



UNIVERSITAT<sup>DE</sup>  
BARCELONA

**Treball de Fi de Grau**

**GRAU D'ENGINYERIA INFORMÀTICA**

**Facultat de Matemàtiques i Informàtica**

**Universitat de Barcelona**

---

**GESTIÓ DEL CICLE DE VIDA D'UN PROJECTE  
D'APIS PER AL SECTOR BANCARI**

---

**Ivet Aymerich Campillo**

Director: Oliver Díaz Montesdeoca

Realitzat a: Departament de

Matemàtiques i Informàtica

Barcelona, Juny 2020

## **RESUM**

En aquest document es mostra el seguiment a realitzar en un projecte d'APIs dintre d'una empresa per a un client bancari. Es documenta la participació en el projecte des de l'anàlisi, presa de requisits i el disseny tècnic i funcional, fins el desenvolupament del software, les proves unitàries a realitzar sobre el projecte i el desplegament del serveis.

En concret s'estudien les targetes de prepagament, amb l'anàlisi i desenvolupament de 16 de les APIs pactades amb el client.

## **RESUMEN**

En este documento se estudia el seguimiento a realizar en un proyecto de API dentro de una empresa para un cliente bancario. Se documenta la participación en el proyecto desde el análisis, la toma de requerimientos y el diseño técnico y funcional, hasta el desarrollo del software, las pruebas unitarias a realizar sobre el proyecto y el despliegue de los servicios.

En concreto se estudian las tarjetas de prepago, con el análisis y el desarrollo de 16 de las API pactadas con el cliente.

## **ABSTRACT**

This document studies the follow-up when developing an application program interface (API) for a bank client. The participation in the project is documented , from the analysis, the requirements gathering and the technical design and the functional design, to the development of the software, the unitary tests, and the deployment of the services.

Specifically, prepaid cards are studied, with the analysis and the development of 16 API's negotiated with the customer.

## ÍNDEX

1. Introducció .....	6
1.1. Contextualització del problema .....	6
1.2. Estat del problema a l'actualitat .....	7
1.3. Estat del desenvolupament del projecte .....	7
1.4. Projecte a desenvolupar .....	8
2. Objectius.....	10
3. Planificació.....	11
3.1. Planificació projecte empresa.....	11
3.2. Costos .....	15
4. Anàlisi i disseny.....	16
4.1. Anàlisi i presa de requisits.....	16
4.2. Disseny tècnic.....	18
4.2.1. Distribució dels serveis .....	18
4.2.2. API RESTful .....	19
4.2.3. APIs Serveis .....	20
4.2.3.1. Exemple API.....	21
4.3. Disseny funcional .....	23
5. Desenvolupament.....	25
5.1. Planificació de l'equip.....	25
5.2. Parts a desenvolupar .....	26
5.2.1. Frontals .....	27
5.2.2. Sistemes per externalitzar .....	27
5.2.3. Sistemes d'integració.....	28
5.2.3.1. Exemple flux sistema d'integració .....	28
5.2.3.2. Connexió amb serveis externs.....	32
5.2.4. Backend.....	33
5.3. Metodologia de desenvolupament .....	33

6. Proves, avaluació i resultat.....	35
6.1. Plans de proves .....	35
6.1.1. Exemple pla de proves API.....	35
6.2. Proves test .....	36
6.2.1. Exemple proves realitzades API.....	36
6.3. Proves preproducció.....	38
6.4. Desplegament .....	39
6.5. Avaluació.....	39
7. Conclusions i treball futur .....	40
8. Referències .....	41

## Índex de Taules

Taula 1. Planificació tesi inicial .....	11
Taula 2. Planificació tesi final .....	11
Taula 3. Planificació inicial D.13.....	12
Taula 4. Planificació inicial D.20.02.....	12
Taula 5. Planificació inicial D.21.....	13
Taula 6. Planificació inicial D.21.08.....	13
Taula 7. Planificació inicial E.23 .....	13
Taula 8. Planificació final D.13.....	13
Taula 9. Planificació final D.20.02 .....	14
Taula 10. Planificació final D.21 .....	14
Taula 11. Planificació final D.21.08.....	14
Taula 12. Planificació final E.23 .....	14
Taula 13. Relació dels serveis .....	19
Taula 14. Rutes dels serveis API.....	20
Taula 15. Headers comuns a tots els serveis.....	21
Taula 16. Dades entrada API 14.....	21
Taula 17. Dades sortida API 14 .....	22
Taula 18. Codis d'error API 14 .....	22

## Índex de figures

Figura 1. Disseny funcional API 14 .....	23
Figura 2. Esquema estructura projecte .....	26
Figura 3. Esquema crides sistema integració .....	28
Figura 4. Flux mètode getStatementSE sistema integració API 14.....	29
Figura 5. Flux mètode getStatementCP sistema integració API 14 .....	30
Figura 6. Esquema crides backend.....	33
Figura 7. Dades invocació prova execució correcte .....	37
Figura 8. Dades sortida prova execució correcte .....	37
Figura 9. Dades invocació prova execució error .....	38
Figura 10. Dades sortida prova execució error .....	38

## **1. Introducció**

### **1.1. Contextualització del problema**

Vivim en un món en el qual les persones necessitem disposar de diners en qualsevol moment, amb la necessitat de poder-lo guardar i utilitzar de la manera més segura possible. És en aquest fet que els bancs han vist l'oportunitat de seguir innovant-se i crear nous productes per a arribar a la gent, i així poder créixer econòmicament. Estem en un moment on es necessita innovar, mostrar nous productes a la població, tot per no quedar endarrerit amb respecte a la competència. Una competició en la qual la companyia que no innova, creant nous productes i pensant amb la necessitat de la gent en cada moment, pot quedar simplement en el record per el que havia estat i no per el que segueix sent.

Amb el desenvolupament del comerç electrònic (e-commerce) es creen noves necessitats a l'hora d'adquirir productes o serveis per internet. És per això que es crea una nova inquietud als usuaris de la banca, que deixen les dades de la targeta bancària a una pàgina web completament desconeguda amb la incertesa de si el receptor d'aquestes pot fer-ne un ús maliciós. Així doncs, es comença a desenvolupar un nou mètode en el qual el client es senti més segur a l'hora de fer aquestes transaccions, sorgint així la creació de targetes de prepagament, que representen targetes amb els mateixos beneficis que poden tenir les targetes de dèbit o de crèdit però sense tenir la necessitat d'estar vinculades en un compte bancari. D'aquesta manera, un usuari pot carregar els diners que necessiti a la targeta i realitzar de forma segura moviments tant per compres online com per compres en comerços tradicionals, inclòs també per poder anar de viatge i per altres beneficis que s'hi poden trobar.

Aquesta seguretat majoritàriament s'obté al tenir una quantitat baixa de diners guardats a la targeta. Es carrega només quan sigui necessari, com pot ser al realitzar compres online o abans de marxar de viatge. En el cas d'ús fraudulent, pèrdua o robatori tampoc es podrà perdre una gran quantitat ja que és l'usuari qui controla el límit màxim de diners disponible a la targeta.

Aquestes targetes de prepagament són més segures que les tradicionals degut a aquesta autonomia del control dels diners. També són útils per a controlar les despeses, tant per a un mateix com per a un potencial beneficiari (com podria ser el cas de la targeta per a un fill al qual se li volen assignar uns diners de manera controlada), així com també per a fer regals amb targetes d'una sola càrrega.

## **1.2. Estat del problema a l'actualitat**

Les targetes de prepagament arriben amb l'augment de les compres per internet i amb l'objectiu de donar una seguretat a les compres online. Des de llavors ja han sortit altres mètodes de seguretat que han fet perillar el seu consum. Tot i la desconfiança dels clients als nous mètodes, aquest tipus de targetes es segueixen considerant de molta utilitat.

Aquest seguit de targetes ja porten temps en el mercat. Així doncs, un es pot preguntar quina és la funció a desenvolupar en aquest projecte. Doncs bé, tal i com s'ha dit prèviament, el món bancari, així com molts d'altres, són una carrera de fons i cada un dels bancs vol aportar el màxim de les funcions i serveis als seus clients per evitar que marxïn a la competència. Per tant, a cada banc li interessa disposar d'aquest tipus de targetes com a servei per els seus clients.

## **1.3. Estat del desenvolupament del projecte**

En aquest projecte es treballarà en un desenvolupament internacional dels serveis informàtics associats a aquestes targetes de prepagament. Per a dur a terme aquest desenvolupament es disposa d'un model de targetes ja creades per a un client local, el qual vol exportar les targetes en un banc internacional. Aquesta internacionalització del projecte consisteix en la investigació del funcionament bancari del país destí i el desenvolupament corresponent per a poder adaptar ambdós funcionaments.

De manera que es comença amb una base prèviament creada del client local que servirà com a base de suport per el nou projecte internacional, això suposarà un treball en l'adaptació de totes les funcions.

El nou projecte d'internacionalització es troba ja en un primer estat inicial desenvolupat, que consta d'un seguit de 9 serveis:

1. Creació d'un client.
2. Modificació d'un client.
3. Eliminació d'un client.
4. Consulta d'un client.
5. Creació de documents vinculats a un client.
6. Eliminació de documents vinculats a un client.
7. Modificació de documents vinculats a un client.



8. Creació de targetes.
9. Llistat de targetes d'un client.

Comptant ja amb els primers serveis creats s'obté una visió prèvia del funcionament per els següents desenvolupaments previstos i els quals es duran a terme al llarg d'aquest treball.

#### **1.4. Projecte a desenvolupar**

Es té previst obtenir un aprenentatge més extens, no només d'una sola àrea del projecte, que sigui útil i transferible per a possibles futurs projectes d'àmbits diferents, ja que els processos a seguir a l'hora de desenvolupar altres tasques parteixen d'una estructura comuna.

En aquest projecte es participarà en la creació d'una gran variació de serveis per a unes noves targetes de prepagament. Aquesta participació abasta a totes les fases que es duen a terme al llarg del projecte, de manera que al final s'haurà obtingut un aprenentatge general de cada una d'elles.

Les fases que es duran a terme per a cada un dels serveis són:

- Anàlisi previ i presa de requisits per al disseny d'una nova API (Interfície de programació d'aplicacions) on s'especifica de manera abstracta com es tindran que desenvolupar les diferents funcions, així com les dades d'entrada i sortida necessàries de cada una d'elles. Posant un exemple d'API per a aquestes dades, si es vol demanar un servei per consultar les dades d'una targeta, aquesta especificarà les dades d'entrada per fer la petició, en aquest cas un identificador de la targeta, i les dades de sortida que retorna el servei, és a dir totes les dades que té el sistema de la targeta introduïda.
- Desenvolupament i proves unitàries per a la comprovació del seu correcte funcionament, veient així inclús possibles errors durant el procés de la presa de decisions i disseny, amb totes les possibles afectacions que això pot originar i per acabar es durà a terme el seu desplegament.

Se seguirà el projecte amb els següents 16 serveis nous i s'estudiaran cada un d'ells segons les seves necessitats.

Els serveis que es desenvoluparan en el projecte són els següents:

1. Activació de la targeta.

2. Detall de la targeta.
3. Llistat de targetes.
4. Creació del compte de la targeta.
5. Firma del compte de la targeta.
6. Pagament de rebut.
7. Llistat de comptes de la targeta.
8. Detall compte de la targeta.
9. Llistat transaccions de la targeta.
10. Llistat transaccions del compte de la targeta.
11. Detall d'un rebut.
12. Cancel·lació de pagament d'un rebut.
13. Llistat liquidacions transacció.
14. Llistat de rebuts.
15. Detall del mètode de pagament d'un rebut.
16. Cash Advance

## 2. Objectius

En aquest projecte, l'objectiu és desenvolupar un seguit de serveis informàtics per a la creació de targetes de prepagament per al sector bancari. Comptant amb un conjunt de serveis ja creats i una llista encara extensa de futurs serveis a desenvolupar, el projecte es basarà en 16 serveis, amb els quals es pretén veure tot el procés necessari per a la seva creació.

Per a la realització del projecte es vol:

- Dur a terme un anàlisi previ i una presa de requisits per augmentar l'eficiència, la productivitat i l'efectivitat dels serveis en la creació d'una nova API.
- Donar suport a l'equip de desenvolupament dels serveis, de manera que es treballarà amb diferents eines i a diferents nivells del desenvolupament.
- Desenvolupar serveis de petició (GET) i publicació (POST) de dades bancàries.
- Veure la comunicació necessària amb serveis externs de l'empresa on es treballarà, per tal de la realització del producte final.
- Fer un seguit de proves unitàries per a la comprovació del correcte funcionament dels serveis i una avaluació dels resultats obtinguts.

### 3. Planificació

Per a planificar el projecte es té en compte el temps que es necessitarà per part de l'empresa i el temps del treball individual, extern a l'empresa a desenvolupar per a la creació de la tesi.

Aquest treball es va desenvolupar gràcies a un conveni entre la Universitat de Barcelona y Capgemini en un període de 5 mesos (novembre 2019-març 2020). Es van destinar 450 hores a l'empresa per realitzar diferents tasques dins el projecte.

La creació de la memòria es va realitzar tant simultàniament al treball com posteriorment un cop tot el projecte ja ha estat desenvolupat. D'aquesta manera la planificació inicial general de la tesi es duria a terme amb els terminis indicats a la Taula 1.

Nom	Nov	Des	Gen	Feb	Mar	Abr	Mai	Jun
<b>Projecte empresa</b>	■							
<b>Redacció memòria</b>				■				

Taula 1. Planificació tesi inicial

Finalment el temps final empleat és detalla a la Taula 2. Es pot veure que el projecte d'empresa s'ha allargat un mes i mig més degut a endarreriments en la tapa d'anàlisi i documentació, així com haver de augmentar el temps previst de desenvolupament dels serveis.

Nom	Nov	Des	Gen	Feb	Mar	Abr	Mai	Jun
<b>Projecte empresa</b>	■							
<b>Redacció memòria</b>					■			

Taula 2. Planificació tesi final

#### 3.1. Planificació projecte empresa

Abans de començar amb la definició dels serveis i el seu desenvolupament, es necessita una planificació de les tasques a realitzar. Per tal de planificar es necessita tenir clar quines són les tasques .

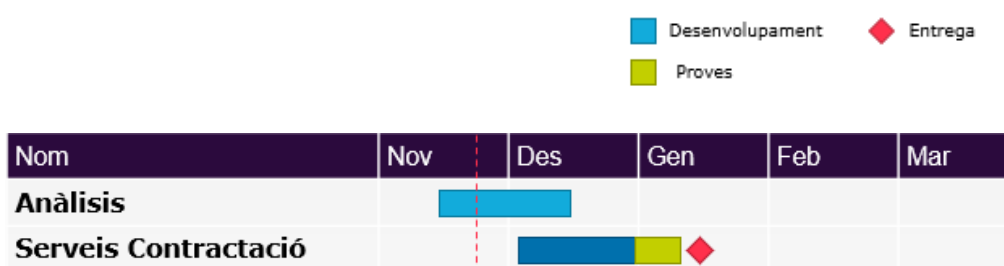
El projecte a desenvolupar per el client està dividit en diferents blocs, els blocs en els quals es troben les tasques a realitzar en aquest projecte són els següents:

- **D.13. Contractació:** Creació dels mètodes relacionats per a la creació i contractació del contracte.
- **D.20.02. Operacions:** Mètodes relacionats amb les operacions de les transaccions realitzades, com són els diferents llistats de transaccions.

- **D.21. Cicle de vida:** Desenvolupament tant dels mètodes per a l'activació de targetes com per a consultes de les dades dels comptes i de les targetes.
- **D.21.08. Crèdit:** Creació de les funcions per a les operacions de l'obtenció del mètode de pagament i per a peticions de crèdit (cash advance).
- **E.23. Rebuts:** Bloc relacionat amb el desenvolupament de pagament de rebuts. Posteriorment s'afegiran els mètodes encarregats de controlar, gestionar i obtenir rebuts.

Per cada un dels blocs s'estableix una planificació inicial dintre del projecte global del temps que implicarà cada una de les tasques a realitzar en ells:

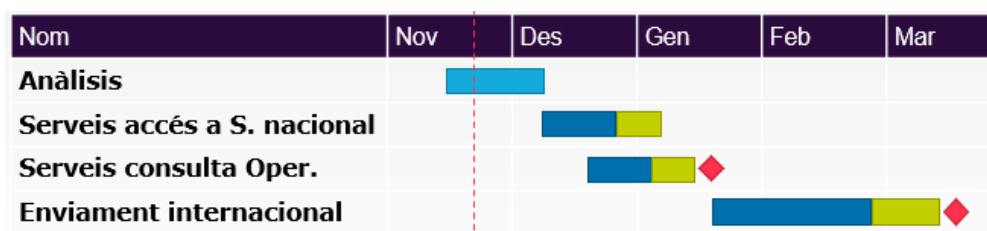
- **D.13. Contractació:**
  1. Tapa d'anàlisi i disseny dels serveis de contractació.
  2. Desenvolupament dels serveis de contractació.



Taula 3. Planificació inicial D.13

**D.20.02. Operacions:**

1. Tapa d'anàlisi i disseny dels serveis d'operació.
2. Desenvolupament dels serveis d'accés al servei nacional.
3. Desenvolupament dels serveis de consulta d'operacions.
4. Batch de l'enviament al servei internacional.



Taula 4. Planificació inicial D.20.02

- **D.21. Cicle de vida:**
  1. Tapa d'anàlisi i disseny dels serveis del cicle de vida.
  2. Desenvolupament dels serveis de consulta i activació.

Nom	Nov	Des	Gen	Feb
<b>Anàlisis</b>	■			
<b>Serveis Consulta i Activació</b>		■	■	◆

Taula 5. Planificació inicial D.21

- **D.21.08. Crèdit:**

1. Tapa d'anàlisis i disseny dels serveis de crèdit.
2. Desenvolupament del mètode de pagament.
3. Desenvolupament del cashAdvance.

Nom	Nov	Des	Gen	Feb	Mar	
<b>Anàlisis</b>	■					
<b>Mètode de pagament</b>			■	■	◆	
<b>CashAdvance</b>				■	■	◆

Taula 6. Planificació inicial D.21.08

- **E.23. Rebuts:**

1. Tapa d'anàlisis i disseny dels serveis dels rebuts.
2. Desenvolupament del pagament anticipat.

Nom	Nov	Des	Gen	Feb	Mar	
<b>Anàlisis</b>			■			
<b>Pagament Anticipat</b>				■	■	◆

Taula 7. Planificació inicial E.23

Tenint una planificació inicial de les dades d'entrega del projecte ens trobem que, a l'hora de realitzar el projecte, les dates d'entrega s'han modificat per endarreriments en l'inici que han fet moure totes les entregues del projecte.

En la planificació inicial no consten alguns serveis, com el de crèdit que està afegit més recentment i alguns dels blocs també queden modificats. S'afegeix temps per el desenvolupament dels projectes i es comencen més tard. L'entrega dels serveis a test queda per a finals de març.

Finalment la planificació queda de la següent manera:

- **D.13. Contractació:**

Nom	Nov	Des	Gen	Feb	Mar
<b>Anàlisis</b>	■				
<b>Serveis Contractació</b>			■	■	◆

Taula 8. Planificació final D.13

- **D.20.02. Operacions:**

Nom	Nov	Des	Gen	Feb	Mar	Abr
Anàlisis	■					
Serveis accés a S. nacional		■		■	◆	
Serveis consulta Oper.				■		◆
Enviament internacional					■	

Taula 9. Planificació final D.20.02

- **D.21. Cicle de vida:**

Nom	Nov	Des	Gen	Feb	Mar	Abr
Anàlisis	■					
Serveis Consulta i Activació				■		◆

Taula 10. Planificació final D.21

- **D.21.08. Crèdit:**

Nom	Nov	Des	Gen	Feb	Mar
Anàlisis	■				
Mètode de pagament				■	
CashAdvance				■	

Taula 11. Planificació final D.21.08

- **E.23. Rebuts:**

Nom	Nov	Des	Gen	Feb	Mar	Abr
Anàlisis		■				
Pagament Anticipat					■	
Llistat de Rebuts					■	
Detall de Rebut					■	

Taula 12. Planificació final E.23

### **3.2. Costos**

Per a l'estimació dels costos del projecte es prepara un document amb l'avaluació econòmica que suposarà el projecte. Aquest cost s'estudia mitjançant les hores previstes per a cada una de les tasques a realitzar.

Per tant, per a fer l'estudi del cost total del projecte, primer s'estudia cada una de les tasques i per cada una es valorarà les hores que suposarà el seu desenvolupament. El sumatori de totes aquestes hores serà l'estimació global del cost del projecte.

Aquesta avaluació s'envia al client el qual tindrà que acceptar la proposta i s'arribarà a un acord per assignar un cost econòmic per a les hores.

Les hores assignades a cada una de les tasques són utilitzades també per la gestió de l'equip i repartir així cada una de les tasques. A més d'utilitzar una eina per el seguiment de les hores en les quals treballa cada membre de l'equip i així poder comprovar al final les hores reals utilitzades en les tasques. Tenint d'aquesta manera un valor real final de les hores gastades en el projecte i poder realitzar un estudi de la rendibilitat del projecte.



## 4. Anàlisi i disseny

En la etapa d'anàlisi i disseny de les 16 diferents APIs es defineix la base del projecte que es desenvolupa. És una etapa de comunicació entre totes les parts que estan implicades en el projecte des del client, el qual demana els requisits, la nostre empresa i altres empreses també implicades en el desenvolupament. Totes les parts es reuneixen i argumenten per a la creació de les APIs.

### 4.1. Anàlisi i presa de requisits

En l'anàlisi del projecte es reuneixen els líders dels equips implicats i el client per acordar una proposta de resolució. Es mantenen un seguit de reunions on es dur a terme un anàlisi de cada un dels requisits que dona el client.

Per cada un dels serveis es fa un estudi de les funcions que realitza:

1. Activació de la targeta (ActivateCard): Servei per a l'activació d'una targeta ja creada utilitzant el seu identificador. Retornarà una resposta segons si l'activitat s'ha realitzat correctament o hi ha hagut un error, informant de l'error corresponent.
2. Detall de la targeta (GetCardDetail): Obtenció de les dades de la targeta mitjançant el seu identificador. S'estudien totes les dades que seran necessàries de retornar, així com també els errors d'execució que es poden detectar.
3. Llistat de targetes (ListCards): Obtenció de la llista de targetes amb un identificador de client, ja puguin ser documents nacionals identificatius del client o amb identificadors propis. Es podrà fer una recerca en el llistat de targetes amb diferents criteris de cerca (tipo o marca de la targeta o depenent de la titularitat del client entre d'altres). Es retorna un llistat de les targetes resultants a la cerca amb les dades de cada un d'elles o l'error resultant de l'execució.
4. Creació del compte de la targeta (CreateCardAccount): Creació del contracte d'una targeta, estudiant totes les dades que són necessàries, tant les que són obligatòries per a dur a terme la funció com les que no ho son. Mitjançant aquestes dades es crearà el compte i si el resultat de l'execució és correcte es retorna l'identificador de la petició del contracte, en cas contrari es retornarà un error informat.
5. Firma del compte de la targeta (CardAccountSign): Firma del contracte. Un cop donat l'identificador de la petició del contracte de la funció anterior es crea una petició de firma del contracte i en el cas que sigui processada correctament es retornarà un identificador del contracte i una llista de les targetes associades a ella.

6. Pagament de rebut (PayStatement): Pagament d'un rebut (pagament d'una obligació o prestació de servei o producte). Per a dur a terme aquesta funció serà necessari un identificador de l'extracte, l'import i el compte de destinació del pagament, juntament amb altres dades opcionals a aportar. La resolució aportada serà un codi d'acceptació del moviment o un missatge d'error.
7. Llistat de comptes de la targeta (ListCardAccount): Petició del llistat de contractes, al igual que el llistat de targetes, es durà a terme mitjançant un identificador del client i la opció de fer una cerca mitjançant diferents criteris (estat del compte o tipo o marca de la targeta entre d'altres). La sortida del servei retornarà el llistat de comptes resultants de la cerca o l'error resultant.
8. Detall compte de la targeta (GetCardAccount): Obtenció del detall d'un contracte mitjançant l'identificador d'aquest. S'estudia la informació que serà necessària retornar del compte i els possibles errors resultants.
9. Llistat transaccions de la targeta (ListCardTransactions): Llistat dels moviments que s'efectuen en una targeta, mitjançant l'identificador de la targeta i amb diferents criteris de cerca (per data, import, liquidació o situació). El resultat de la cerca serà el llistat resultant de la petició de la cerca amb les dades de cada un dels moviments obtinguts.
10. Llistat transaccions del compte de la targeta (ListCardAccountTransactions): Al igual que el servei anterior s'efectua una cerca dels moviments però amb l'identificador del contracte i amb els mateixos criteris de cerca.
11. Detall d'un rebut (GetStatements): Consulta d'un extracte utilitzant l'identificador d'aquest, s'obté com a resposta la informació relacionada amb el rebut (l'estat, forma de pagament, import de pagament...).
12. Cancel·lació de pagament d'un rebut (CancelStatementPayment): Petició per a la cancel·lació del pagament d'un rebut efectuat mitjançant l'identificador de l'extracte i de la petició a realitzar. S'obtindrà a la sortida un codi de l'estat segons si es realitza correctament la petició.
13. Llistat moviments liquidació (ListClearanceTransactions): Llistat dels moviments d'una liquidació d'un contracte a través de l'identificador de la liquidació i uns criteris de cerca com la data o la situació. Retornant el llistat de moviments corresponent a la liquidació segons els possibles criteris de cerca.
14. Llistat de rebuts (ListStatements): Obtenció de la llista de rebuts d'un contracte. Al igual que la resta dels llistats, es fa una petició mitjançant l'identificador del

contracte i amb diferents criteris de cerca (data o estat). El resultat de la cerca serà el llistat resultant d'ella o el codi d'error corresponent.

15. Detall del mètode de pagament (GetPaymentMethod): Consulta de la forma de pagament d'un contracte amb l'identificador d'aquest. En cas de que l'execució sigui correcte es retornarà la informació del mètode de pagament del rebut o l'error informat.
16. Cash Advance (CashAdvance): Servei per a la petició d'un avançament d'efectiu amb les dades d'entrada del identificador de la targeta, iban de la compte destí i l'import a sol·licitar. En cas de que la petició s'executi correctament es retornarà la informació dels diferents imports corresponents a l'avançament d'efectiu.

## 4.2. Disseny tècnic

### 4.2.1. Distribució dels serveis

Les funcions es distribuïran en diferents serveis, els quals contindran els mètodes amb el flux de cada una de les prestacions i es retornarà les dades de sortida corresponents. Aquests serveis són:

- Cards: Mètodes corresponents per a les dades d'una targeta.
- CardsActions: Mètodes per a les accions que es realitzen a la targeta.
- CardsInfo: Mètodes per la informació de targetes.
- CardsAccounts: Mètodes per a les dades d'un contracte.
- CardsAccountsActio: Mètodes per a les accions que es realitzen a un contracte.
- CardsAccountsInfo: Mètodes per a la informació de contractes.
- Statements: Mètodes relacionats a les dades d'un rebut.

Núm. Servei	Servei	Servei Integració	Bloc del projecte
1	ActivateCard	CardsActions	D.21
2	GetCardDetail	Cards	D.21
3	ListCards	CardsInfo	D.21
4	CreateCardAccount	CardsAccounts	D.13
5	CardAccountSign	CardsAccounts	D.13
6	PayStatement	CardsAccountsActio	E.23
7	ListCardAccount	Statements	D.21
8	GetCardAccount	CardsAccountsInfo	D.21
9	ListCardTransactions	CardsAccounts	D.20.02

10	ListCardAccountTransactions	CardsInfo	D.20.02
11	GetStatements	CardsAccountsInfo	E.23
12	CancelStatementPayment	Statements	E.23
13	ListClearanceTransactions	Statements	D.20.02
14	ListStatements	CardsAccountsInfo	E.23
15	GetPaymentMethod	CardsAccountsActio	D.21.08
16	CashAdvance	CardsAccountsActio	D.21.08

*Taula 13. Relació dels serveis*

#### 4.2.2. API RESTful

Per fer un anàlisi extens del disseny tècnic i funcional de la funció es crearà una API REST per cada un dels serveis.

Les operacions més importants en una API REST que segueixen la especificació HTTP (Hypertext Transfer Protocol), el protocol de comunicacions utilitzat a la web, són les següents:

- POST: Creació de nous recursos a la col·lecció.
- GET: Petició de la informació de recursos.
- PUT: Modificació de recursos existents.
- DELETE: Eliminació de recursos.

Cada una de les operacions consta d'una petició (Request) i d'una resposta (Response).

La Request consta de 3 parts importants:

- URL: Adreça única del recurs.
- Mètode HTTP (post, get, put o delete): S'informa del mètode que descriu l'acció a realitzar.
- Headers i Body: Informació a enviar, en el cos s'envien les dades corresponents als recursos.

La Response, al igual que la petició, consta de Headers i Body però la part més important de la resposta es l'Status Code, codi numèric que indica si la petició s'ha realitzat correctament o si s'ha trobat amb un error. Estan agrupats en cinc grups diferents:

- Respostes informatives (100-199)
- Respostes correctes (200-299)
- Redireccionaments (300-399)
- Errors del client (400-499)
- Errors del servei (500-599)

Per a cada una de les APIs, que es crearan seguint aquest protocol, es decidirà quin mètode HTTP seguiran, quin ruta (URL) tindran, com serà l'enviament de dades i la resposta a rebre.

Núm.	Servei	Mètode HTTP	URL
1	ActivateCard	POST	/cards/{cardId}/activate
2	GetCardDetail	GET	/cards/{cardId}
3	ListCards	GET	/cards
4	CreateCardAccount	POST	/cardAccounts
5	CardAccountSign	POST	/cardAccounts/sign/{cardAccountRequestId}
6	PayStatement	POST	/statements/{statementId}/payment
7	ListCardAccount	GET	/cardAccounts
8	GetCardAccount	GET	/cardAccounts/{cardAccountId}
9	ListCardTransactions	GET	/cards/{cardId}/transactions
10	ListCardAccountTransactions	GET	/cardAccounts/{cardAccountId}/transactions
11	GetStatements	GET	/statements/{statementId}
12	CancelStatementPayment	POST	/Statements/{statementId}/paiment/cancel
13	ListClearanceTransactions	GET	/clearances/{clearanceId}/transactions
14	ListStatements	GET	/cardAccounts/{cardAccountId}/statements
15	GetPaymentMethod	GET	/cardAccounts/{cardAccountId}/paymentMethod
16	CashAdvance	POST	/cards/{cardId}/cashAdvance

Taula 14. Rutes dels serveis API

### 4.2.3. APIs Serveis

Per a la creació de cada una de les API dels serveis s'utilitza Apicurio, una eina de codi obert que proporciona un desenvolupament de APIs d'una manera visual, ràpida i amb zero codi a realitzar. Es crea un dipòsit en comú per les APIs dels serveis i se li dona accés a cada membre de l'equip, tant dels responsables de la documentació del projecte com de l'equip de producció, que poden accedir tant per consultar-les com a modificar-les.

A més s'utilitza Confluence, un software de col·laboració d'equips utilitzat per a compartir un espai comú per a tots els diferents equips i empreses que estan implicats en el desenvolupament del projecte. D'aquesta manera es tenen les APIs i totes les dades necessàries per el seu desenvolupament entre equips.

A la llarga l'ús de Apicurio queda absolut degut a la exigència d'haver de comptar amb Confluence, l'espai comú entre les empreses implicades. I per tant es considera innecessari el manteniment de la gestió que suposa que les dades de ambdós llocs siguin equivalents, evitant així crear confusions a l'equip de desenvolupament.

### 4.2.3.1. Exemple API

Cada una de les APIs serà creada seguint la informació del protocol HTTP, en el cas de l'API 14-ListStatements corresponent al llistat de rebuts d'un contracte tindrà la següent informació:

- **Mètode Http:** GET
- **URL:** /cardAccounts/{cardAccountId}/statements
- **Headers:** (Comuns per a tots les APIs)

Nom	Obligatori	Descripció
X-CHANNEL	X	Canal d'accés.
X-APPLICATION		Aplicació on es realitza la crida.
X-UNIT-CODE		Codi de la unitat.
X-BANK.CODE		Codi del banc.
X-OPERATOR-CODE		Codi de l'empleat.
X-LANGUEGE-ID	X	Idioma
X-BRANCH-CODE		Codi de l'oficina.
X-REQUEST-ID	X	Identificador de la petició del sistema d'origen, codi d'operació.

Taula 15. Headers comuns a tots els serveis

- **Les dades d'entrada:**

Nom	Tipo paràmetre	Tipo de dada	Obligatori	Descripció
cardAccountId	Path Param	String(20)	X	Identificador del contracte.
statementStatus	Query Param	String(2)		Estat del rebut.
startDate	Query Param	String(10)		Data d'inici, ISO-8601 (YYYY-MM-DD).
endDate	Query Param	String(10)		Data final, ISO-8601 (YYYY-MM-DD).
key	Query Param	String(200)		Token d'informació que fa referència a la paginació.
keyAction	Query Param	String(1)		Acció a realitzar (P- Previous Page / N - Next page / null- First Page).
externalCustomerId	Query Param	String(16)		Identificador del client extern.

Taula 16. Dades entrada API 14

- **Les dades de sortida:**

Nom	Tipo paràmetre	Descripció
<b>Statements</b>	<b>Llista</b>	
<b>Statement</b>	<b>Bean</b>	
statementId	String(34)	Identificador del rebut
startDate	String(10)	Data d'inici, ISO-8601 (YYYY-MM-DD)
endDate	String(10)	Data final, ISO-8601 (YYYY-MM-DD)
statementStatus	Map<key(2), value(100)>	Estat del rebut (Actiu, Baixa, Pendent, Procés, Impagat, Pagat, Anul·lat, Finalitzat)
<b>StatementBalances</b>	<b>Llista</b>	
<b>StatementBalance</b>	<b>Bean</b>	
type	Map<key(3), value(100)>	Tipo d'import a informar
status	Map<key(2), value(100)>	Estat del rebut (Actiu, Baixa, Pendent, Procés, Impagat, Pagat, Anul·lat, Finalitzat)
<b>StatementBalanceAmount</b>	<b>Bean</b>	
amtPaid	Decimal(15,2)	Import pagat
amt	Decimal(15,2)	Import
ccy	String(3)	Clau de moneda (EUR)
key	String(200)	Token d'informació que fa referència a la paginació.
moreElementsAsc	Boolean	Indica si hi ha elements endavant.
moreElementsDesc	Boolean	Indica si hi ha elements endarrere.
statusCode	Status Code	200: OK / 400: Bad Request / 500: Internal Server Error

Taula 17. Dades sortida API 14

- **Errors:**

Codi	Missatge	HTTPStatus
<b>008</b>	Uninformed Channel. The petition requires the X-CHANNEL.	400
<b>009</b>	Uninformed request id. The petition requires the X-REQUEST-ID.	400
<b>010</b>	Uninformed operator code. The petition requires the X-LANGUAGE-ID.	400
<b>012</b>	El cliente no tiene relación con el contrato.	400
<b>016</b>	El contrato informado no existe.	400
<b>017</b>	No existen datos para la consulta.	204
<b>024</b>	El extracto no pertenece al cliente seleccionado.	400
<b>062</b>	Error parámetros entrada – contrato no informado.	400
<b>145</b>	El valor de la clave de acción para paginar debe ser P- Previous Page / N - Next page / null- First Page	400
<b>146</b>	El formato de las fechas debe ser YYYY-MM-DD.	400
<b>149</b>	El valor del estado del extracto debe ser 10 - Activo/11 – Baja...	400
<b>999</b>	Error interno.	500

Taula 18. Codis d'error API 14

### 4.3. Disseny funcional

Per el disseny funcional de cada una de les APIs creades s'estudia la funcionalitat de cada un dels serveis.

Per a la primera fase del desenvolupament es preparen serveis mocks a desenvolupar. Un servei mock consisteix en una primera fase del desenvolupament en la qual es prova el funcionament dels serveis.

El disseny funcional descriu com ha de ser implementat el servei. En el cas de la API 14 del llistat de rebuts d'un contracte s'implementaran el següent flux:

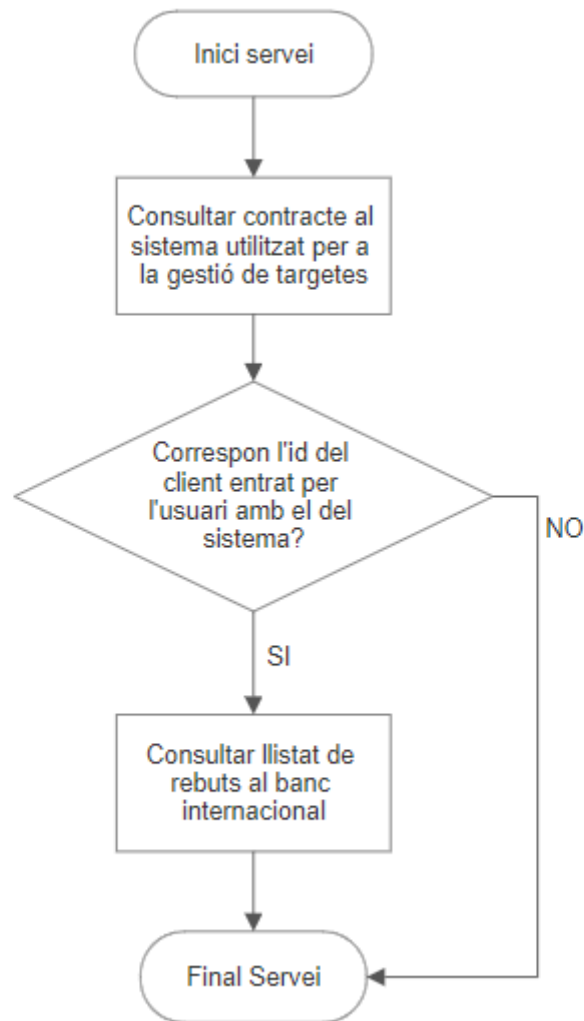


Figura 1. Disseny funcional API 14

En aquest exemple es necessiten crides a diferents backends i serveis. Primerament, a tots els serveis es realitza una crida al backend d'inici de servei i s'hauran de dur a terme dues crides extres. Una crida al sistema utilitzat per a la gestió de targetes que es dur a terme per una altre empresa amb la qual es treballa paral·lelament. I finalment una crida



per a la petició de les dades dels rebuts al sistema del banc internacional, el qual també està gestionat per un altre equip i amb el qual s'estableix una relació de treball.

Les dades obtingudes en la execució de les crides seran les mostrades en la sortida de la petició. Aquest procés a realitzar s'explicarà més detalladament en la següent secció de desenvolupament.

## 5. Desenvolupament

En l'etapa del desenvolupament es passen les decisions preses en l'etapa d'anàlisi, la creació de les APIs i els dissenys funcionals, a una primera versió funcional dels serveis. S'intenten seguir les decisions preses en la mesura que sigui possible, tot i que poden sorgir inconvenients interns i per tant la necessitat de pactar canvis en els dissenys.

Per a la producció dels serveis es necessiten desenvolupar els diferents elements indispensables per el funcionament de cada un d'ells i una bona planificació de l'equip de desenvolupament.

La creació dels serveis es dur a terme amb una connexió pròpia a la xarxa del client bancari per obtenir així una seguretat i restringint l'accés, evitant possibles atacs o accessos no desitjats.

L'entorn de desenvolupament utilitzat per el treball és l'Eclipse, una eina de codi obert multiplataforma que permet la creació d'entorns de treballs propis. El client bancari compta amb una arquitectura pròpia creada que permet el desenvolupament dels diferents serveis, tan dels sistemes d'externalització, d'integració o la part del backend, que s'explicaran més detalladament en els següents apartats. Aquesta arquitectura està creada amb la utilització de Java com a llenguatge de programació, un dels llenguatges de programació més coneguts i per tant facilita l'aprenentatge de noves incorporacions, que només necessiten desenvolupar un estudi del funcionament de l'arquitectura.

### 5.1. Planificació de l'equip

Per tal de realitzar el projecte d'una manera organitzada i eficient es necessita una bona planificació de cada un dels membres de l'equip de desenvolupament. Es tenen en compte diferents factors per tal de gestionar l'equip.

Dintre de l'equip de desenvolupament hi ha un 'team leader', el qual gestiona l'equip i assigna tasques a cada membre. Una bona gestió de l'equip evita molts possibles problemes a l'hora de realitzar el treball individual. De manera que no es repeteixi feina i aquesta es realitzi de la manera més eficient possible.

El 'team leader' també és l'encarregat que durant el desenvolupament del codi generat per cada un dels membres de l'equip, aquests disposin de tota la informació prèviament definida de manera clara, per evitar errors i per tant no perdre temps innecessari en el desenvolupament.

Un cop les tasques estan assignades es treballa sobre un mateix codi. Per a treballar de manera conjunta i sense trepitjar el treball dels companys cada membre de l'equip utilitza Git, una eina per al desenvolupament col·laboratiu de software. D'aquesta manera tots els membres de l'equip poden treballar sobre un mateix codi, pujant en cada moment el seu treball i mantenint-ho actualitzat.

Ja en la part final del desenvolupament es necessita pujar i mantenir actualitzat l'entorn de test dels serveis per tal de tenir la última versió en tot moment a l'hora de realitzar les proves necessàries. En aquesta part és imprescindible mantenir una bona comunicació en tot moment dels membres implicats, per tal d'evitar possibles problemes i pèrdues de codi.

Per a mantenir una bona comunicació es realitzen Daily Meeting, reunions diàries on assisteix cada membre de l'equip i es parla de cada tasca realitzada, de com es porta la feina i de possibles problemes que es puguin ocasionar. L'entorn de comunicació més utilitzat és el 'Microsoft Teams', tant per les reunions i comunicacions grupals com per tenir un espai comú per a la col·laboració d'arxius de tota la informació del projecte.

## 5.2. Parts a desenvolupar

El projecte consta de diferents apartats a desenvolupar per a cada un dels serveis. Per al seu desenvolupament es crea un frontal, el qual mostra la informació a l'usuari, aquest realitzarà les peticions i enviarà la informació al sistema. La informació es processarà mitjançant diferents serveis de sistemes de externalització, encarregat d'enviar dades entre frontal i els sistemes inferiors, sistemes d'integració que processen les dades entre sistemes d'integració i el backend, que poden ser de sistemes propis o de sistemes externs a la empresa.

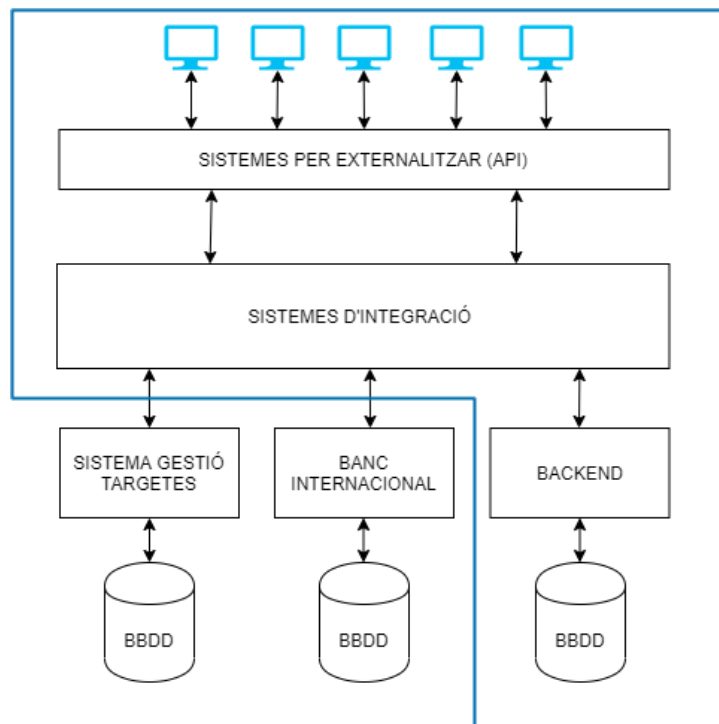


Figura 2. Esquema estructura projecte

### 5.2.1. Frontals

Els frontals de la aplicació estan creats per un equip de treball, encarregats de crear prototips de dissenys, que són presentats i aprovats per el client.

Per tant, per cada una de les pantalles de l'aplicació (les diferents rutes) es creen prototips de disseny i un cop són acceptats per el client es desenvolupen les pantalles finals. Aquests frontals mostren totes les dades necessàries acordades en les APIs. Per exemple, en la petició d'un llistat de rebuts s'entra en la ruta (/cardAccounts/{cardAccountId}/statements), aquest enviarà les dades necessàries al sistema i mostrarà una pantalla en la qual es veuran cada una de les dades definides.

Aquest desenvolupament no és necessari de tenir en compte a l'hora de desenvolupar les següents parts ja que l'entrega dels serveis es dur a terme mitjançant *mocks*, en la qual es comprova simplement que l'enviament de les dades sigui el correcte.

### 5.2.2. Sistemes per externalitzar

Els sistemes d'externalització de dades són creats per un equip de treball juntament amb els sistemes d'integració i els backends.

Aquests sistemes es creen seguint l'especificació de les APIs. Són els encarregats de la conversió de les dades, de manera que reben la informació entrada per l'usuari a través dels frontals, i executen la crida al servei corresponent d'integració, on informen de les dades rebudes i del qual reben la resposta amb les dades necessàries a mostrar en el frontal. Es defineixen dos 'Transfer Objects' (TO), objectes de transferència de dades entre processos, així com la seva relació amb el TOs dels serveis d'integració.

Aquests TOs contindran cada una de les dades d'entrada i sortida definides per la API, seguint el format especificat d'aquestes. El servei obté aquest TO d'entrada i s'encarrega de crear un TO d'entrada per el sistema d'integració el qual es crida amb les mateixes dades. Al igual que rep la resposta amb el TO de sortida del servei d'integració, crea el TO de sortida d'aquest amb les dades retornades i l'envia de resposta al frontal.

El flux del servei queda de la següent manera:

1. TO d'entrada, amb les dades a rebre.
2. Crida al mètode del sistema d'integració utilitzant les dades rebudes en el TO d'entrada.
3. Mètode de sortida que rep el TO de sortida del servei d'integració i es fa el mapeig amb el TO de sortida del servei d'externalització.

Per tal de realitzar les proves per a les primeres entregues dels serveis, en els quals només es realitzen els *mocks*, s'utilitza la crida a aquests serveis d'externalització per assegurar el funcionament correcte de tot el servei.

### 5.2.3. Sistemes d'integració

Els sistemes d'integració s'encarreguen de fer les diferents crides per a l'obtenció i la transformació de les dades necessàries per a adaptar-les a la interfície dels serveis.

Cada un dels mètodes conté un TO d'entrada i un de sortida amb les dades especificades a l'API, iguals als TOs dels sistema d'externalització. Així com un flux de tots els mètodes i invocacions a realitzar. Aquest flux es veurà amb més detall l'esquema que segueix en l'exemple del següent apartat.

El contingut de cada un dels mètodes d'un sistema d'integració segueix el següent patró:

1. TO d'entrada de les dades rebudes per el sistema d'externalització.
2. Flux dels mètodes i les crides als sistemes de negoci del backend o a altres mètodes de sistemes d'integració complementaris.
3. TO de sortida amb les dades obtingudes de les diferents peticions.

Per a cada un dels serveis a desenvolupar es creen dos mètodes dintre del mateix sistema d'integració. El primer mètode serà l'invocat per el sistema d'externalització, és el responsable de comprovar que els camps d'entrada obligatoris vinguin informats i de que el format de les dades rebudes correspongui amb l'especificació de l'API. En cas de trobar errors retorna el missatge d'error corresponent i si tot és correcte invoca al segon mètode, que s'encarrega de les crides a la resta de serveis. En aquest segon mètode es fan crides a altres possibles mètodes de sistemes d'integració i a sistemes de negocis dels backends. En cada una de les crides és necessari fer validacions de les dades d'entrada i de sortida per tal de documentar els possibles errors. Aquestes crides utilitzen els TOs d'entrada i de sortida corresponents a cada un dels mètodes invocats, creant el TO en el cas de la invocació i reben el TO de sortida en el cas de la resposta.

L'esquema de les crides als sistemes d'integració queda de la següent manera:

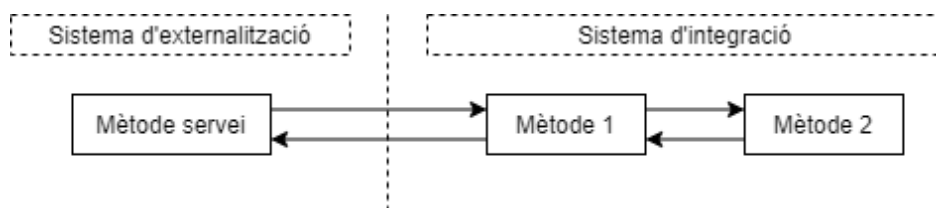


Figura 3. Esquema crides sistema integració

#### 5.2.3.1. Exemple flux sistema d'integració

Per tenir clar com es crea exactament la capa del sistema d'integració d'un servei es mostra un exemple, en el qual s'expliquen els dos fluxos dels mètodes relacionats amb un servei de petició del llistat de rebuts on es fan diferents crides als sistemes (API 14 ListStatements). Aquesta API consta de dos mètodes dintre del sistema d'integració de CardsAccountsInfo, que tal i com s'ha explicat anteriorment conté els mètodes responsables de gestionar la informació corresponent als contractes, en aquest cas el llistat de rebuts d'un contracte.

Per entendre el flux correctament és necessari veure els diferents tipus d'activitats que el creen, cada una marcada amb una lletra i color diferents:

- Les activitats corresponents al mateix mètode del sistema d'integració estan assignades amb la lletra P, són els mètodes creats dintre d'una mateixa classe que gestionen el flux, creen els TOs d'entrada i obtenen els TOs de sortida de cada una de les invocacions externes, prenen les decisions de les activitats a executar i gestionen cada una les dades obtingudes.
- Les activitats a crides a un mètode d'un sistema d'integració es mostren amb el símbol de la lletra A.
- Les activitats que invoquen a un sistema de negoci d'un backend es marquen amb la lletra H.
- Finalment les activitats que mostren el mètode d'error amb la lletra E.

El primer mètode (getStatementSE) és invocat per el sistema d'externalització i en el seu flux consta dels següents mètodes:

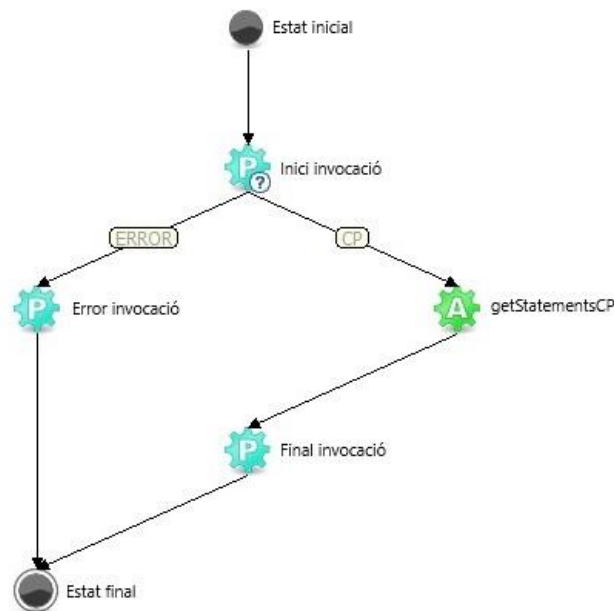


Figura 4. Flux mètode getStatementSE sistema integració API 14

- **Inici invocació:** Rep les dades del TO d'entrada, les processa, comprovant possibles errors, com els format de les dades o que els valors obligatoris vinguin informats. Si totes les dades són correctes es segueix el flux amb la decisió CP la qual crida el segon mètode (getStatementCP), per fer la crida es crea un TO d'entrada del segon mètode on es fa el mapeig de les dades d'entrada. En cas de trobar errors, es guarda l'error trobat i es segueix amb la decisió ERROR.
- **Error invocació:** En el mètode d'error de la invocació es crea el TO de sortida on s'informa de l'error trobat en el mètode anterior.

- **Crida al segon mètode:** Fa la crida al segon mètode.
- **Final invocació:** El mètode de la invocació final rep la sortida de la invocació i fa el mapeig del TO de la sortida rebuda al TO de sortida del mètode, tan si es rep un error, com el llistat de les dades finals.

El segon mètode del sistema d'integració (getStatementCP) invocat per el mètode anterior conté el següent flux:

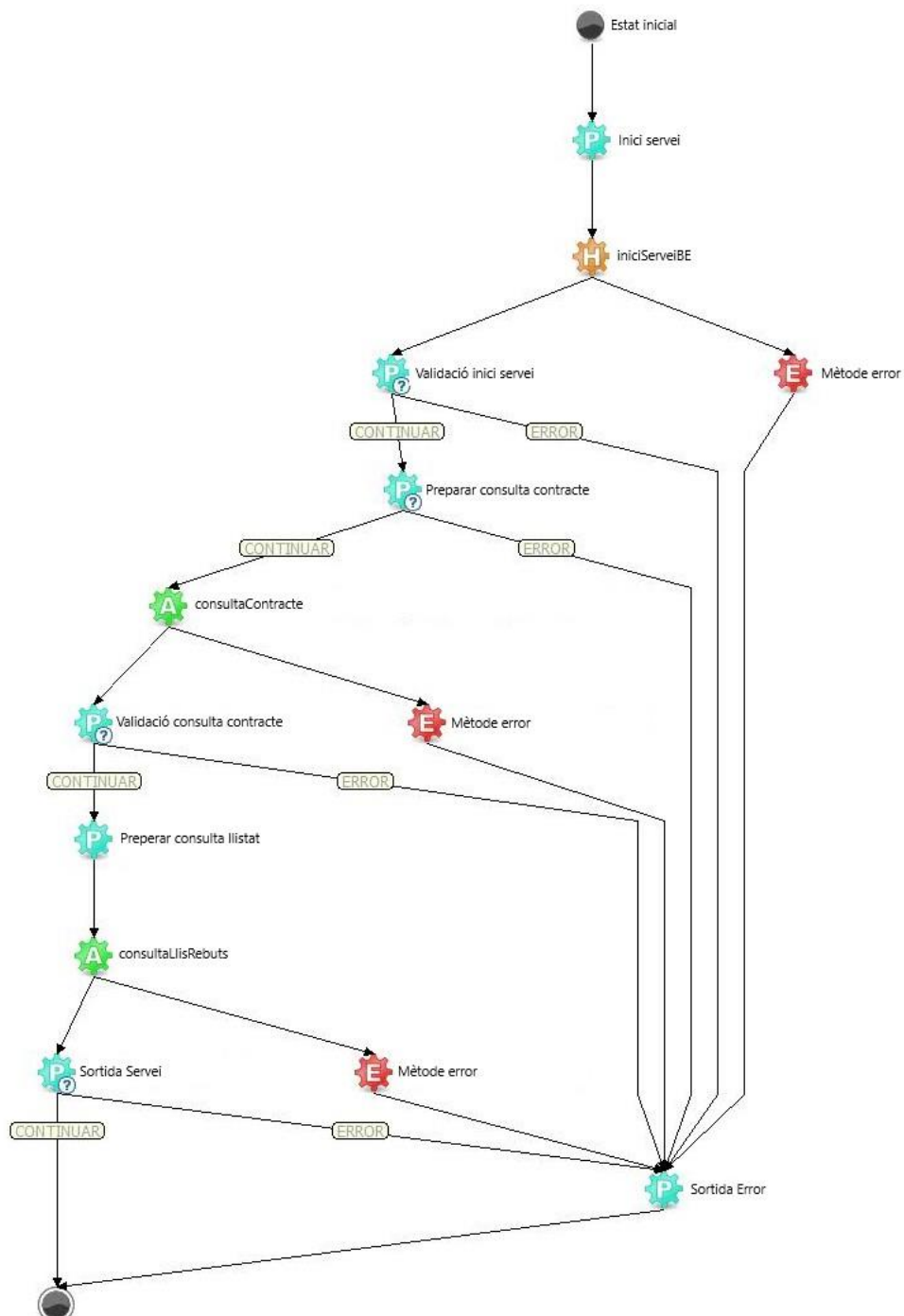


Figura 5. Flux mètode getStatementCP sistema integració API 14

- **Inici servei:** Mètode invocat per el `getStatementSE` del qual rep el TO d'entrada amb les dades introduïdes, fa la crida al mètode d'inici del servei al servei del backend (`iniciServeiBE`) i inicialitza el seu TO d'entrada.
- **Crida backend inici servei:** Es realitza la invocació a un backend propi responsable de la inicialització dels serveis. Si s'ocasiona un error intern es retorna la sortida al mètode error i en el cas de que tota la execució sigui correcte la sortida es retornarà al mètode de validació d'inici servei.
- **Validació inici servei:** Rep el TO de sortida del mètode `iniciServeiBE` i comprova que l'execució no retorni un error informat. Si es així, es pren la decisió Error i es segueix amb l'activitat de la sortida d'error. Si tot està correcte es continua al mètode per a preparar la consulta del contracte.
- **Preparar consulta contracte:** A l'entrada del servei es rep de manera obligatòria l'identificador del contracte del qual es fa la petició del llistat de rebuts i de manera opcional és possible rebre un identificador del client extern. En cas de que sigui informat, és necessari fer una comprovació al sistema de gestió de targetes per tal de verificar que el contracte correspon al client. Per fer aquesta comprovació s'invoca al mètode de consulta de contracte, que correspon a un sistema d'integració complementari encarregat de les crides al sistema de gestió de targetes. Inicialitza el TO d'entrada del mètode amb l'identificador del contracte (`cardAccountId`) rebut.
- **Invocació consulta contracte:** S'invoca el mètode `consultaContracte` del sistema d'integració responsable de les crides al sistema extern de gestió de targetes. Igual que a totes les invocacions és possible rebre la sortida d'error o al mètode de validació.
- **Validació consulta contracte:** Rep el TO sortida de la invocació de la consulta del contracte, on s'informa de l'identificador del client corresponent al contracte. Es comprova que l'identificador obtingut sigui el mateix que l'`externalCustomerId` de la entrada del servei. En cas de que no corresponguin es retorna un error informat i es segueix per la sortida error. En cas de que el client del contracte correspongui amb el client entrat se segueix la execució per a llistar els rebuts del contracte.
- **Preparar consulta llistat rebuts:** Amb les dades d'entrada del TO d'entrada del servei es crida al servei d'integració responsable de la invocació al backend local per a la petició del llistat de rebuts. Per a fer la petició es necessita l'identificador del contracte i de manera opcional els possibles filtres del llistat, que poden ser les dates d'activació/baixa i/o la situació del contracte (Actiu, Baixa, Pendent, En procés, Impagat, Pagat, Anul·lat o Finalitzat). Per a la paginació, en cas de ser la primera pàgina, no és necessari informar de cap camp. En les altres pàgines s'informarà de l'acció a realitzar (`keyAction`) amb els valors N si es la següent o P si es l'anterior, i de



la clau (key) corresponen a la pàgina actual, que també ve informada a l'entrada del TO del servei. Amb tota aquesta informació es crea el TO d'entrada per a la invocació del mètode consultaLlisRebuts.

- **Invocació mètode llistat de rebuts:** Invocació al mètode de consultaLlisRebuts del sistema d'integració complementari encarregat de les crides al sistema de negoci del backend propi, on estan guardades les dades relacionades amb el llistat de rebuts indicat. La resposta del servei tan pot ser retornada per la sortida del mètode d'error com un TO de sortida del servei per el mètode de sortida del servei.
- **Sortida servei:** L'activitat de sortida del servei rep el TO de sortida de la invocació al llistat de rebuts i amb les dades rebudes crea el TO de sortida que serà retornat finalment al primer mètode invocat per el sistema d'externalització. Es controla si la sortida és correcte i retorna un llistat o si en cas contrari, la sortida retornada per el backend del llistat de rebuts ha retornat un error.
- **Mètode error:** En l'activitat del mètode error resultant dels possibles errors ocasionats en les crides dels serveis es controla i es processa l'error rebut.
- **Sortida error:** La sortida error s'encarrega de crear el TO de sortida del servei en cas de que s'hagi originat qualsevol error. En la sortida d'error s'informa del codi d'error, el missatge corresponen i el codi d'estat. Els codis d'errors possibles que es poden ocasionar són els informats per l'API mostrada en l'apartat anterior d'exemple d'API.

#### 5.2.3.2. Connexió amb serveis externs

Per a la connexió amb els serveis externs a l'empresa, s'acorden un seguit d'infraestructures de backend, que seran creades per les empreses associades, seguint la mateixa arquitectura del client bancari.

Un cop acordat l'enviament de dades i amb el servei extern creat es pot cridar per fer la petició de les dades necessàries.

Hi han dues possibles crides a serveis externs, al servei del banc internacional i al sistema de gestió de targetes.

Per administrar totes les peticions a realitzar al servei de gestió de targetes es crea un sistema d'integració encarregat de fer les crides a aquests. De manera que en cada una de les invocacions a realitzar a aquests serveis externes es dirigirà primerament aquest nou servei d'integració. Aquest sistema disposarà d'un mètode, el qual serà responsable d'enviar les sol·licituds amb les dades corresponents als serveis creats per empreses externes. Cada una d'aquestes peticions externes tenen una missatgeria d'entrades i sortides diferents, de manera que es crea un model de dades base per a per a facilitar així la utilització.

#### 5.2.4. Backend

Per a la petició de certes dades es necessita la creació de backends propis, aquests backends contenen un sistema de negoci i objectes d'accés a dades (DAO).

Els sistemes de negoci estan formats per diferents operacions/mètodes que construeixen i parametritzen les invocacions del backend. Reben un TO d'entrada amb la informació necessària, aquesta informació és utilitzada per la següent crida al DAO, del qual rep un TO de resposta, finalment gestiona les dades obtingudes per el DAO i les retorna al TO de sortida d'aquest al sistema per el qual ha estat invocat.

El sistema de negoci invoca a un DAO (Objecte d'accés a dades) del mateix backend, aquest DAO conté les queries SQL (Structured Query Language) necessàries a executar sobre les taules SQL. Per cada una de les modificacions o consultes de les dades es necessitaran crear les peticions SQL corresponents. Amb les dades obtingudes es crea el TO de sortida del DAO, rebut per el sistema de negoci.

Així doncs el procés final que es realitza en el backend és el següent:

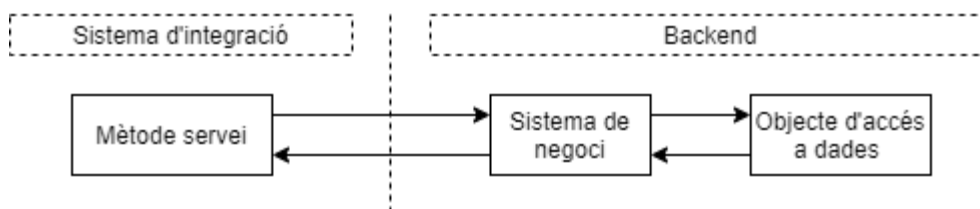


Figura 6. Esquema crides backend

#### 5.3. Metodologia de desenvolupament

Per el desenvolupament de cada un dels serveis és necessari crear els diferents mètodes i serveis indicats.

La metodologia individual a seguir un cop es rep la tasca a desenvolupar és la següent:

1. Estudi de la informació del servei, entendre correctament quines són les parts necessàries a desenvolupar, la funció que es realitza i com gestionar cada informació.
2. Creació, si s'escau, del backend relacionat al servei, en cas de que no estigui ja creat per un equip extern. En cada un dels serveis es realitza una crida al backend d'inicialització dels serveis. Aquest backend només es desenvolupa al principi del projecte i per tant no cal tenir en compte cap desenvolupament extra.
  - 2.1. Creació de l'objecte accés a dades (DAO) i de les funcions de SQL necessàries per a l'execució del servei.

- 2.2. Creació del sistema de negocis amb la invocació al DAO, TO d'entrada i sortida i mapeig de les dades.
3. Creació o modificació de possibles sistemes d'integració implicats en el servei, com els relacionats a les crides a serveis externs (sistema de gestió de targetes o el servei del banc internacional).
  4. Creació dels mètodes del sistema d'integració, TO d'entrada i sortida, flux del servei amb les invocacions als serveis ja prèviament creats o modificats i lògica interna del servei.
  5. Creació del mètode del sistema d'externalització, TO d'entrada i de sortida i mapeig de les dades rebudes del sistema d'integració.
  6. Proves prèvies a la pujada del servei creat a l'entorn de test, per a la comprovació del correcte funcionament de cada un dels sistemes creats. Es realitza desplegant el servei mock en un servidor local, en el qual es pot accedir mitjançant el navegador, per ingressar les dades d'entrada i rebre la sortida esperada del servei. L'ordre de publicació dels serveis al servidor és important per evitar problemes de dependències en ells. Primer es realitza la publicació dels serveis de backend, posteriorment els sistemes d'integració i finalment els sistemes d'externalització.
  7. Arreglar els possibles errors o modificacions a fer sobre el servei, provant el resultat en el servidor local.
  8. Pujada de tots els sistemes creats o modificats durant la creació del servei a l'entorn de test.

## 6. Proves, avaluació i resultat

L'etapa final del projecte consisteix en la creació de casos de proves de cada un de les APIs per els diferents entorns i en el desplegament final dels projectes a producció.

### 6.1. Plans de proves

Per cada una de les APIs es creen els diferents plans necessaris a l'hora de presentar els serveis. Aquests casos de proves són pactats conjuntament amb el client i les diferents parts afectades per el servei, ja que ambdues les tenen en compte prèviament durant el desenvolupament.

Cada un dels casos de proves queda relacionat amb la API i la sortida del servei, com també amb la relació als codis d'errors possibles per cada una d'elles. Així doncs, es creen casos de proves tenint en compte les dependències dels serveis i els errors d'execució prèviament definits.

#### 6.1.1. Exemple pla de proves API

Analitzant l'API 14 Llistat de rebuts es decideix els casos de proves a realitzar:

1. Llistat de rebuts sobre un identificador de contracte. El resultat esperat és un codi 200 i un cos de sortida amb els camps especificats. Es realitza la comprovació del format dels camps de sortida per tal que corresponguin amb els especificats a la API.
2. Llistat de rebuts sobre un identificador de contracte combinant els 4 possibles filtres existents (estat del rebut, data d'inici, data de finalització i identificador del client extern). S'espera un codi 200 de resultat amb un cos de sortida que contingui els camps fixats. També és necessari realitzar la comprovació del format de cada un dels camps.
3. Llistat de rebuts d'un contracte que no conté rebuts. És necessari comptar amb l'identificador d'un contracte que no tingui rebuts. El resultat esperat és un codi d'estat 204 amb el codi d'error (017: No existeixen dades per la consulta).
4. Llistat de rebuts d'un contracte no existent. S'informa d'un identificador de contracte inexistent. S'espera una sortida amb codi d'estat 400 i l'error (167: No existeix el contracte).
5. Llistat de rebuts d'un contracte amb un identificador de client extern no relacionat amb el contracte. És necessari informar tant de l'identificador del contracte com el de client extern. La sortida esperada és un codi 400 amb l'error (012: El client no té relació amb el contracte).

6. Llistat de rebuts d'un contracte amb paginació. És necessari comptar amb un identificador de contracte que contingui els suficients rebuts com per executar la paginació. Amb l'identificador corresponen s'executa el servei amb paginació cap endavant i endarrere, comprovant que cada una de les accions s'executen correctament amb el codi d'estat de sortida a 200 i el cos corresponent.
7. Llistat de rebuts d'un contracte sense paginació. Execució del servei sobre un identificador en el qual no s'executa la paginació. El resultat previst a obtenir és un codi 200 amb el cos de sortida que mostri els rebuts obtinguts.

## **6.2. Proves test**

Un cop els serveis estan desenvolupats i provats en el servidor local es puguen a l'entorn de test. Es segueix un horari per a fer les pujades, amb tres hores de pujades diferents al llarg del dia. Les pujades tarden una mitjana de 4 hores a estar carregades a l'entorn, per tant, s'ha de tenir en compte que a més de tenir només tres hores de pujada, hi ha un període de temps d'espera per a poder provar-los.

Per cada un dels serveis es realitzen les proves d'acceptació ja determinades per el pla de proves així com un seguit de proves de regressió, que tenen l'objectiu de descobrir errors i carències de funcionalitat, forçant els serveis per buscar tots els possibles errors que es poguessin generar durant l'execució.

Es generen documents de proves per cada un dels serveis seguint els plans de proves ja especificats. Aquests documents mostren captures de les dades d'entrada utilitzades, així com el resultat previst a obtenir i el resultat final que s'obté. També s'inclou qualsevol observació que sigui necessària d'expressar. Finalment són compartits en un espai comú de tots els equips de treball implicats.

### **6.2.1. Exemple proves realitzades API**

En el cas de l'API 14 es crea el document d'evidències seguint els 7 casos previstos en el pla de proves.

El primer cas a documentar consisteix en obtenir un resultat "OK 200" juntament amb el llistat de rebuts mitjançant un identificador del contracte. En aquest cas només es demana documentar l'identificador de contracte.

L'entrada enviada al servidor conté l'identificador en la URL (00020003000000015148) i es genera la crida al servei amb la informació d'entrada (Figura 7). La sortida obtinguda conté l'estat en el qual es troba, en aquest cas "OK 200" juntament amb un JSON de sortida amb el llistat dels rebuts resultants de l'execució (Figura 8).

S'observa que la prova ha resultat correctament amb la comprovació de totes les dades de sortida obtingudes, en les quals no es troben incongruències (valors de paginació correctes, format de les dates seguint la API especificada, objectes mapejats amb els valors correctes..).

```
GET https://xxxxx/cardAccounts/00020003000000015148/statements HTTP/1.1
Accept-Encoding: gzip, deflate
X-APPLICATION: 1
X-UNIT-CODE: 1
X-BRANCH-CODE: 1
X-LANGUAGE-ID: ES
X-REQUEST-ID: 1234567890
X-BANK-CODE: 1
X-CHANNEL: 0002
X-OPPERATOR-CODE: 1
Host: xxxxx
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Figura 7. Dades invocació prova execució correcte

```
{
  "key": "001      200019876543210987654321000008",
  "moreElementsAsc": true,
  "moreElementsDesc": false,
  "Statements": [
    {
      "statementId": "2000198765432109876543210000089",
      "endDate": "2019-02-20",
      "startDate": "2019-02-01",
      "statementStatus": {"19": "Pagado"},
      "StatementBalances": [
        {
          "status": {"17": "Informativo"},
          "type": {"100": "Deuda total"},
          "StatementBalanceAmount": {
            "amt": 2000,
            "ccy": "EUR",
            "amtPaid": 0
          }
        },
        [...]
      ]
    },
    [...]
  ],
  [...]
}
}
```

Figura 8. Dades sortida prova execució correcte



#### **6.4. Desplegament**

Com a projecte futur, un cop acabat el procés de preproducció amb totes les proves efectuades, es desplegaran els serveis a l'entorn final de producció.

Un equip extern serà l'encarregat de realitzar les proves internes dels serveis i preparar la producció final d'aquest. Per tal de poder realitzar les proves serà necessari habilitar un seguit de targetes i comptes de prova d'una manera controlada, per tal de no generar cap tipus de problema a l'hora de l'entrega del producte.

Finalment ja es pot donar per acabat el desenvolupament i començar la distribució final de les targetes de prepagament.

#### **6.5. Avaluació**

Per a realitzar una avaluació final del projecte, es dur a terme un anàlisi de les hores empleades per l'empresa i les hores pactades amb el client, les quals seran les cobrades. És interessant obtenir un balanç positiu de cara a l'empresa per assegurar la rendibilitat del projecte.



## 7. Conclusions i treball futur

En resum, el treball realitzat ha estat el desenvolupament d'un seguit de serveis corresponents a un projecte de creació de targetes de prepagament. Creació seguint la especificació d'una API preparada pensant amb possibles projectes futurs i per tant, emprant un temps extra per a la seva preparació. Creant així una aplicació més versàtil i independent, que no depèn totalment del client actual. Així doncs, alhora d'acceptar a possibles propers clients, l'adaptació del projecte actual compondria un menor temps de realització.

El treball personal dut a terme ha estat com a suport en diferents parts dels apartats explicats. Ajuda en la creació de la documentació a entregar al client, així com el suport en la creació de la API per a la plataforma d'Apicurio. Creació dels mètodes ens els diferents sistemes d'externalització i integració d'algunes de les APIs esmentades. Realització d'alguns dels plans de proves amb la creació de la documentació d'aquests. I finalment donant suport als errors ocasionats al llarg de les proves i post entregues.

Analitzant la realització del treball per part de l'equip s'han trobat diferents problemes al llarg del projecte que han suposat un obstacle alhora d'avançar de la manera més eficient. Per una banda, el temps emprat per l'anàlisi i documentació del projecte ha suposat un temps més extens del previst, endarrerint així el treball de l'equip de desenvolupament. També s'han trobat problemes en la realització de la documentació, com errors en escrits, que han suposat la reescriptura d'aquests, i per tant més temps perdut.

D'altra banda també s'han trobat problemes amb l'equip de desenvolupament, degut a la falta d'escalabilitat de les responsabilitats en el projecte, on la gran càrrega de responsabilitats requeia en un sol membre de l'equip, i per tant la dificultat per cada un dels membres de tenir la informació necessària en cada moment. Per a la resolució del problema entra a l'equip un nou membre com a suport per a la repartició de la feina de gestió de l'equip de desenvolupament. L'entrada suposa una millora de l'equip i ajuda en la eficiència general de l'equip. Això si, l'entrada arriba en les portes de l'entrega del projecte, on encara gran part del desenvolupament estava pendent. Aquest fet porta a una implicació extra de l'equip per a poder realitzar l'entrega dintre del termini establert.

Realitzant un anàlisi autocrític d'aquest fet, es podria haver evitat amb una reclamació individual prèvia a l'arribada d'un estat crític. Així doncs, la gestió i la iniciativa individual dels membres de l'equip podria haver evitat el problema.

Com a treball futur queden pendents encara un seguit de serveis a desenvolupar per a completar el producte final. Tenint en compte que l'equip s'ha trobat amb els problemes ja esmentats i han estat resolts, els treballs pendents compten amb aquest aprenentatge, per tant es preveu un millor rendiment. També queda pendent el total desplegament del servei, ja que només han estat desplegats en l'entorn de preproducció fins el moment de comptar amb el servei total i realitzar les proves a producció.

## 8. Referències

Gabriele Ferluga. ¿Tiene todavía sentido utilizar las tarjetas de crédito de prepago?. A: Elpais. Publicat el 13 Abril 2018. URL: [https://elpais.com/economia/2018/04/10/actualidad/1523354233\\_429586.html](https://elpais.com/economia/2018/04/10/actualidad/1523354233_429586.html). [Consultat el 02/04/2020]

Juan del Real Martín. Las tarjetas prepago. A: FinLit. URL: <https://www.finlit.es/las-tarjetas-prepago/>. [Consultat el 02/04/2020]

Lukas Rosenstock. Designing APIs with Apicurio Studio. A: Dzone.com. Publicat el 15 de Març de 2018. URL: <https://dzone.com/articles/designing-apis-with-apicurio-studio>. [Consultat el 09/05/2020]

Rest API Tutorial. A: Restfulpi.net. URL: <https://restfulapi.net/> [Consultat el 07/05/2020]