetent actualitzacions en temps real i la sincronització d'informació entre el servidor i el client. Aquesta funcionalitat és ideal per a aplicacions que requereixen dades en temps real, com aplicacions de xat o eines de col·laboració.
- **Monitorització i Anàlisi:** A través d'**Apollo Studio**, proporciona eines avançades per a **monitorització i anàlisi**. Aquestes eines permeten rastrejar el rendiment de les consultes i identificar problemes de rendiment, facilitant l'optimització i la millora contínua de l'aplicació.

#### 7.4.4 Llenguatge: 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

En el projecte, la base de dades s'ha implementat utilitzant MongoDB Atlas, un servei de base de dades en el núvol gestionat per MongoDB. Atlas ofereix una solució NoSQL basada en el núvol que destaca per la seva escalabilitat, flexibilitat i facilitat de gestió. A diferència de les bases de dades relacionals tradicionals, MongoDB adopta un model de dades orientat a documents, emmagatzemant la informació en documents BSON (Binary JSON). Aquest enfocament proporciona una gran flexibilitat en l'estructura de les dades, adaptant-se millor a les necessitats canviants dels projectes.

MongoDB Atlas també integra característiques avançades de seguretat, com el xifrat de dades tant en repòs com en trànsit, controls d'accés detallats i monitorització en temps real. Aquestes funcions asseguren que les dades estiguin protegides contra accessos no autoritzats i permeten als administradors vigilar l'estat de la base de dades de manera eficaç.

A més, Atlas proporciona eines completes per a la còpia de seguretat i la recuperació de dades, incloent la creació automàtica de còpies de seguretat i la recuperació en cas de fallades. També ofereix integració amb diversos serveis de dades i aplicacions, així com eines avançades de consulta i agregació que faciliten la manipulació i l'anàlisi de dades.

### 7.5.1 Opcions





# Anàlisi i disseny del sistema

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

## 8.1 Diagrama de casos d'ús

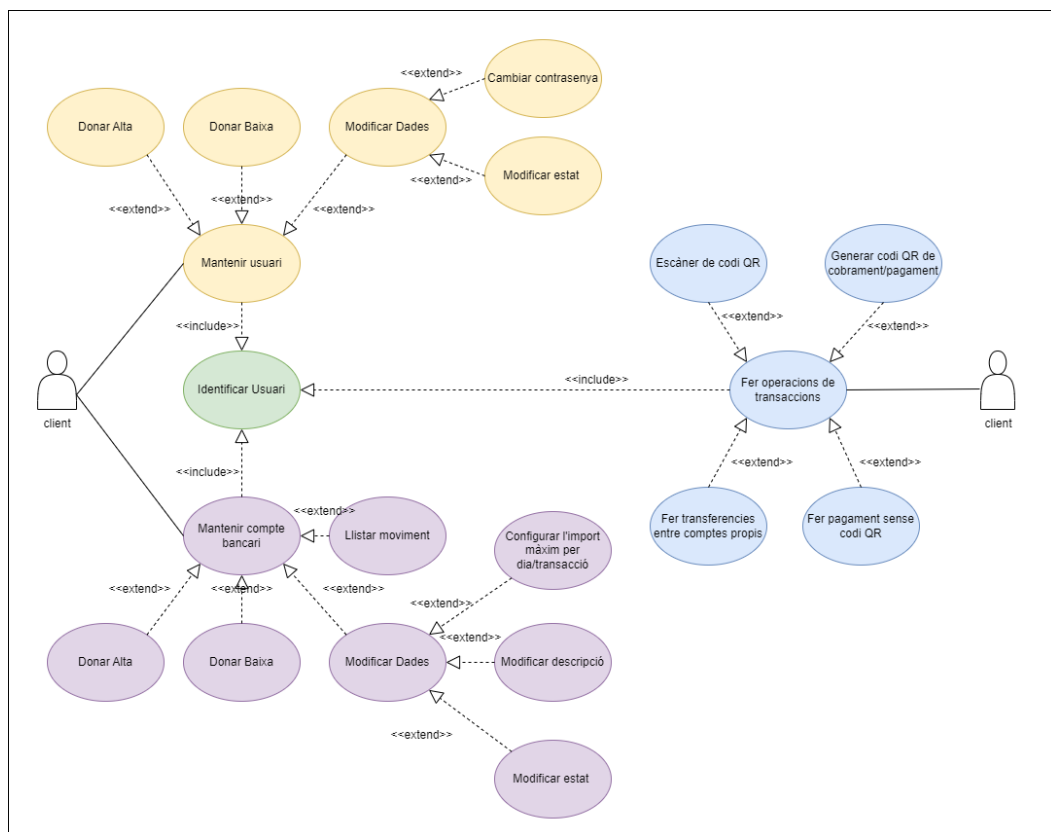


Figura 8.1: Diagrama de casos d'ús de context de client. “Canviar” hauria de ser “Canviar”, i “transferències” hauria de ser “transferències”. Si la llegenda és massa llarga i no vols que a l'índex de figures del principi t'hi quedi una frase tan llarga, pots utilitzar els [], tal com t'ho he posat en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

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

Aquest diagrama de casos d'ús 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). Cadascuna d'aquestes instàncies està connectada a diversos casos d'ús.

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 financera, ja que permeten al client executar transaccions de manera ràpida i segura.

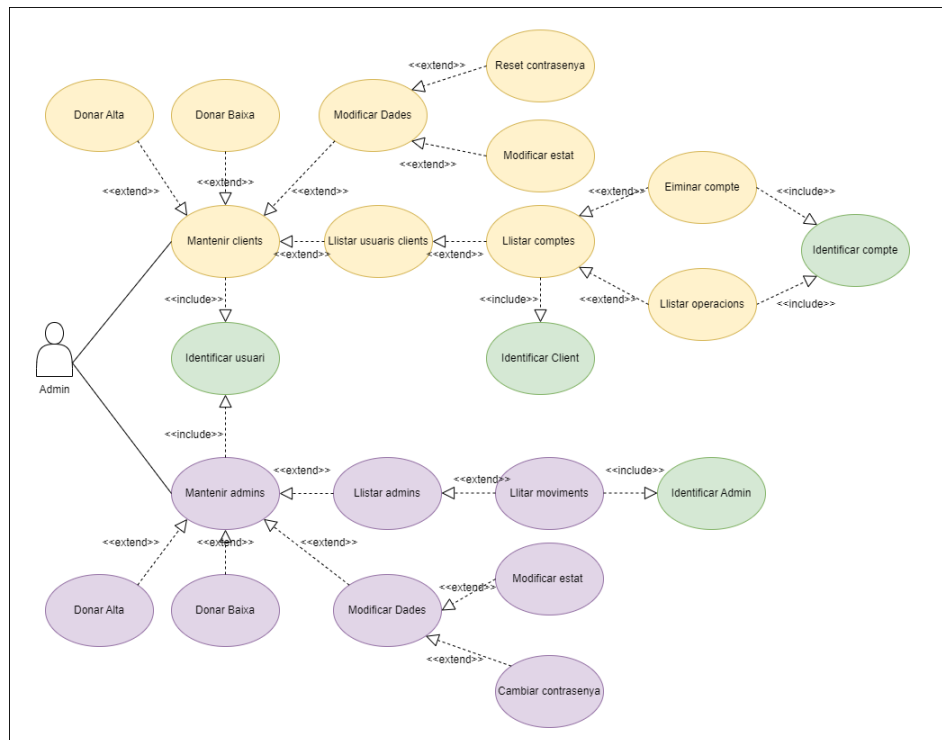


Figura 8.2: Diagrama de casos d'ús de context d'administrador. “Llitar” hauria de ser “Llistar”, i “Cambiar” hauria de ser “Canviar”.

Cal citar bé la figura.

Aquest segon diagrama de cas d'ús 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. 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 ope-

racions i la necessitat d'autenticació o identificació per dur-les a terme.

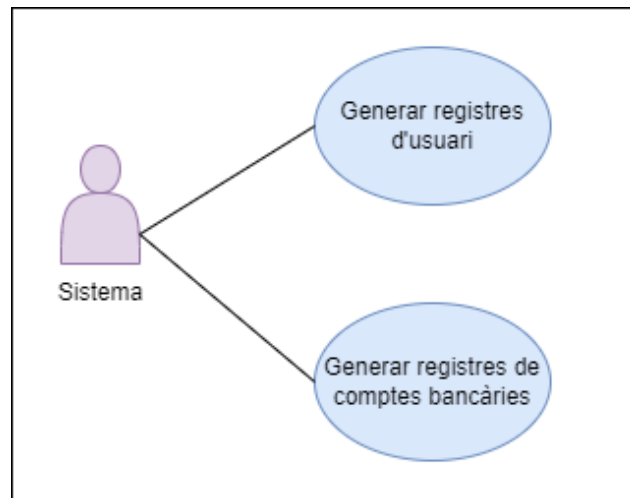


Figura 8.3: Diagrama de casos d'ús de context del sistema. “bancàries” hauria de ser “bancaris”.

Cal citar bé la figura.

Adicionalment, el sistema disposa de la funció de generar *logs* per a cada operació. Això significa 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 auditories o revisió posterior, garantint la traçabilitat i la seguretat del sistema.

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

## 8.2 Diagrama de la base de dades

Cal citar bé la figura.

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.

### 8.2.1 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,

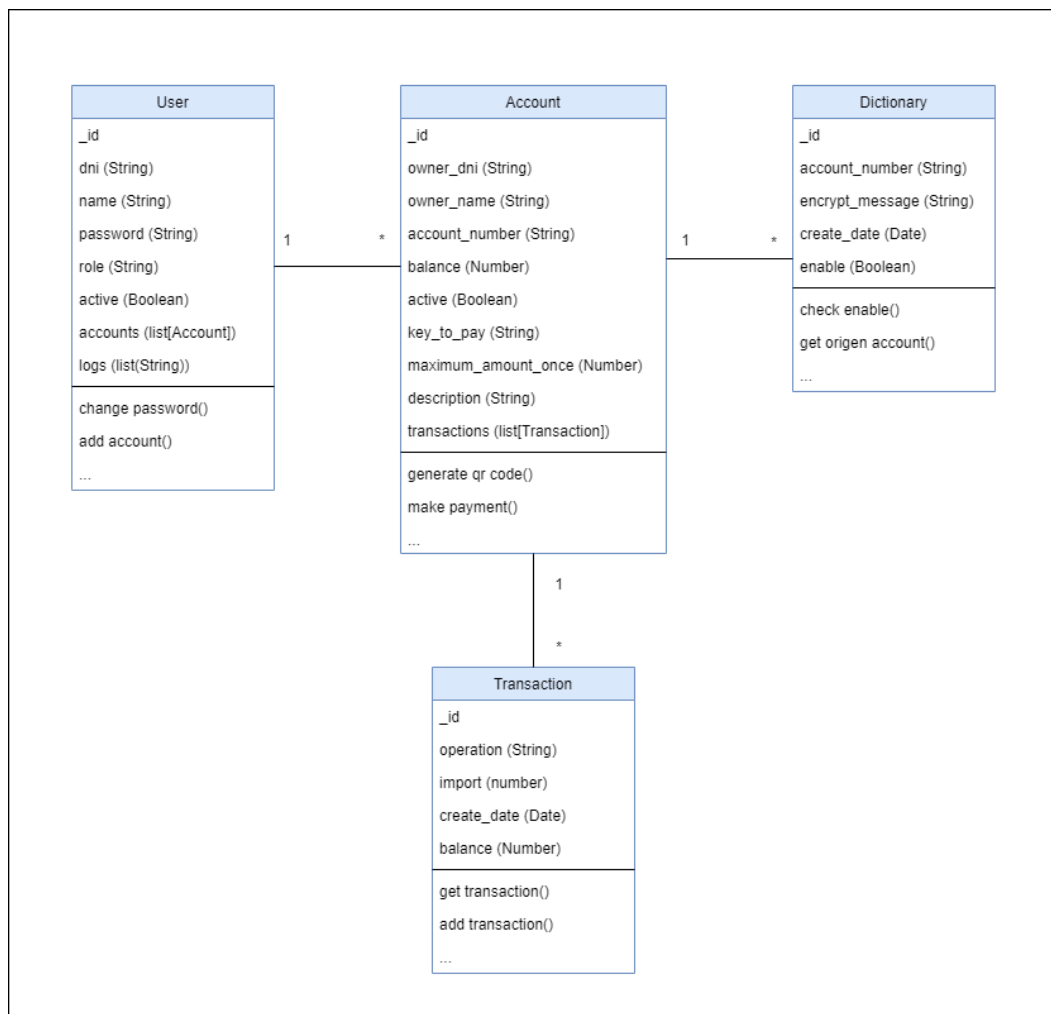


Figura 8.4: Diagrama de la base de dades.

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'encrpta 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.

### 8.2.2 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.

### 8.2.3 Dictionary

L'entitat **Dictionary** 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.

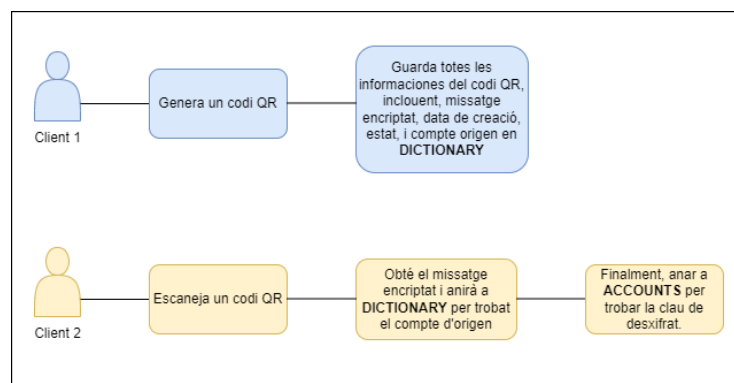


Figura 8.5: Ús de Dictionary. **Canviar “informaciones” per “informacions”, “inclouent, missatge” per “incloent missatge”, “en DICTIONARY” per “a la taula Dictionary”, “a DICTIONARY per trobat” per “a la taula Dictionary per trobar”, i “a ACCOUNTS” per “a la taula Account”.**

**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.

#### 8.2.4 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.

### 8.3 Interfícies d'usuari

El disseny de les interfícies d'usuari ha estat una prioritat abans d'endinsar-nos 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 descrivint.

### 8.3.1 Client

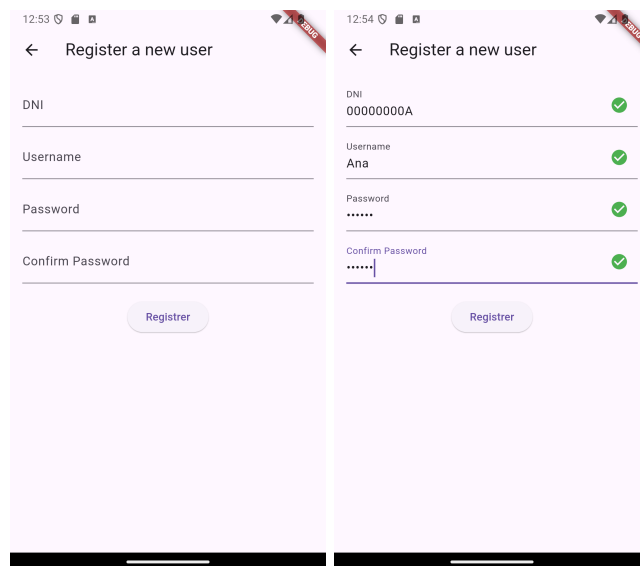


Figura 8.6: Registrar nou usuari.

### 8.3.2 Administrador

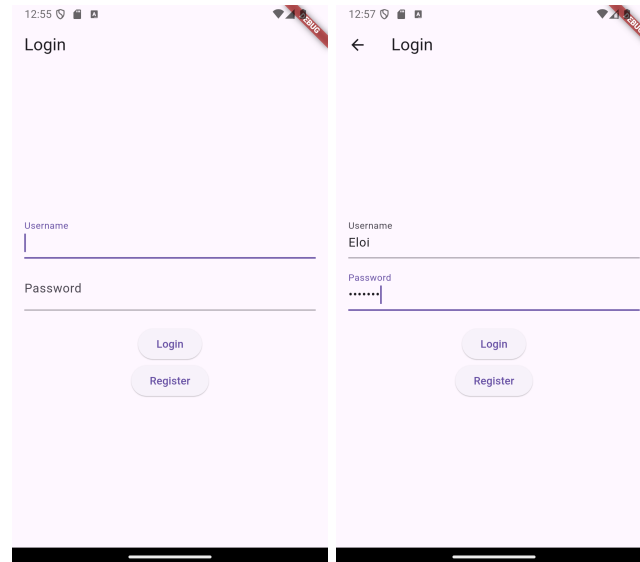
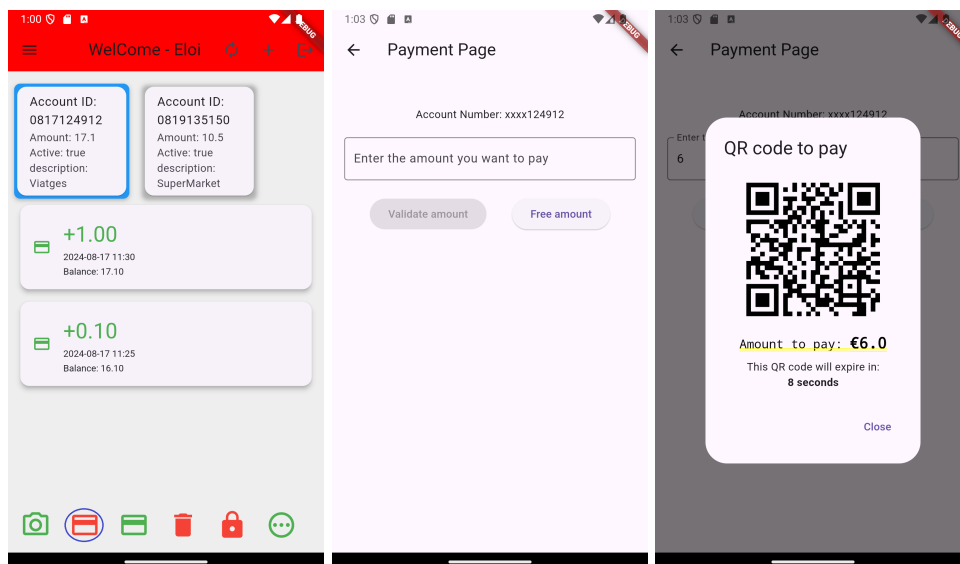
Figura 8.7: Pàgina de *login* - Client.

Figura 8.8: Pàgina per generar un codi QR de pagament.

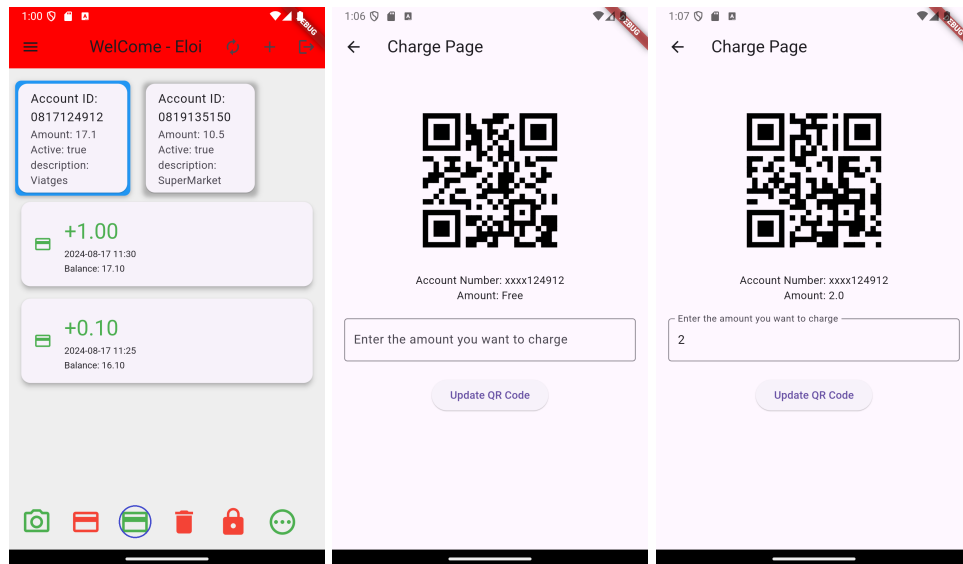


Figura 8.9: Pàgina per generar un codi QR de cobrament.

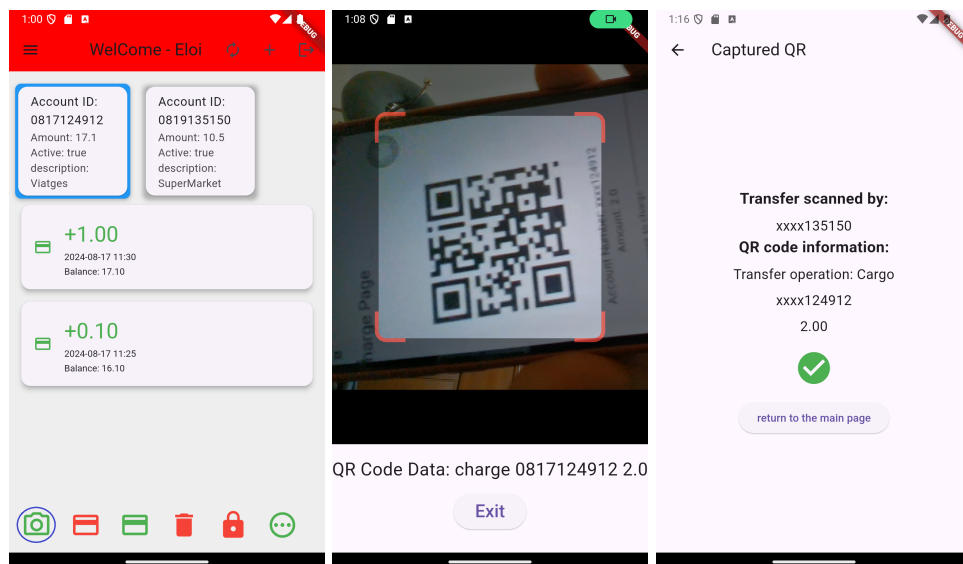
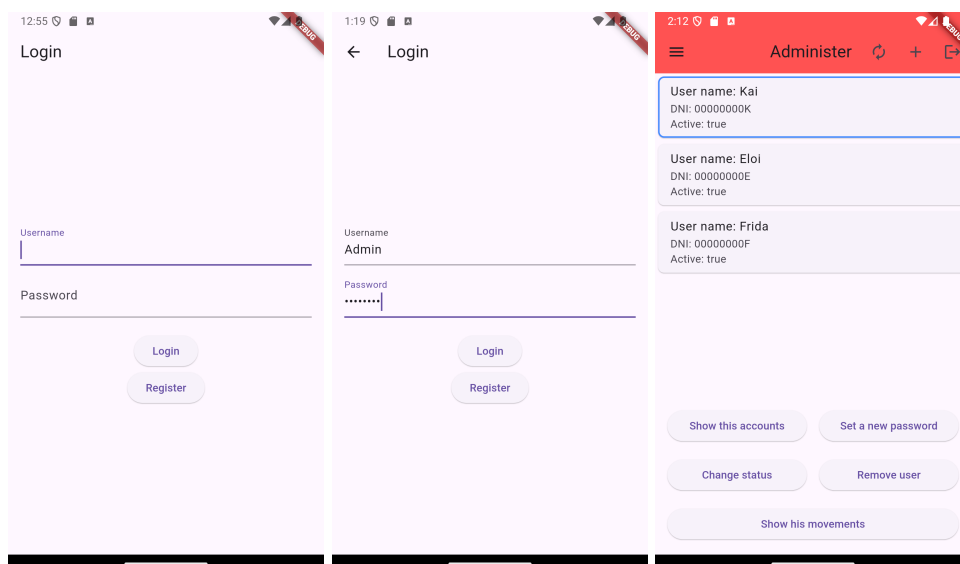


Figura 8.10: Pàgina per escanejar un codi QR.

Figura 8.11: Pàgina de *login* - Administrador

## 8.4 Diagrama de processos

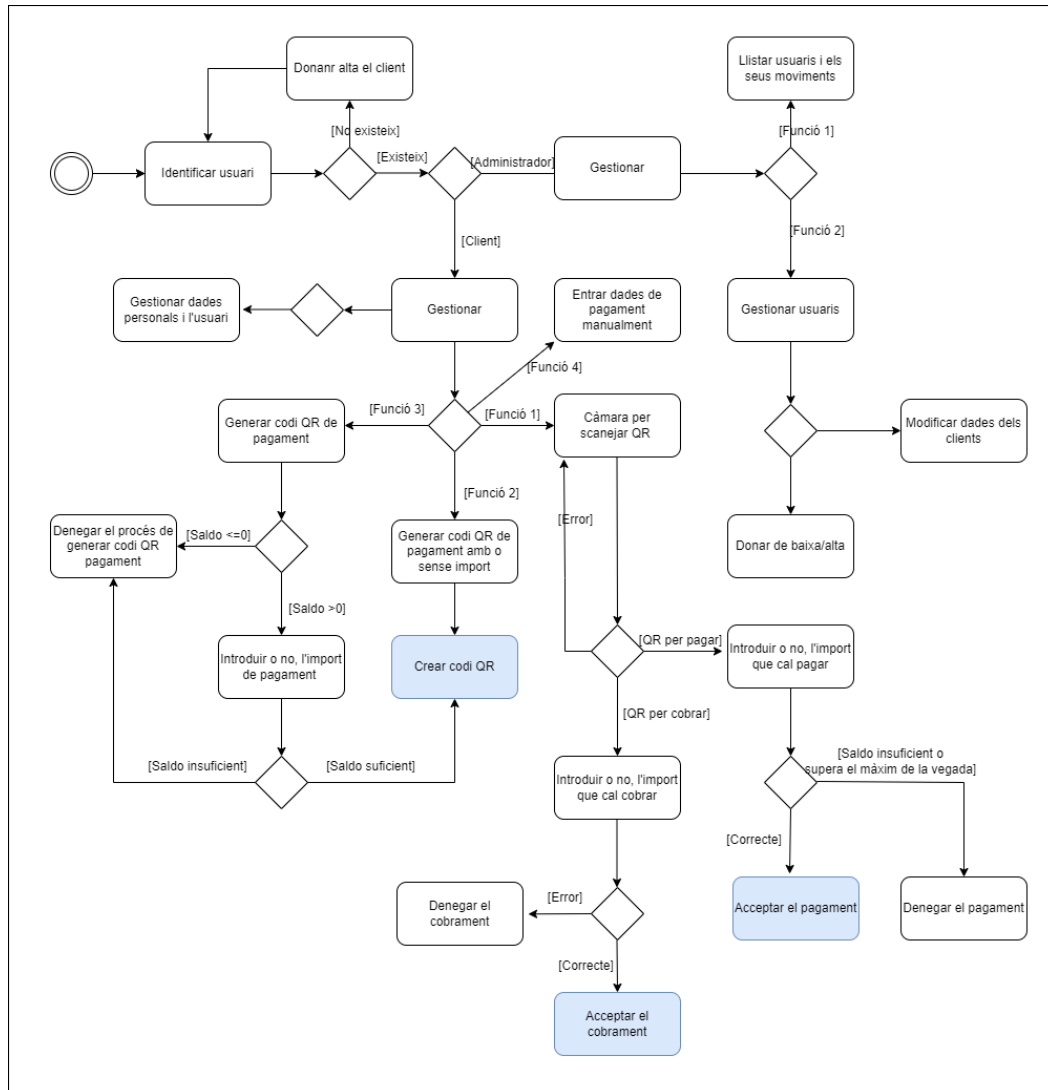


Figura 8.12: Diagrama de processos.

**Cal citar bé la figura.**

Aquesta diagrama de flux 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.

### 8.4.1 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 res (**no cal fixar res, o no cal fixar-se en res?**), 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.

# 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, al desenvolupament actual de la mateixa. També ho serà incloure i explicació de com l'aplicació s'ha provat constantment (potser millor “També s'explicaran les diferents proves que s'han dut a terme per verificar el funcionament de l'aplicació”).

## 9.2 Organitza la idea (potser millor “Idea inicial”)

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'oferir, 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 sòlida. 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 Desenvolupament d'aplicació

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.

Vaig començar a desenvolupar l'aplicació API (vols dir l'API de l'aplicació?), 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 aplicacions d'interfície d'usuari (potser millor “vaig començar a treballar en les interfícies d'usuari”). Vaig desenvolupar primer l'aplicació bàsica (aplicació?), 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 abstracció i reutilitzar el codi de manera eficaç per simplificar el desenvolupament.



Si en el futur es necessita afegir una altra aplicació (**aplicació?**), el procés serà més senzill, ja que molts components i funcions podrien ser reutilitzats.

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

## 9.4 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:

- **Proves d'unitat:** Aquestes proves se centren en verificar el correcte funcionament de les parts més petites del codi, com funcions o mètodes, de manera independent.
- **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.
- **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.
- **Proves de sistema:** Aquestes proves verifiquen que tot el sistema compleixi amb els requisits establerts, utilitzant escenaris d'ús que reflecteixen situacions reals.
- **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.
- **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 funcioni 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 trobava un error en les proves, actualitzava les proves i després corregia el codi real. Si no es trobava 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.

# 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 la 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.

El projecte té dues vies potencials per a la seva implantació futura. La primera opció és integrar l'aplicació amb els sistemes bancaris estatals, similar a com funciona Bizum. Aquesta integració permetria que l'aplicació ofereixi funcionalitats similars a les d'un servei de pagament a través de bancs. La segona opció és establir una passarel·la de pagament pròpia, creant una solució similar a PayPal que permeti operar com una empresa de pagaments independent. Aquesta opció ofereix una major autonomia i flexibilitat en la gestió de les transaccions.

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).

## 10.2 Resultats

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 valuosa 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. A continuació, es realitzaran proves exhaustives per garantir el bon funcionament del sistema abans del llançament oficial de l'aplicació.

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).

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

## CAPÍTOL 11

# Conclusions

---

L'objectiu d'aquest projecte és permetre que les persones puguin realitzar operacions bancàries amb la màxima flexibilitat i seguretat possible. La idea va sorgir després de la meua experiència a la Xina el 2019, on vaig observar que la majoria de les transaccions es realitzaven a través de l'aplicació de missatgeria "WeChat". Aquesta aplicació integra una passarel·la de pagament molt còmoda, que facilita les transaccions i ofereix una major transparència en les operacions. A més, l'ús de pagaments electrònics contribueix a reduir els robatoris associats amb l'ús de diners en efectiu.

Recentment, vaig visitar una sala de jocs amb amics on vaig notar un canvi significatiu en el sistema de pagament. Abans, només es podien utilitzar monedes d'1 € per activar les màquines, però ara s'ha afegit un mini TPV (Terminal Punt de Venda) per a cada màquina. Aquesta actualització requereix una inversió en infraestructura per als TPV. Només cal imaginar què passaria si aquest procés es pogués simplificar utilitzant un codi QR: cada màquina podria estar configurada amb l'import a cobrar, mentre que els clients podrien pagar fàcilment escanejant el codi QR a través del seu dispositiu mòbil. Aquesta solució ofereix una alternativa més flexible i eficient, tant per a les empreses com per als usuaris.

El sistema de pagament amb codi QR ofereix una sèrie d'avantatges clau. En primer lloc, simplifica el procés de pagament, ja que els clients poden realitzar transaccions simplement escanejant un codi QR amb el seu dispositiu mòbil. 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'encryptació de dades i l'autenticació de dos factors, per protegir les dades dels usuaris i prevenir frau.

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 fluida i assegurar la compatibilitat amb diversos dispositius i aplicacions.

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. Tot i que plantegen nous reptes, especialment en termes de seguretat, la continuïtat en la innovació tecnològica pot ajudar a superar aquests obstacles i millorar l'experiència de pagament per a usuaris i negocis per igual.

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.

## CAPÍTOL 12

# Treball futur

---

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

### 12.1 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 fluida i moderna.

### 12.2 Seguretat de dades

Un aspecte clau a destacar és el sistema d'encryptació. Actualment he utilitzat l'encryptació AES (Advanced Encryption Standard), que és un algorisme de xifrat simètric àmpliament utilitzat per la seva eficàcia i seguretat. No obstant això, considero que seria molt més avantatjós implementar l'encryptació RSA (Rivest-Shamir-Adleman), que és un sistema d'encryptació asimètric. L'encryptació 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.

## 12.3 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

## Bibliografiaaaaa

---



## Pressupost

---

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.

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

---