

## Treball final de grau

**Estudi:** Grau en Enginyeria Informàtica

**Títol:** Digitalització d'una botiga de llaminadures

**Document:** Memòria

**Alumne:** Jordi Rubió Vert

**Tutor:** Dr. Ignacio Clemente Martin Campos

**Departament:** Informàtica, matemàtica aplicada i estadística

**Àrea:** Llenguatges i sistemes informàtics

**Convocatòria (06/21)**

## Index

|                                                                      |    |
|----------------------------------------------------------------------|----|
| 1. Introducció, motivacions, propòsit i objectius del projecte ..... | 5  |
| 1.1. Introducció .....                                               | 5  |
| 1.2. Motivacions .....                                               | 5  |
| 1.3. Propòsit .....                                                  | 6  |
| 1.5. Objectius del projecte .....                                    | 6  |
| 1.6. Objectius del TFG .....                                         | 7  |
| 2. Estudi de viabilitat .....                                        | 8  |
| 2.1. Recursos necessaris pel desenvolupament del projecte .....      | 8  |
| 2.2. Pressupostos Inicials .....                                     | 9  |
| 2.2.1. Hosting .....                                                 | 9  |
| 2.2.2. Domini i certificats .....                                    | 9  |
| 2.2.3. Altres .....                                                  | 10 |
| 2.2.4. Cost de desenvolupament del sistema .....                     | 10 |
| 2.2.5. Total estimació inicial .....                                 | 10 |
| 2.3. Viabilitat Tecnològica .....                                    | 11 |
| 2.4. Viabilitat econòmica .....                                      | 12 |
| 2.5. Resultats de l'estudi de viabilitat .....                       | 14 |
| 3. Metodologia .....                                                 | 15 |
| 3.1. Metodologies àgils .....                                        | 15 |
| 3.2.2. Mètode Kanban .....                                           | 16 |
| 3.3. Versionat del software .....                                    | 17 |
| 4. Planificació .....                                                | 18 |
| 4.1. Com he planificat el projecte .....                             | 18 |
| 4.2. Anàlisi funcional de les diferents parts del projecte .....     | 19 |
| 4.2.1. Back-office .....                                             | 19 |
| 4.2.2. Front-office (web) .....                                      | 19 |
| 4.3. Desglossament en tasques .....                                  | 19 |
| 4.4. Estimació del temps .....                                       | 20 |
| 5. Marc de treball i conceptes previs .....                          | 21 |
| 5.1. Estudis .....                                                   | 21 |
| 5.1.1. Estudi del consumidor .....                                   | 22 |
| 5.1.2. Estudi dels comerços electrònics .....                        | 25 |
| 5.2. Informàtica en el núvol .....                                   | 26 |
| 5.2.1. Allotjament web .....                                         | 26 |
| 5.3. Stack tecnològic .....                                          | 27 |

|                                                         |    |
|---------------------------------------------------------|----|
| 5.4. API .....                                          | 27 |
| 5.4.1. Classificació segons accés.....                  | 27 |
| 5.4.2. Classificació segons protocol.....               | 28 |
| 5.5. Programació asíncrona.....                         | 29 |
| 5.6. Arquitectura MVVM.....                             | 29 |
| 5.7. DNS (Domain Name System).....                      | 30 |
| 5.8. Comerç electrònic .....                            | 30 |
| 5.9. Altres conceptes relacionats amb el projecte ..... | 30 |
| 5.10. MVP .....                                         | 31 |
| 6. Requisits del sistema.....                           | 32 |
| 6.1. Requisits Generals.....                            | 32 |
| 6.2. Requisits Back-office .....                        | 32 |
| 6.3. Requisits front-office.....                        | 33 |
| 7. Estudis i decisions.....                             | 34 |
| 7.1. Llenguatges de programació i llibreries .....      | 34 |
| 7.1.1. Base de Dades .....                              | 34 |
| 7.1.2. Front Office .....                               | 34 |
| 7.1.3. Back Office.....                                 | 36 |
| 7.1.4. API .....                                        | 36 |
| 7.1.5. Altres .....                                     | 36 |
| 7.2. Eines de treball.....                              | 39 |
| 7.2.1. Generals .....                                   | 39 |
| 7.2.2. Front Office .....                               | 40 |
| 7.2.3. Back Office.....                                 | 40 |
| 7.2.4. API i BD.....                                    | 41 |
| 7.3. Estudi de la competència .....                     | 42 |
| 8. Anàlisi i disseny del sistema .....                  | 43 |
| 8.1. Anàlisi .....                                      | 43 |
| 8.1.1. Estructura general i comunicació .....           | 43 |
| 8.1.2. Casos d'ús.....                                  | 44 |
| 8.1.2. Existència de dades .....                        | 46 |
| 8.1.3. API .....                                        | 47 |
| 8.1.4. Front office .....                               | 49 |
| 8.1.5. Back office .....                                | 54 |
| 8.2. Disseny dels diferents sistemes .....              | 61 |
| 8.2.1. Back-office.....                                 | 61 |

|                                         |     |
|-----------------------------------------|-----|
| 8.2.2. Front-office.....                | 63  |
| 9. Implementació i proves.....          | 67  |
| 9.1. Base de dades.....                 | 67  |
| 9.1.1. Problemes i solucions.....       | 67  |
| 9.1.2. Model actual.....                | 67  |
| 9.2. API .....                          | 68  |
| 9.2.1. Problemes i solucions.....       | 68  |
| 9.2.2. Diagrama de classes .....        | 70  |
| 9.3. Front office .....                 | 71  |
| 9.3.1. Conceptes de React.....          | 71  |
| 9.3.2. Problemes i solucions.....       | 72  |
| 9.3.3. Estructura dels components ..... | 77  |
| 9.3.4. Codi més rellevant.....          | 79  |
| 9.4. Back office .....                  | 81  |
| 9.4.1. Problemes i solucions.....       | 81  |
| 9.4.2. Algorismes més rellevants.....   | 86  |
| 9.5. Testing.....                       | 89  |
| 9.5.1. API .....                        | 89  |
| 9.5.2. Front-office.....                | 89  |
| 9.5.2. Back-office.....                 | 89  |
| 10. Implantació i resultats.....        | 90  |
| 10.1. Implantació backend .....         | 91  |
| 10.1.1. Base de dades.....              | 91  |
| 10.1.2. API .....                       | 94  |
| 10.1.3. Back-office.....                | 95  |
| 10.2. Implantació frontend .....        | 96  |
| 10.3. Ajustaments finals.....           | 97  |
| 10.4. Resultats.....                    | 98  |
| 10.4.1. Back-office.....                | 98  |
| 10.4.2. Front-office.....               | 106 |
| 11. Conclusions.....                    | 111 |
| 12. Treball futur.....                  | 113 |
| 13. Bibliografia .....                  | 114 |
| 14. Manual d'instal·lació .....         | 117 |

# 1. Introducció, motivacions, propòsit i objectius del projecte

## 1.1. Introducció

La transformació digital és cada cop més necessària per la supervivència dels petits comerços i la pandèmia ha accelerat encara més aquesta necessitat.

La plataforma de venda que estic desenvolupant pot suposar l'entrada al mercat digital de forma senzilla per molts petits negocis, combinant un sistema d'encàrrec i recollida a botiga amb un de venda 100% online amb enviament (en futures fases del projecte).

## 1.2. Motivacions

La motivació del projecte apareix durant la pandèmia, al veure com la feina dura i constant de molts botiguers es veia fortament afectada i perillava la seva continuïtat.

La meua mare té una botiga de llaminadures que, com tantes altres, ha notat una caiguda significativa de la facturació.

Vaig començar a buscar plataformes que permetessin crear una botiga online de forma senzilla (com Shopify), però cap de les que vaig trobar s'adaptava a la casuística del negoci (compra per pes, productes que poden vendre's per pes o per unitats depenent de si es compren en pack o de forma individual, recollida a botiga en comptes d'enviament, etc).

Per tant, vaig decidir crear-ne una jo mateix.

## 1.3. Propòsit

Existeix una versió de la web que està en producció (<https://can-balots.web.app>) és un MVP del projecte, desenvolupat durant el darrer mes de confinament i mesos posteriors.

En aquesta versió tots els productes estan incrustats dins la web (no es comunica amb base de dades), i la forma de processar les comandes és enviant un e-mail amb HTML amb els detalls d'aquesta.

Tenir aquesta web operativa ens ha permès analitzar el comportament dels usuaris mentre navegaven i les necessitats no cobertes que tenen.

## 1.5. Objectius del projecte

Aquesta eina està especialment enfocada a comerços d'alimentació (per les casuístiques concretes que tenen, tot i que pot ser utilitzat per qualsevol negoci).

La intenció és aportar valor en la gestió i la fase de compra, tal com podem veure descrit a continuació.

### Per al botiguer

Oferir al botiguer una forma senzilla de gestionar el que es mostra a la web, permetent crear/modificar/desactivar productes i visualitzar les comandes realitzades per la web.

Tenir visibilitat a Internet ajuda a captar nous clients potencials i potencialment a augmentar la conversió.

### Per al consumidor

Oferir una alternativa online per realitzar compres amb la comoditat de la web, però a la vegada ajudant al petit comerç i fent compra de proximitat.

Es vol implementar un model que permeti al client decidir què prefereix:

- Compra online amb enviament.
- Encàrrec online amb recollida a la botiga. (pagament online o a la botiga)
- Encàrrec online i recollida a un dels punts designats (fase més avançada que implicaria un cost menor de distribució, però que de moment no es contempla).
- Estalvi de temps.

En el context de pandèmia que hem viscut també aporta un punt extra de seguretat al client del comerç local. Si decideix recollir l'encàrrec a la botiga, s'estalviarà cues i manipulació de producte.

## 1.6. Objectius del TFG

L'objectiu principal és separar el projecte de la següent manera:

|               |                                                                  |
|---------------|------------------------------------------------------------------|
| Back-office   | Eina privada per el botiguer per gestionar productes i comandes. |
| Front-office  | Web que visualitzarà el client (estirarà les dades d'una API).   |
| API           | Capa intermitja entre la base de dades i el front-office.        |
| Base de dades | Contenedor per la informació                                     |

Durant l'elaboració d'aquest treball em centraré en muntar l'estructura i deixar-ho tot funcionant en local, i si és possible també realitzaré el deploy al núvol, però això ja escapa de l'abast del TFG.

## **2. Estudi de viabilitat**

En aquest apartat justificaré quins són els recursos que necessitaré, així com els costos estimats que tindrà el projecte i una petit estudi de viabilitat tecnològica i econòmica.

### **2.1. Recursos necessaris pel desenvolupament del projecte**

Per part de la botiga de llaminadures necessitaré:

- Informació dels productes (nom, pes, preu unitat, preu kg, imatge, etc)
- Correu electrònic que vincularem amb l'enviament de comandes i amb tots els serveis de tercers que s'hauran de contractar.

També necessitaré:

- Icones (utilitzaré les que ofereix la llibreria Material UI).
- GIF de càrrega per l'aplicació de back-office.

El component més important a l'hora de desenvolupar aquest projecte és temps, ja que utilitzaré varies tecnologies diferents, de les quals només he treballat amb dos. Per tant, hauré de dedicar bastant temps a formar-me i buscar informació.



## 2.2. Pressupostos Inicials

Durant la fase de desenvolupament no hi ha costos associats al projecte.

En la fase de producció si que hi hauran certs costos associats, que es detallen a continuació.

### 2.2.1. Hosting

S'ha optat per utilitzar Amazon Web Services (AWS en endavant) ja que ofereix molts serveis integrats a un cost prou assumible.

Tot i oferir una capa gratuïta durant un any, he estimat la inversió necessària (anual) passat aquest termini:

| Services (2)                                                                                                                                              |          |           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-----------|
| <b>Amazon Simple Storage Service (S3)</b>                                                                                                                 |          |           |
| Region: EU (Paris) <span>Edit</span> <span>Action ▾</span>                                                                                                |          |           |
| S3 Standard storage (20 GB per month)                                                                                                                     | Monthly: | 0.57 USD  |
| DT Inbound: Internet (5 GB per month), DT Outbound: Internet (5 GB per month)                                                                             | Monthly: | 0.36 USD  |
| <b>Amazon EC2</b>                                                                                                                                         |          |           |
| Region: EU (Paris) <span>Edit</span> <span>Action ▾</span>                                                                                                |          |           |
| <b>Quick estimate</b>                                                                                                                                     |          |           |
| Operating system (Linux), Quantity (1), Pricing strategy (EC2 Instance Savings Plans 1 Year No Upfront), Storage amount (20 GB), Instance type (t2.small) | Monthly: | 15.75 USD |
|                                                                                                                                                           | Upfront: | 0.00 USD  |

Total → 15,75\$ + ~1\$ / mes → 201\$ / any → 164,35€ / any (a partir del segon any).

### 2.2.2. Domini i certificats

Els dominis i certificats oscil·len entre 20-50€ depenent de la configuració i nom que volem. Inicialment suposaré 40€ anuals.

## 2.2.3. Altres

### EmailJS

| FREE<br>For Testing                                                                          | PERSONAL<br>For Personal Use                                                                                                        | PROFESSIONAL<br>Entrepreneurs & Freelancers                                                                                                                         | BUSINESS<br>Best For Small Business                                                                                                                                   |
|----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \$0 /mo                                                                                      | \$4 /mo                                                                                                                             | \$12 /mo                                                                                                                                                            | \$40 /mo                                                                                                                                                              |
| 200 monthly requests<br>2 email templates<br>Requests up to 50kb<br>Limited contacts history | 1,000 monthly requests<br>5 email templates<br>Whitelist<br>Unlimited contacts<br>Attachments up to 500kb<br>Completely white label | 5,000 monthly requests<br>Unlimited email templates<br>Whitelist<br>Unlimited contacts<br>Attachments up to 2mb<br>Completely white label<br>Priority email support | 25,000 monthly requests<br>Unlimited email templates<br>Whitelist<br>Unlimited contacts<br>Attachments up to 20mb<br>Completely white label<br>Priority email support |
| SIGN UP FREE                                                                                 | SIGN UP FREE                                                                                                                        | SIGN UP FREE                                                                                                                                                        | SIGN UP FREE                                                                                                                                                          |

*Il·lustració 1 Tarifes de preus d'EmailJS*

Inicialment no crec que hi hagin més de 200 comandes mensuals, i donat que es guarda la informació de les comandes a BD no seria realment necessari.

Tot i això és interessant tenir un mètode alternatiu de notificació per si algun dia hi ha problemes amb l'emmagatzemament de les comandes al núvol, per tant, l'inclourem al pressupost contemplant el pla personal (4\$/mes → 48\$/any ~ 39,5 €/any).

### Pasarel·la de pagament (Stripe)

El primer any no s'implementarà, però es cobra un 1,4% de comissió sobre la venda més una quota fixe de 0,25€ per transacció. (bastant alt, valorarem altres opcions).

## 2.2.4. Cost de desenvolupament del sistema

Hi ha una previsió de 200h en feina de desenvolupament + 40 variables (anàlisi, implantació, formació, etc)

Preu desenvolupament (20€/h)

Total cost desenvolupament: 240 \* 20 = 4.800€

Aquest cost no l'ha de pagar el client (funcionarà per sistema de subscripció), però va bé calcular-lo per saber quant temps i clients necessitem per recuperar la inversió.

## 2.2.5. Total estimació inicial

Tenint en compte els costos dels punts anteriors el total estimat per any serà de: 243,85€

## 2.3. Viabilitat Tecnològica

Per estudiar la viabilitat tecnològica s'han considerat els següents punts:

### Arquitectura dels servidors per allotjar la pàgina web i l'API

- Elecció d'un domini.
- Elecció de l'empresa de hosting i avaluació dels costos.

### Llenguatges de programació que s'utilitzaran

- Des d'un primer moment tinc clar que m'interessa aprendre React, per tant aquest està assegurat.
- Deixant aquest a part, amb quins llenguatges puc avançar més ràpidament?

### Estructures de dades necessàries i interaccions entre serveis

- Quines són les dades que realment necessitem i quines són prescindibles, què volem analitzar, i què podem ignorar.
- Quines interaccions amb l'API hi haurà de forma pública, i quines de forma privada.
- El back-office (.exe) pot connectar-se directament a la base de dades o també ha de passar per l'API?

### Funcionalitats i requeriments del sistema a desenvolupar

- Gestió dels productes.
- Gestió de l'estoc.
- Mètodes de pagament.
- Gestió de les dades sensibles de clients.

### Optimització i bon us del tràfic

Tinc clar que s'utilitzarà molt des de dispositius mòbils, i tot i que a dia d'avui les tarifes d'internet són pràcticament il·limitades, m'interessa no consumir masses recursos del client.

Després d'analitzar aquests punts, podem determinar que tecnològicament és possible, i de fet, adequat per conduir el client cap a un procés de transformació digital.

## 2.4. Viabilitat econòmica

El principal recurs necessari per desenvolupar aquesta plataforma és temps.

S'ha realitzat una relació dels ingressos esperats vs els costos del projecte.

Per fer l'estimació dels ingressos esperats s'han analitzat dades d'influència online del negoci.

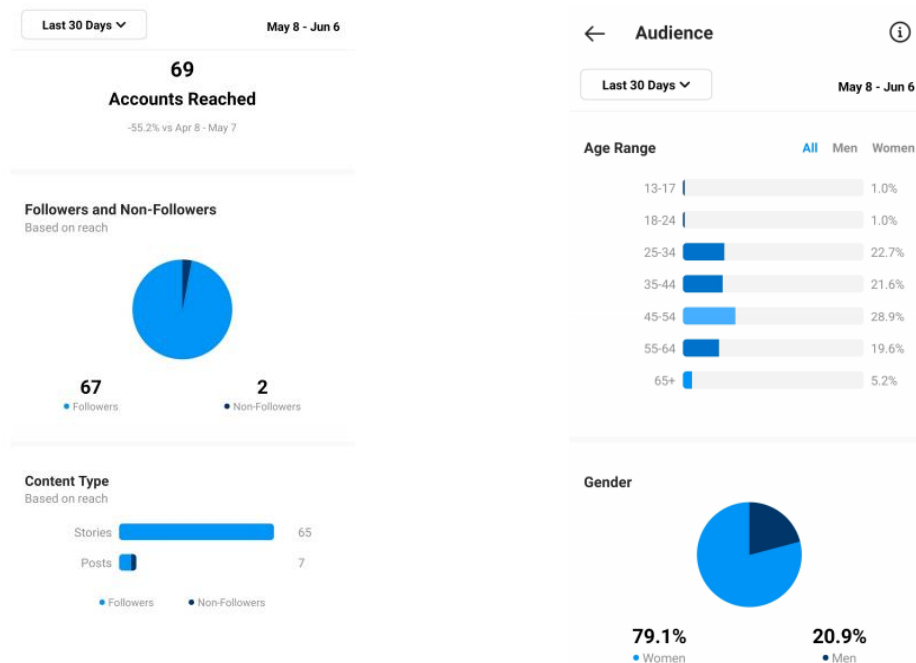
### Google my Business



Il·lustració 2 Anàlisi d'influència online

Moltes visualitzacions són sol·licitant com arribar al negoci (probablement perquè està al centre del poble), però la part interessant són les cerques, de les quals un ~58% són per descobriment i un 42% per cerca directe.

### Instagram



Un tema pendent és fer un seguiment més actiu de les xarxes socials, però amb el poc contingut que hi ha actualment està prou bé arribar a 69 comptes.

## Dades obtingudes del MVP

La mitjana de preu de les comandes que hem rebut a la web actual (MVP) és superior a la mitjana de compra a botiga, i es situa sobre uns 6€. (Tenint en compte que encara no es venen digitalment pastissos ni caixes personalitzades).

## Estimació

No contemplarem la recurrència d'usuaris ja que no tenim dades per analitzar, simplement ens basarem amb la conversió de clients potencials.

Tenim una mitjana de 1000 cerques mensuals a Google MyBusiness + Instagram, suposem una conversió d'un 10% i un 5% d'aquests realitzen dues compres el mateix mes.

$$50 * 6 = 300€$$

$$50 * 12 = 600€$$

Suposem una captació mitjançant màrqueting d'un 10% extra (sobre 1000 usuaris, que equival a uns 5€ amb publicitat d'instagram) amb una conversió d'un 7%. i un 5% realitza dues compres el mateix mes.

$$35 * 6 = 210€$$

$$35 * 12 = 420€$$

$$\text{Total mensual brut} = 300 + 600 + 210 + 420 = 1530€$$

$$\text{Total anual brut} = \sim 1500 * 12 = 18.000€$$

Estimem un increment anual de vendes d'un 5% (degut a millora del servei, venda a domicili, més opcions de compra, fidelització del client recurrent, etc etc)

| Mètrica                                                                    | Any 1                                      | Any 2                                                                             | Any 3                                               | Total 3 anys            |
|----------------------------------------------------------------------------|--------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------|-------------------------|
| Total vendes pel canal online (€)                                          | 18.000€                                    | 18.900€                                                                           | 19.845€                                             | 56.745€                 |
| Costos per comissions en la pasarel·la de pagament                         | 0                                          | ~1.4% del 50% de la facturació = 132,30€                                          | ~ 160€                                              | 1050                    |
| Costos d'implementació de la plataforma tecnològica (feina de programació) | 200h (4000€ fent a 20€/h) (descompte 100%) | 100h (impl. passarel·la pagament + manteniment + millores) 2000€ (descompte 100%) | 50h (manteniment + millores) 1000€ (descompte 100%) | 7.000€ (100% descompte) |
| Costos manteniment plataforma tecnològica (servidors, eines, etc)          | 0                                          | ~250€                                                                             | ~250€                                               | 500€                    |
| Costos de Material (packaging)                                             | 200€                                       | 300€                                                                              | 300€                                                | 800€                    |
| Costos de marketing online                                                 | 100€                                       | 150€                                                                              | 150€                                                | 400€                    |
| Fluxe de caixa                                                             | 17.700€                                    | 18900 - 832,3 = 18.067€                                                           | 19.845 - 860 = 18.985€                              | 54.752€                 |

*Il·lustració 3 Estimació inicial*

## 2.5. Resultats de l'estudi de viabilitat

És evident que si a un producte digital li treus el cost de desenvolupament i potencies el màrqueting és bastant probable que recuperis la inversió (o sent optimista, la milloris).

Tot i que segurament hi hauran altres costos relacionats que desconec, aquesta primera estimació em fa veure que el projecte pot ser bastant rentable.

Tenint en compte que m'he basat només en les compres a la pàgina MVP actual (on només hi han llaminadures),

Al haver-me basat en el preu mig de les compres de la pàgina actual (MVP), he deixat de banda el potencial d'incloure-hi altres productes, com pastissos de llaminadures o caixes boniques amb llaminadures i missatges personalitzats per regalar.

Tampoc he estimat quina seria la repercussió d'aplicar campanyes de màrqueting en dates festives com Nadal o Pasqua (que és quan més es factura).

Això em porta a pensar que rebut mitjà (i per tant la facturació) pot augmentar bastant més del que s'ha vist en la taula anterior.

També hem de tenir en compte que segurament hi hauran altres costos relacionats. Però aparentment la facturació respecte la inversió i costos recurrents (ROI) és molt interessant.

# 3. Metodologia

## 3.1. Metodologies àgils

Es basen en la metodologia incremental, en la que cada cicle de desenvolupament aporta noves funcionalitats a l'aplicació final. Tot i això els cicles són molt més curts i ràpids, pel que es van afegint petites funcionalitats enlloc de grans canvis.

Això permet adaptar el software a les necessitats que van apareixent.

Són les més utilitzades degut a la seva alta flexibilitat i agilitat.

Un dels principals beneficis és la comunicació constant amb els stakeholders i el feedback que aquests proporcionen; evitant així comportaments no desitjats en l'aplicació i la burocràcia de metodologies tradicionals.

Les principals metodologies dins aquest grup són:

- **Kanban (La que utilitzaré)**
- Scrum
- Lean
- Programació extrema (XP)

Ja que he escollit aquesta metodologia, a continuació explicaré en més detall els seus pilars principals i com aplicar-la.

### 3.2.2. Mètode Kanban

Kanban és un mètode per aplicar metodologia àgil de forma bastant simple, sobretot quan es treballa en un equip petit.

La paraula japonesa "kanban" significa "tauler visual"

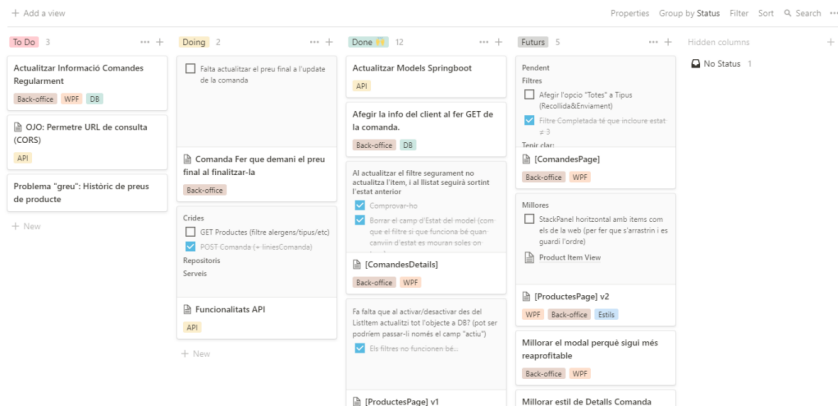
Aquest sistema s'utilitza des dels anys 60, quan Toyota el va definir per limitar les tasques en progrés que tenien paral·lelament. La visió actual del mètode es va definir el 2007 per David J. Anderson.

Els principis i bones pràctiques més rellevants d'aquest mètode són:

- Centrar-se en les necessitats del client.
- Limitar les tasques "en progrés".
- Planificar cicles de revisió amb el client.

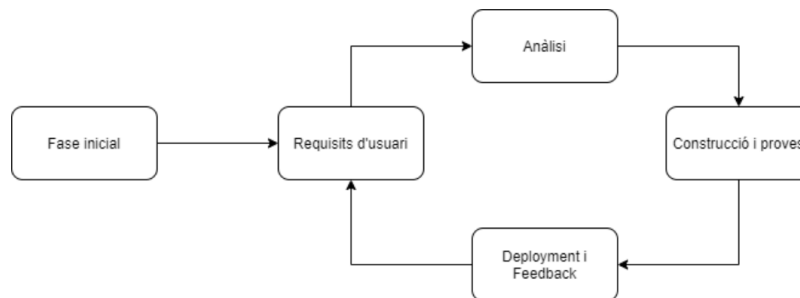
Aporta una visió global de les tasques a realitzar, així com el detall i la informació que es vulgui en cada una d'elles.

L'utilitzo des de fa temps i és genial per augmentar l'eficiència i el focus en les parts importants del projecte, deixant com a tasques futures aquelles que no són tant rellevants a curt termini.



Il·lustració 4 Exemple de taulell de Kanban

El format que utilitzo per estimar les tasques és el següent:



Il·lustració 5 Metodologia d'estimació



### **3.3. Versionat del software**

És el procés continu d'assignació d'un codi o número únic a un producte, per indicar el seu nivell de maduresa.

He utilitzat un sistema de versionat tradicional i simple:

Format numèric amb 3 grups: [major . menor . fix ].

- Major: Assoliment d'un punt estable definit amb el client.
- Menor: Implementació d'una nova funcionalitat estable.
- Fix: Correcció d'errors dins la versió estable menor actual.

Versió actual back-office: 0.8.2

Versió actual front-office: 1.6.0

# 4. Planificació

## 4.1. Com he planificat el projecte

### Fase Inicial

Realització d'un resum a alt nivell de les funcionalitats desitjades a la versió 1.0 del producte. Permetrà realitzar un seguiment dels objectius i tenir en compte una visió global del producte des de l'inici del desenvolupament.

Per cada iteració:

#### Requisits d'usuari

En aquesta fase s'elabora un llistat de requeriments per el mòdul actual i es determina fins on s'haurà d'arribar en aquesta iteració.

#### Anàlisi

S'elabora documentació de tots els requisits del sistema i s'estructura per tal de començar la programació de forma organitzada i calculada.

#### Construcció i proves

Desenvolupament del mòdul i paral·lelament realització de proves i correcció de possibles errors.

#### Deployment i feedback

El sistema passa a pre-producció i s'analitza amb el client (i de ser possible amb un percentatge dels usuaris o consumidors del servei).

Si tot funciona correctament es passa a producció.

## **4.2. Anàlisi funcional de les diferents parts del projecte**

L'anàlisi està detallat en l'apartat 6, aquí farem una breu introducció al resultat final que es vol obtenir sense entrar en detalls tècnics.

### **4.2.1. Back-office**

Aquesta part del projecte ha de permetre al botiguer gestionar les diferents parts la botiga online de forma senzilla.

1. Productes
2. Comandes
3. Visualització de KPIs (Indicadors clau de rendiment)

### **4.2.2. Front-office (web)**

Aquesta part del projecte fa referència al què veurà l'usuari final (comprador), i ha de permetre realitzar tot un cicle normal de compra; comparable al d'altres comerços electrònics existents,

1. Selecció de productes (que s'emmagatzemen en una cistella)
2. Revisió i modificació d'aquests productes o quantitats des de la cistella.
3. Procés de compra
  1. Enviament (amb pagament online)
  2. Recollida a botiga (pagament in-situ)

Aquest procés de compra ha de complir amb les normes vigents de protecció de dades.

Un punt que ens agradaria implementar és un bon compliment de les regles d'accessibilitat per tal que qualsevol usuari pugui utilitzar aquest lloc web.

## **4.3. Desglossament en tasques**

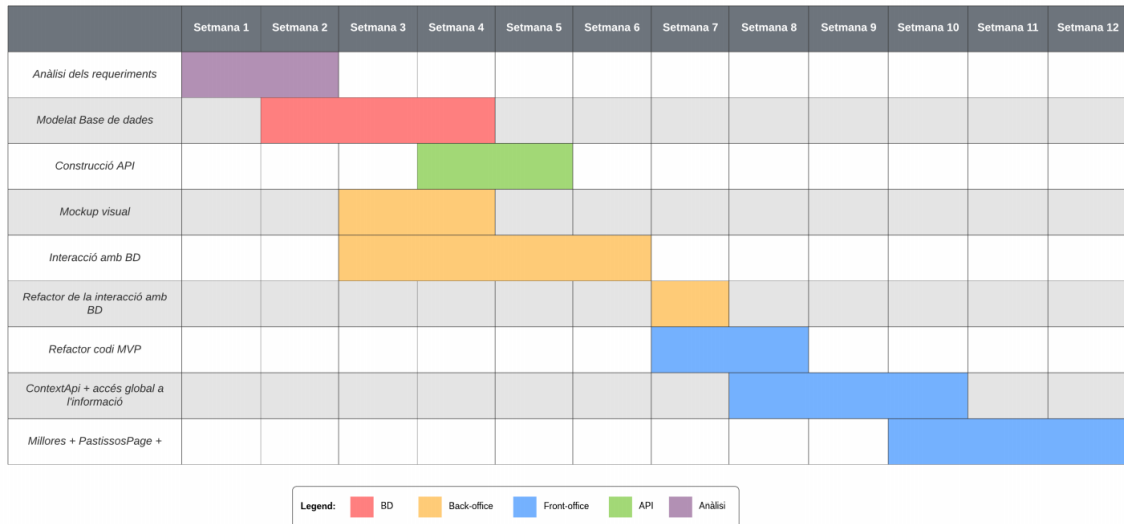
En cada iteració he anat desglossant les tasques obtingudes durant la fase inicial del projecte, i arribant a un acord de les necessitats actuals i reals en cada punt amb el client.

## 4.4. Estimació del temps

1.0 [MVP] — Maig 2020 - Octubre 2020

2.0 [Release]— Març 2021 - Juliol 2021

Diagrama d'estimació de la duració de tasques a realitzar durant el projecte de la versió 2.0.



Il·lustració 6 Diagrama de Gantt

## 5. Marc de treball i conceptes previs

En aquest apartat s'intentarà donar el context necessari al lector perquè pugui entendre el sector i les decisions preses durant el desenvolupament.

### 5.1. Estudis

Per tal d'avaluar la situació dels comerços i la necessitat de digitalització del sector, vaig buscar i comparar estudis de mercat, que analitzen l'estat del petit comerç i el comerç digital després de "l'estabilització" del covid.

A continuació veurem dues parts d'un estudi de l'empresa Elogia, especialitzada en digital commerce marketing.

La primera analitza els comportaments del consumidor, i la segona amb informació de negocis.

"Elogia.net" (Juliol, 2020). *Estudio Covid19 Consumo Online & eCommerce*.  
<https://elogia.net/estudio-covid19-consumo-online-e-commerce/>.

### Metodología

iab españa  
#IABeCommerce



Los datos han sido ponderados para representar la distribución de la población española según datos del ONTSI.

ELABORADO POR:



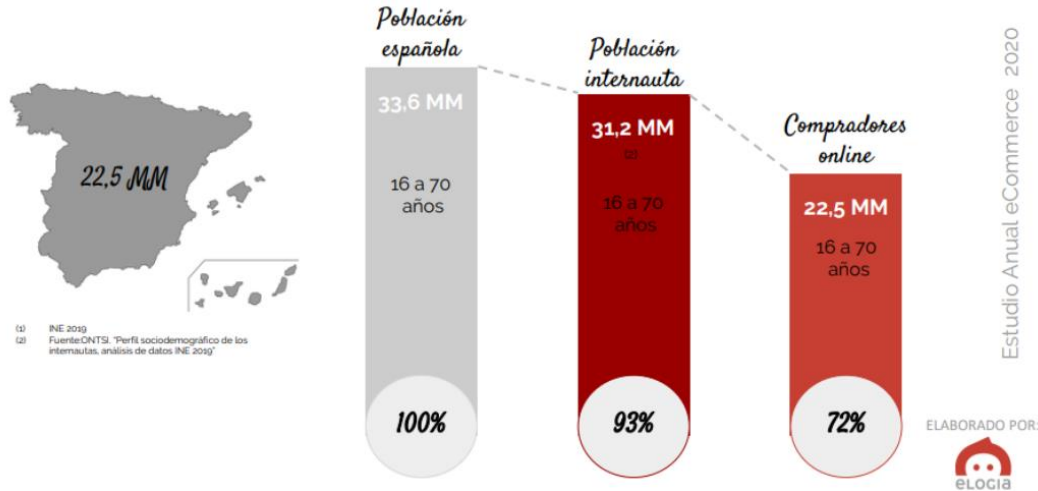
Estudio Anual eCommerce 2020

## 5.1.1. Estudi del consumidor

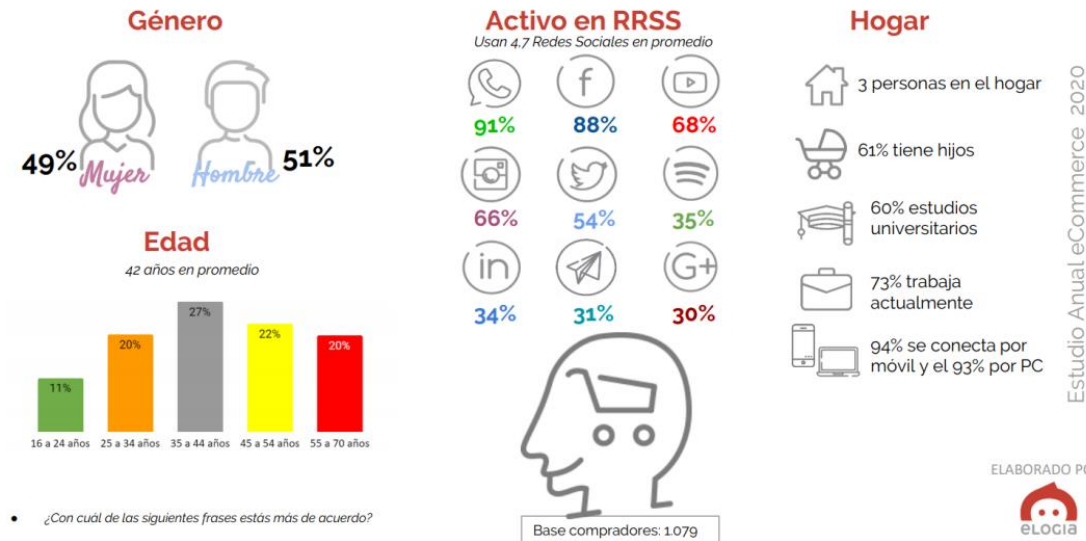
### Dimensionamiento del mercado | Penetración actual



Del total de población internauta española de 16 a 70 años, un 72% ya usa internet como canal de compra.



### Dimensionamiento del mercado | Perfil

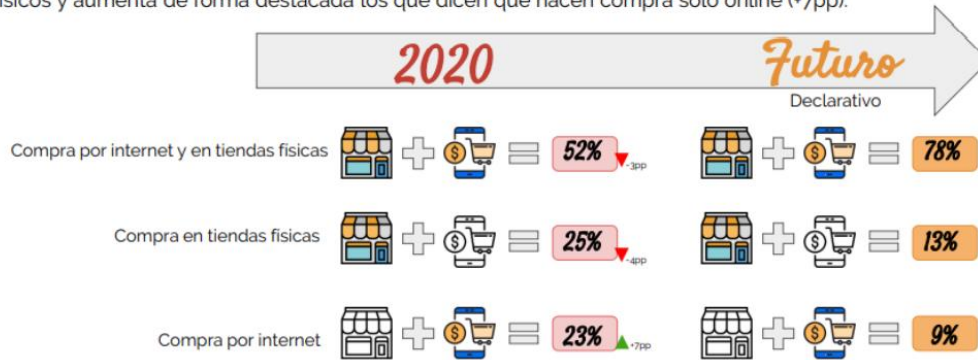


# Dimensionamiento del mercado | Online vs. Offline |



#IABeCommerce

Este año los resultados han sido diferentes respecto a otras ediciones, probablemente influido por el contexto de Covid19. Disminuye ligeramente la proporción de gente que declara combinar on+off, aunque esto no impide que de cara al futuro se siga viendo como "la opción". Disminuye el porcentaje de exclusivos físicos y aumenta de forma destacada los que dicen que hacen compra sólo online (+7pp).



Estudio Anual eCommerce 2020

ELABORADO POR:



• ¿Con cuál de las siguientes frases estás más de acuerdo?

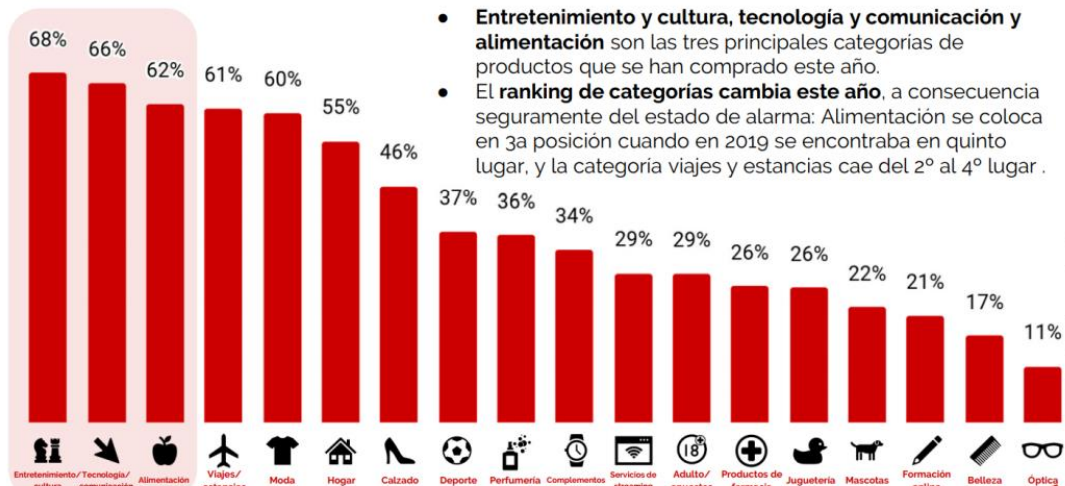
Base compradores: 1.079

▲ dif. significativas vs 2019

# Uso y hábitos de compra | Tipos de productos y servicios | Total categorías



#IABeCommerce



- **Entretenimiento y cultura, tecnología y comunicación y alimentación** son las tres principales categorías de productos que se han comprado este año.
- El **ranking de categorías cambia este año**, a consecuencia seguramente del estado de alarma: Alimentación se coloca en 3ª posición cuando en 2019 se encontraba en quinta lugar, y la categoría viajes y estancias cae del 2º al 4º lugar.

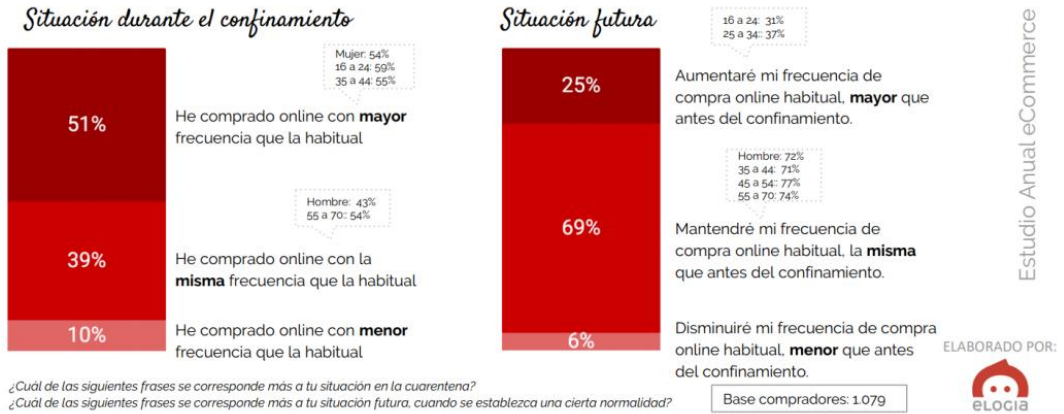
Estudio Anual eCommerce 2020

• ¿Qué tipo de productos compras o contratas online?

Base compradores: 1.079



- Una de cada dos personas entrevistadas declara que durante el confinamiento ha aumentado la frecuencia de compra online.
- La mayoría de compradores declaran que cuando vuelva cierta normalidad volverán a su frecuencia de compra online habitual; sin embargo, un 25% declara que seguirá con una frecuencia mayor.



## Indicadors clau de l'estudi de consum

Un 70% de la població entrevistada compren per internet.

El comprador potencial més abundant és un home entre 30 i 40 anys que combina la compra online amb l'offline.

La compra d'alimentació ha tingut una forta pujada de vendes pel canal online el darrer any.

Un 25% de la gent que ha comprat més per internet durant el confinament, indica que ho seguirà fent per aquest canal.

L'smartphone ja supera l'ordinador de sobretaula per realitzar compres online.



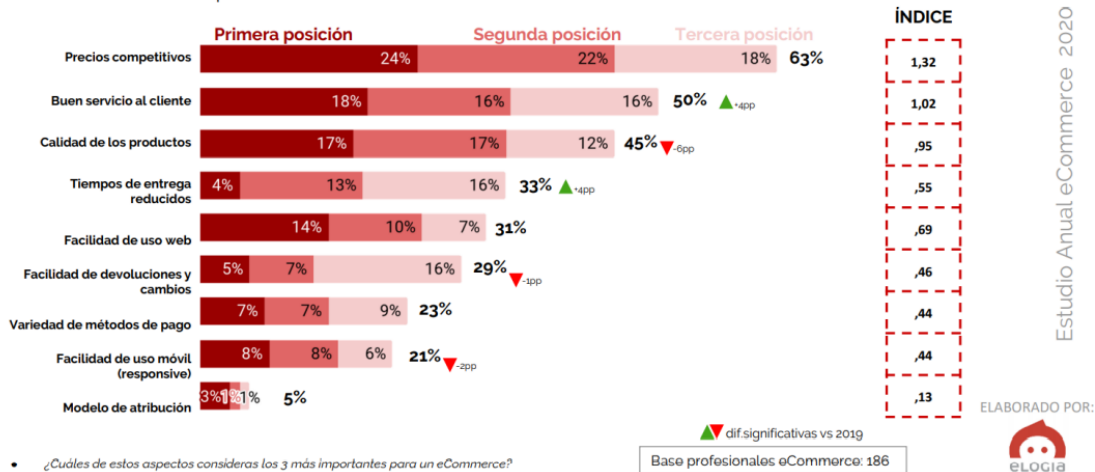
## 5.1.2. Estudi dels comerços electrònics

### Los expertos | Aspectos más relevantes de un eCommerce



#IABeCommerce

- Los profesionales consideran que los precios competitivos son los aspectos más relevantes de un eCommerce.
- Ofrecer un buen servicio al cliente y productos de calidad son los siguientes atributos, aunque cabe destacar que este año cambia la posición de estos dos atributos.

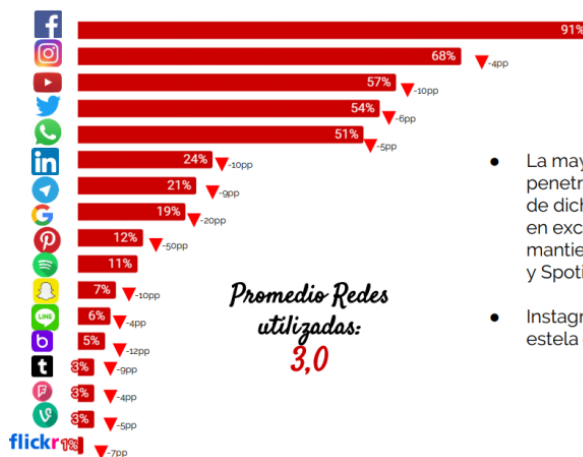


- ¿Cuáles de estos aspectos consideras los 3 más importantes para un eCommerce?

### Los expertos | Uso Redes Sociales a nivel comercial



#IABeCommerce



- La mayoría de plataformas bajan su penetración de compra online (a través de dichas redes) redes este año 2020, en excepción de Facebook que se mantiene en cabeza sin ningún cambio y Spotify.
- Instagram, Youtube y Twitter siguen la estela de Facebook.

- ¿Qué redes sociales usa tu eCommerce a nivel comercial?

ELABORADO POR: ELOGIA

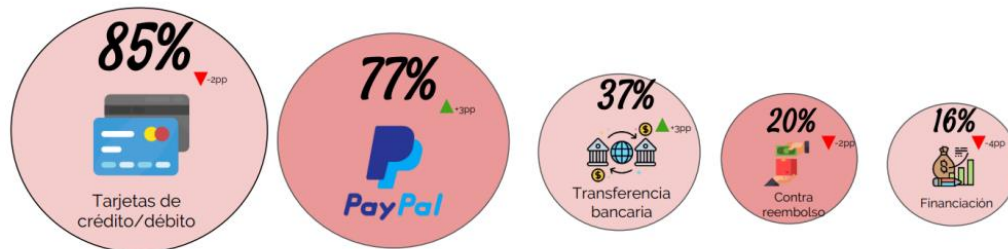
Base profesionales eCommerce: 186

▲ dif. significativas vs 2019

Estudio Anual eCommerce 2020

Estudio Anual eCommerce 2020

Los métodos más usados en los eCommerce son las tarjetas de crédito/débito, aunque disminuye respecto a 2019, seguido de PayPal y transferencia bancaria, que aumentan vs. 2019



Estudio Anual eCommerce 2020

- De los siguientes métodos de pago, ¿cuáles son los más usados en tu eCommerce o en el eCommerce con el que trabajas?

▲ dif. significativas vs 2019  
Base profesionales eCommerce: 186

ELABORADO POR:  
eLOGIA

## Indicadors clau de l'estudi de comerç

El més valorat pels consumidors en una transacció digital és el preu, seguit de l'atenció al client i la qualitat del producte.

Les xarxes més utilitzades per ús comercial són Facebook i Instagram.

El mètode de pagament preferit és targeta de crèdit, tot i que paypal escurça distàncies. El pagament contraremborsament s'utilitza només en un 20% de les vendes.

## 5.2. Informàtica en el núvol

El concepte de la informàtica en núvol va començar en proveïdors de serveis d'internet a gran escala com Google, Amazon... i d'altres que van construir la seva pròpia infraestructura.

Segueix una arquitectura de recursos distribuïts horitzontalment, introduïts com a serveis virtuals escalats massivament.

En general, els clients de la informàtica en núvol no són propietaris de la infraestructura física, sinó que paguen pel lloguer d'ús d'un proveïdor extern. Consumeixen recursos com a servei i paguen només pels recursos que utilitzen.

### 5.2.1. Allotjament web

L'allotjament web (en anglès web hosting) és el servei que proveeix als usuaris d'Internet un sistema per poder emmagatzemar informació, imatges, vídeo, o qualsevol contingut accessible via web.

## 5.3. Stack tecnològic

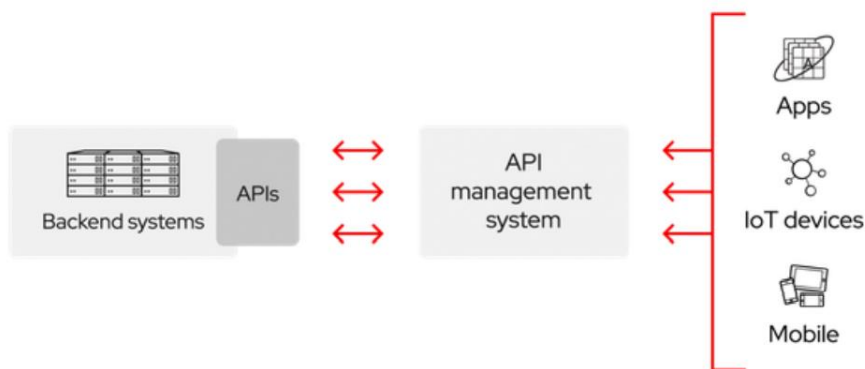
El terme "Stack tecnològic" defineix els llenguatges de programació, frameworks i eines que s'utilitzen en un projecte.

*Exemple: L'stack tecnològic de Facebook és JavaScript, HTML, CSS, PHP i ReactJS.*

## 5.4. API

Una API (Interfície de programació d'aplicacions) és un conjunt de definicions i protocols que s'utilitza per integrar el software d'aplicacions.

Permeten que diferents productes i serveis es comuniquin entre ells via peticions i respostes sense necessitat de saber com estan implementats (abstracció).



*Il·lustració 7 Arquitectura API*

Les APIs permeten restringir l'ús i l'obtenció d'aquestes dades, de forma que només hi tinguin accés usuaris autoritzats.

### 5.4.1. Classificació segons accés

#### Privades

Les APIs només es poden utilitzar internament, així que l'empresa té un control total.

#### De partners

Les APIs es comparteixen amb partners empresarials.

Un exemple d'aquest tipus d'APIs és el que implementa Amadeus IT Group, S.A., empresa espanyola que proveeix informació de disponibilitat hotelera a aerolínies per fer vendes creuades. (entre altres funcionalitats).

## **Publiques**

Tothom té accés a les API, així que altres empreses o particulars poden desenvolupar aplicacions que interactuïn amb l'API.

Per exemple, la web <https://covidtracking.com/> té disponible una API pública perquè tothom pugui desenvolupar aplicacions que informin de l'estat de la pandèmia a la seva regió (EEUU).

### **5.4.2. Classificació segons protocol**

#### **REST (transferència d'estat representacional)**

És un tipus d'arquitectura de desenvolupament web que es basa totalment en l'estàndard HTTP/s.

No guarda l'estat entre peticions.

Utilitza diversos verbs per indicar l'acció que es vol realitzar:

- GET: Per consultar recursos.
- POST: Per crear recursos.
- PUT: Per editar recursos.
- DELETE: Per eliminar recursos.
- PATCH: Per editar certes parts d'un recurs.

Rest és molt flexible i ens permet transmetre pràcticament qualsevol tipus de format.

S'ha d'indicar al Header Content-Type de la crida HTTP/s.

#### **RPC (Remote Procedure call)**

És un protocol que exposa mètodes per manipular dades a través d'HTTP/s.

Els endpoints poden contenir verbs de l'operació que realitzen. (REST no pot, per exemple)

Només utilitza GET y POST:

— GET per obtenir informació.

— POST per tota la resta.

El tipus de dades que utilitza són:

— XML (XML-RPC) o JSON (JSON-RPC).

#### **SOAP (Simple Object Access Protocol)**

És un protocol que només accepta XMLs .

És agnòstic al protocol de transport, i pot enviar dades utilitzant FTP, POP3, TCP, SMTP, etc.

## 5.5. Programació asíncrona

Des del moment que interactuem amb un servidor, les dades no ens arriben immediatament. Això pot causar problemes a nivell de client si no es marca aquesta opció en espera (bloquejaria la interfície d'usuari).

La programació asíncrona ens dona la capacitat de "pausar" l'execució d'una funció a l'espera que es completi una operació o es rebin certes dades. Així evitem bloquejar l'execució fins que s'hagi completat aquesta tasca.

S'utilitza un operador anomenat "await", que permet continuar l'execució i realitzar certes accions quan aquesta crida que esperàvem es completa.

## 5.6. Arquitectura MVVM

És un patró d'arquitectura de software presentat per Microsoft l'any 2005.

MVVM equival a model-vista-"model de vista" (model-view-viewmodel, en anglès).

Es caracteritza per intentar desacoblar al màxim la interfície d'usuari de la lògica de l'aplicació.

### Model

Representa la capa de dades. El modelo conté la informació, però mai les accions o serveis que la manipulen. Tampoc té dependències amb la vista.

### Vista

La intenció de la vista és representar la informació a partir dels elements visuals que es construeixen. Les vistes al patró MVVM son reactives, contenen bindings (enllaços) amb les dades que es mostren (per actualitzar-se automàticament quan aquestes canvien), i events que permeten llançar funcions que implementen la lògica.

### Model de vista

El model de vista és un actor intermediari entre els dos anteriors, conté la lògica de la presentació i es comporta com una abstracció de la vista. La comunicació entre la vista i el viewmodel es realitza a partir dels bindings comentats a l'apartat de la vista.

Aquest patró l'utilitzaré en certes parts del back-office per facilitar l'actualització de la informació entre components.

## 5.7. DNS (Domain Name System)

És una tecnologia basada en una base de dades que serveix per resoldre noms a la xarxa, és a dir, per a conèixer la direcció IP de la màquina on està allotjat el servei al que volem accedir.

## 5.8. Comerç electrònic

És un mètode de compravenda de bens, productes o serveis que utilitza internet com a mitjà.

### Tipus de comerç electrònic

Es diferencien per la naturalesa de les transaccions.

- B2B (Business-to-Business): Empreses que proveeixen a altres empreses.
- B2C (Business-to-Consumer): Empreses que venen a consumidors finals.
- B2G (Business-to-Government): Empreses que proveeixen amb institucions públiques.
- C2C (Consumer-to-Consumer): Comerç entre particulars, on la plataforma s'endú una comissió.
- C2B (Consumer-to-Business): Consumidors que venen a negocis, popular en productes de segona mà.

## 5.9. Altres conceptes relacionats amb el projecte

### Back-office

El back office engloba totes les activitats relacionades amb la gestió interna de l'empresa.

En el cas d'un software fa referència a aquelles àrees de gestió i configuració del negoci, on el consumidor o usuari final no pot accedir

### Front-office

És la part del software amb la que els usuaris o clients finals poden interactuar.

### Politica CORS (Cross-Origin Resource Sharing)

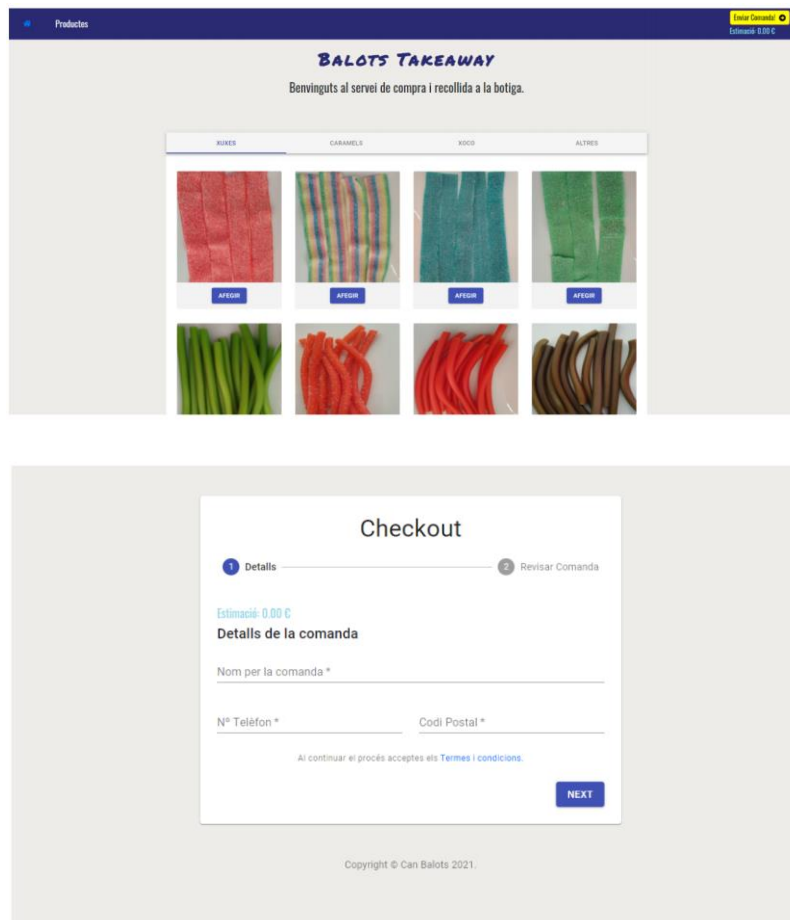
És un mecanisme o política de seguretat que permet controlar les peticions HTTP asíncrones que es poden realitzar des d'un navegador a un servidor amb domini diferent.

## 5.10. MVP

MVP, que prové de l'anglès "Minimum Viable Product", és un producte amb les característiques mínimes que permet validar el model de negoci i la retenció/conversió d'usuaris abans d'invertir-hi moltes hores.

És una bona tàctica, ja que analitzar l'ús d'un MVP és menys costós que desenvolupar un producte amb més funcions que incrementen els costos i els riscos si el producte falla.

### Web actual en producció (MVP)



Format del correu amb informació de la comanda.

Nom: test preu rounded#1751.  
Client: 666444666.  
Data Comanda 22-01-2021 13:14PM  
Preu estimat : 0.66

| ID | Nom                 | Categoria | Quantitat |
|----|---------------------|-----------|-----------|
| 1  | lengua fresa        | Xuxes     | 1         |
| 2  | lengua colores      | Xuxes     | 1         |
| 6  | dulcipica fresa     | Xuxes     | 1         |
| 10 | jelly torcida fresa | Xuxes     | 1         |
| 14 | ositos rojos        | Xuxes     | 3         |

El fet de tenir els productes incrustats dins la web fa que qualsevol canvi (de preu, de disponibilitat, etc) impliqui modificar el codi, compilar i fer el procés de deployment.

# 6. Requisits del sistema

## 6.1. Requisits Generals

- Emmagatzemar i consumir la informació des d'una base de dades.
- Utilitzar AWS S3 per emmagatzemar les imatges dels productes.
- Un cop validat que tot funciona correctament, migrar el projecte al cloud (AWS), substituint el MVP actual per una versió plenament funcional de la web.

Hi hauran dos actors, l'usuari (client) i l'administrador (botiguer).

## 6.2. Requisits Back-office

### Funcionals

Visualitzar els productes i gestionar-los:

- Afegir.
- Editar
- Activar / Desactivar.

Visualitzar les comandes i gestionar-les:

- Veure'n el detall (llista de productes)
- Canviar-les d'estat.
- Contactar amb el client directament mitjançant Whatsapp Desktop.

### No funcionals

- Visualitzar les comandes dins un llistat
- Visualitzar els productes dins un llistat
- Visualitzar els KPIs del negoci de forma ràpida i senzilla.

Com que no es treballarà simultàniament amb aquest programa, reduir al màxim la comunicació amb la base de dades.

- Productes: Rebre'ls un cop i anar actualitzant a mesura que es fan canvis.
- Comandes: Consultar periòdicament per si hi han noves comandes a la BD.



## 6.3. Requisites front-office

### Funcionals

- Visualitzar productes segons categoria / subcategoria.
- Afegir productes a la cistella.
- Extreure productes de la cistella.
- Visualitzar la cistella.
- El procés d'encàrrec ha de permetre indicar que l'usuari és al·lèrgic, i per tant la manipulació del producte s'ha de realitzar de forma segura pel consumidor.

### No funcionals

- Adaptació de les vistes perquè pugui utilitzar-se des de web/mòbil
- Seguir les pautes sobre accessibilitat apreses a l'assignatura de Legislació.
- Garantir el compliment de la RGPD.
- Adaptació dels components a la mida de lletra del navegador.
- No s'emmagatzemar dades del client amb cookies.
- Seguir un estil unificat.
- Visualitzar els al·lèrgens de cada producte d'una forma senzilla.

# 7. Estudis i decisions

En aquest apartat presentaré el stack tecnològic que utilitzo i explicaré perquè l'he escollit.

També llistaré les eines utilitzades.

## 7.1. Llenguatges de programació i llibreries

### 7.1.1. Base de Dades

A partir del model que tinc vaig decidir utilitzar PostgreSQL, programari lliure que implementa un sistema de gestió de bases de dades relacional.



### 7.1.2. Front Office

Per prendre decisions en aquesta part del projecte he utilitzat com a referència les següents webs:

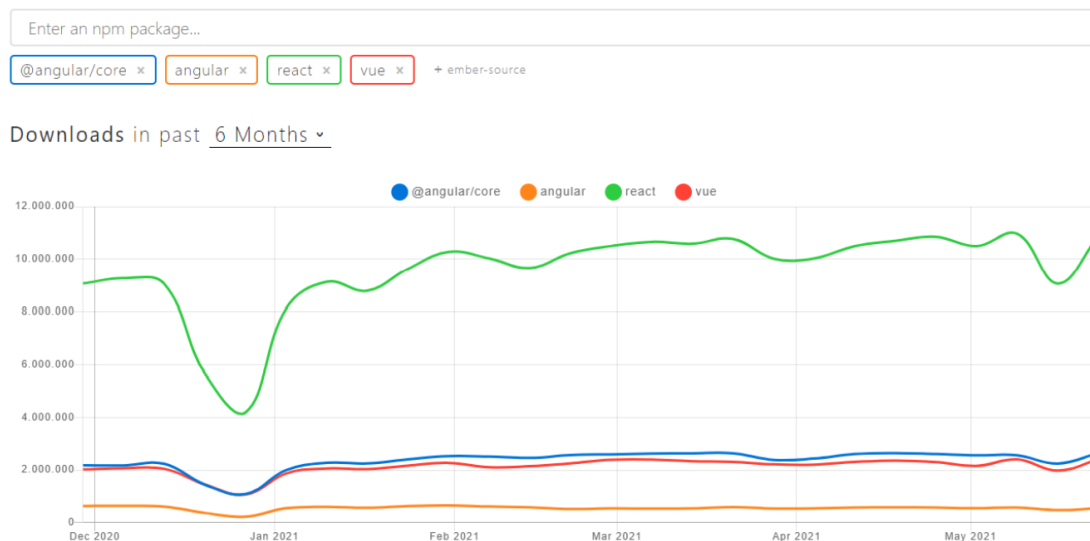
<https://www.npmtrends.com>

<https://stateofjs.com>

He utilitzat HTML/CSS/Javascript amb React; un Framework de codi obert de JavaScript que s'utilitza per a desenvolupar interfícies d'usuari. He escollit aquest Framework perquè s'ajustava a la necessitat, tenia ganes d'aprendre a utilitzar-ne algun, i React és el més usat i que més ofertes de feina té.

 npm trends

@angular/core vs angular vs react vs vue



*Il·lustració 8 Anàlisi rellevant sobre els llenguatges de frontend*

També he optat per utilitzar un pre-compilador de CSS anomenat SASS. Proporciona senzillesa, sintaxi més elegant i implementa diverses característiques molt útils per el reaprofitament de codi.



React és un llenguatge reactiu, s'encarrega d'actualitzar i renderitzar la informació cada cop que es modifica.

Es basa en components, objectes encapsulats que gestionen el seu propi estat (i es comuniquen amb altres).

*Introducció a React.* React – A JavaScript library for building user interfaces. (n.d.). <https://reactjs.org/>.

### 7.1.3. Back Office

La idea inicial era fer-ho tot web, però degut a la necessitat de seguretat i al temps que suposaven certs requeriments vaig decidir que la part de back-office sigués un programa independent per Windows, que només tindrà la persona responsable del negoci.

Per programar-ho he utilitzat WPF (Windows Presentation Foundation). Inicialment utilitzava Entity Framework, però degut a certs problemes amb la comunicació a base de dades, he acabat optant per utilitzar connectar-me a la base de dades mitjançant Npgsql (un Data Provider open source per ADO.NET i PostgreSQL).



### 7.1.4. API

Vaig decidir utilitzar Spring per programar l'API ja que l'havíem treballat a classe.

Tot i això, en futures versions migraré aquest projecte a Node.js amb Express per tal d'utilitzar majoritàriament el mateix stack dins el projecte.



### 7.1.5. Altres

#### *7.1.5.1. Docker*

Docker és un projecte de codi obert que automatitza el desplegament d'aplicacions dins de contenidors de programari, proporcionant així una capa addicional d'abstracció i automatització de virtualització d'aplicacions en diferents sistemes operatius.



### 7.1.5.2. Amazon Web Services (AWS)

És la plataforma al núvol més adoptada i complerta del mercat, ofereix més de 200 serveis al núvol amb centres de dades a nivell global.



Il·lustració 9 Alguns dels serveis oferts per Amazon Web Services

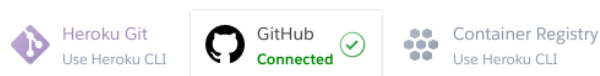
Llista d'algunes de les categories de serveis que ofereix Amazon.

### 7.1.5.3. Heroku

La plataforma Heroku, propietat de Salesforce, és una plataforma com a servei (Paas) de computació al núvol que suporta diferents llenguatges de programació.

Permet, d'una forma realment senzilla, desplegar qualsevol producte i intercomunicar-lo amb altres.

Desplegar és un procés senzill on simplement connectant el repositori a Github amb Heroku, ja pots disparar el deployment o configurar-ho perquè ho faci automàticament quan es detecta un canvi.



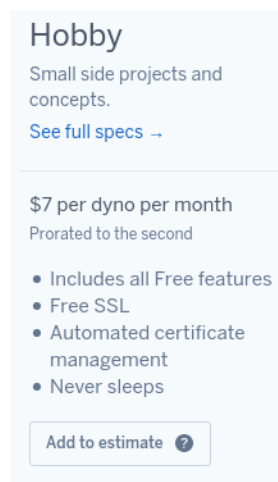
És més car que Amazon WS, però la senzillesa que ofereix és genial en fases inicials del projecte. El preu depèn dels serveis que vulguis connectar, les funcionalitats que vulguis tenir disponibles, el tràfic, etc.

Per aquest projecte aproximadament seran 60\$ al mes.

Tot i això es pot utilitzar una capa gratuïta, ideal per validar el negoci abans d'invertir-hi diners.

L'únic problema amb aquesta capa és que entra en repòs automàticament quan passen 15-20 minuts sense cap visita. Això fa que la primera visita a la web (quan està reposant) sigui molt lenta, oferint una mala experiència d'usuari.

La solució és pagar, com a mínim el pla "Hobby", amb el qual les aplicacions connectades mai dormen.



**Hobby**  
Small side projects and concepts.  
[See full specs →](#)

---

**\$7 per dyno per month**  
Prorated to the second

- Includes all Free features
- Free SSL
- Automated certificate management
- Never sleeps

[Add to estimate](#) ⓘ

Els "dynos" són contenidors independents intercomunicats (molt semblant a Docker).

## 7.2. Eines de treball

### 7.2.1. Generals

#### 7.2.1.1. Github

És un servei de hosting de repositoris Git, el qual ofereix tota la funcionalitat de Git de control de revisió distribuït i administració de codi de la font així com afegint les seves característiques pròpies.



#### 7.2.1.2. Notion

Notion és una plataforma de col·laboració amb suport per markdown que integra taulells kanban, tasques, wikis, i bases de dades. L'empresa diu ser un espai de treball tot-en-un per fer notes, coneixement i administració de dades, administració de projectes i tasques.



#### 7.2.1.3. Postman

Postman és una eina que ens permet fer peticions sobre APIs de forma molt simple per tal de provar-les ràpidament.



## 7.2.2. Front Office

### 7.2.2.1. Visual Studio Code

Visual Studio Code és un editor de codi font desenvolupat per Microsoft per a Windows, Linux i macOS. Inclou suport per a la depuració, control integrat de Git, ressaltat de sintaxi, finalització intel·ligent codi, fragments i refactorització de codi.



### 7.2.2.2. React DevTools (extensió del navegador)

Aquesta extensió permet inspeccionar els components renderitzats a la pàgina web de forma jeràrquica. També permet veure i editar l'estat actual d'aquests components a temps real.



## 7.2.3. Back Office

### 7.2.3.1. Visual Studio 2019

Microsoft Visual Studio és un IDE (entorn integrat de desenvolupament) de Microsoft que permet desenvolupar aplicacions, llocs web, apps web, etc.





## 7.2.4. API i BD

### 7.2.4.1. IntelliJ Idea

IntelliJ IDEA és un entorn de desenvolupament integrat escrit en Java per desenvolupar programari informàtic.



### 7.2.4.2. IntelliJ DataGrip

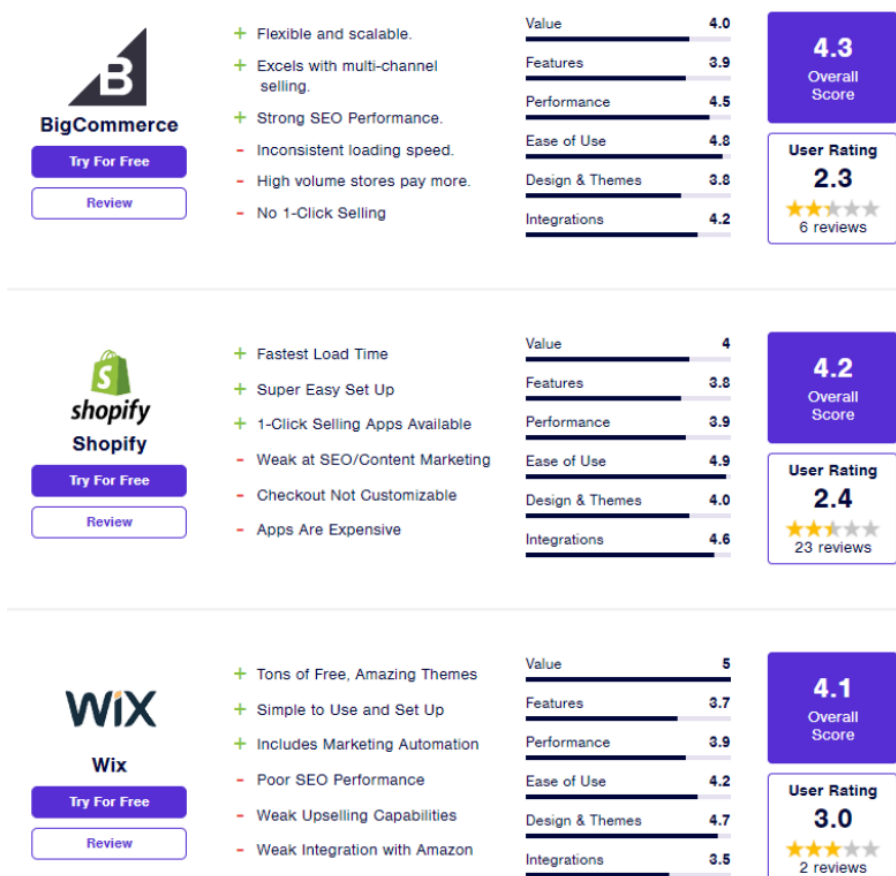
DataGrip és un programa gestor de bases de dades. Pot connectar-se amb bases de dades locals, en un servidor o al núvol. Suporta MySQL, PostgreSQL, Microsoft SQL Server, Oracle, i altres.



## 7.3. Estudi de la competència

Com ja he indicat anteriorment, en una fase inicial del projecte vaig estar buscant plataformes web on integrar la botiga, però no en vaig trobar cap que s'ajustés al que necessitava (segurament buscant més o fent proves amb diferents plataformes hauria pogut configurar-ho).

A nivell de plataforma de comerç electrònic general podem destacar les següents empreses:



Il·lustració 10 Anàlisi de competència

Òbviament aquestes plataformes web estan a un altre nivell, per exemple Shopify va tenir una facturació l'any 2018 de 108,112,569,118 EUR.

Però gran part del meu target no són usuaris coneixedors d'aquestes plataformes, i menys disposats a implementar-les invertint moltes hores pròpies. Per tant, l'alternativa que proposo és una eina molt simple que permeti tenir presència a internet i començar a avaluar si aquest canal els hi funciona.

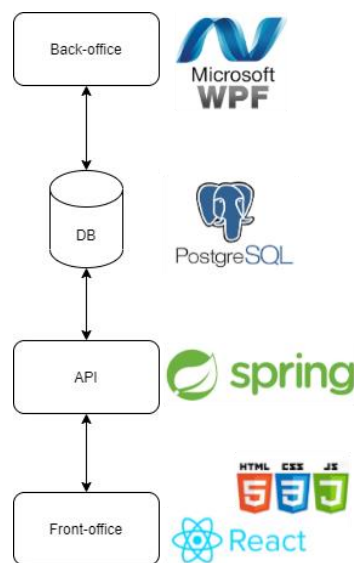
# 8. Anàlisi i disseny del sistema

## 8.1. Anàlisi

### 8.1.1. Estructura general i comunicació

L'aplicació de back-office té accés directe a la base de dades, ja que és un executable que només tindrà el propietari del negoci. (en versions futures s'hauria d'integrar dins la web i fer-lo passar a través de l'API).

El client només té accés a mètodes GET de l'API amb excepció del POST de comanda. (Està previst fer una verificació per captcha abans de publicar la comanda, per evitar atacs).



*Il·lustració 11 Estructura del sistema a dissenyar*

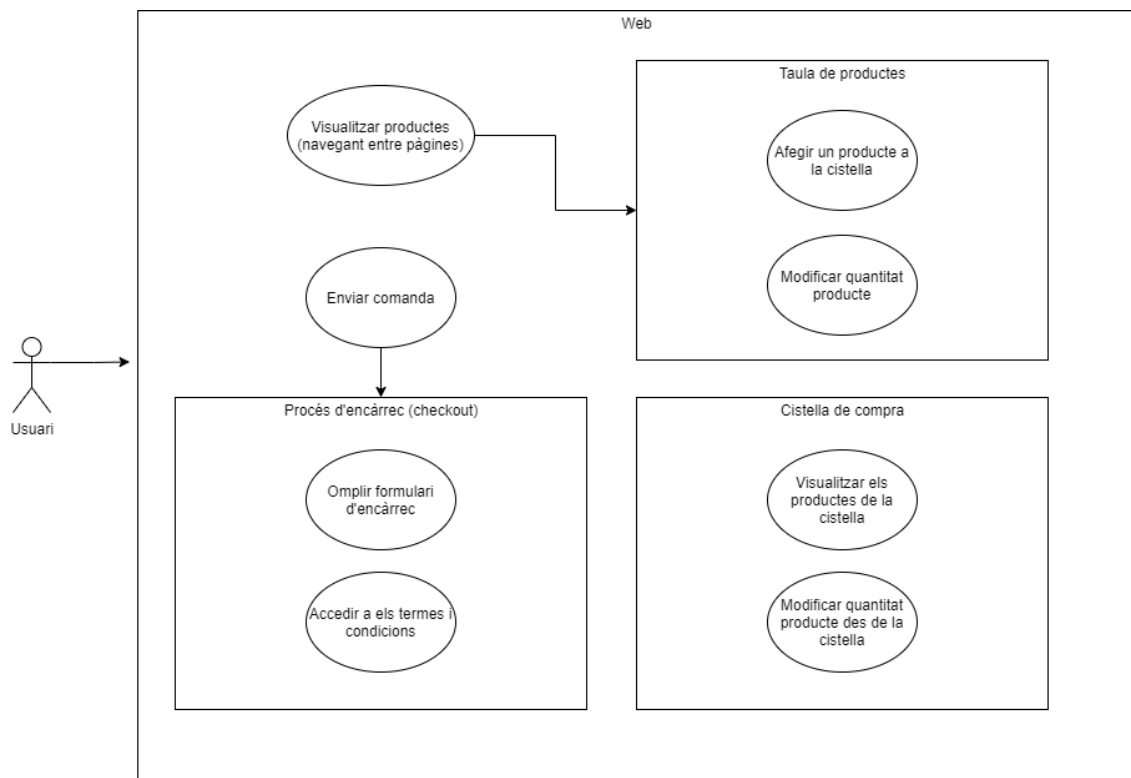
## 8.1.2. Casos d'ús

A continuació mostrarem els casos d'ús dins l'aplicatiu.

En el cas del back-office, considerem "Usuari" al botiguer.

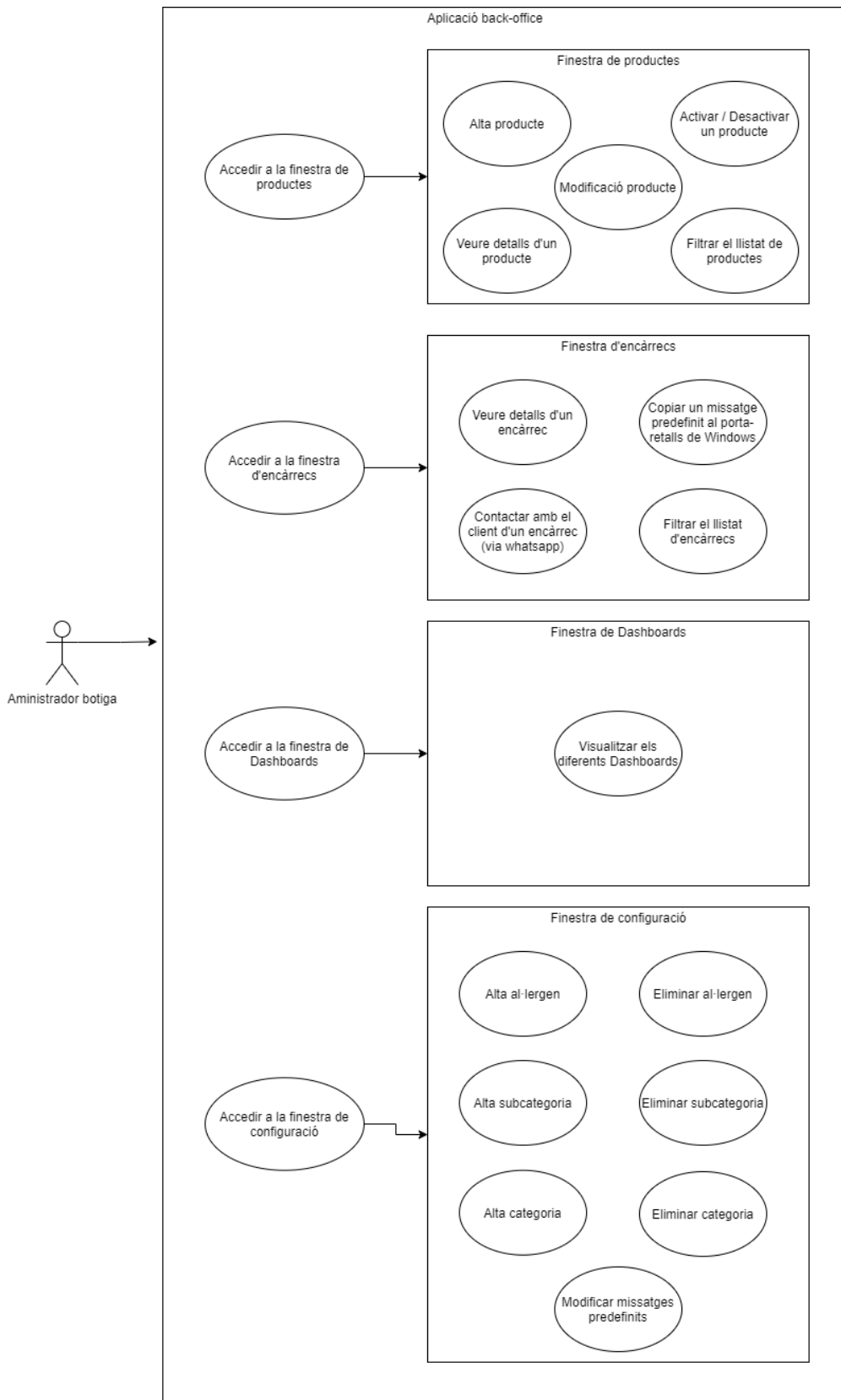
En el cas de front-office, considerem "Usuari" al client final.

### Diagrama casos d'ús front-office



Il·lustració 12 Diagrama casos d'ús front-office

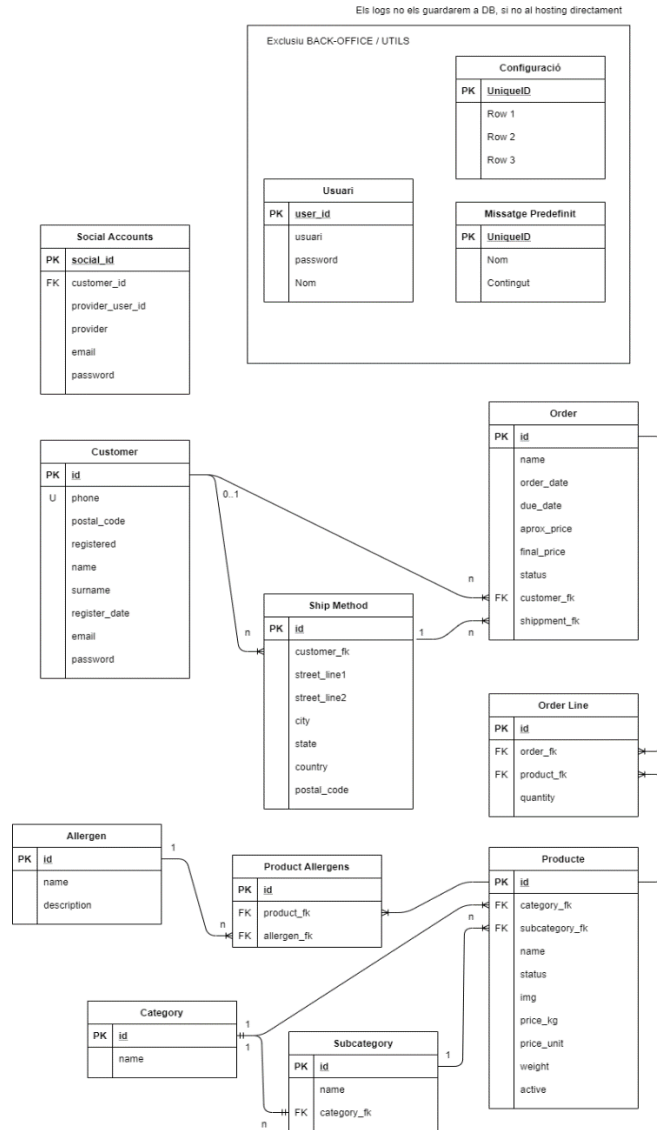
## Diagrama casos d'ús back-office



Il·lustració 13 Diagrama casos d'ús back-office

## 8.1.2. Existència de dades

A continuació mostrarem el model plantejat a l'inici del projecte, conté tota la informació que en un primer moment hem considerat necessària.



A continuació detallem les taules més rellevants:

- Alergen: Conté el nom i una descripció de l'alergen. Es relaciona amb producte utilitzant "productalergens".
- Categoria / Subcategoria: Classificació dels tipus de producte.
- Producte: Conté informació rellevant del producte, i es relaciona LiniesComanda, Categoria / Subcategoria i "productalergens".
- Comanda: Conté informació general de la comanda, així com un una relació amb els productes de la comanda via "LiniesComanda"

### 8.1.3. API

En la primera versió no connectarem el back-office via API, ni guardarem perfils de clients, per tant l'API serà bastant bàsica.

Necessitarem els següents endpoints:

#### **Productes**

[GET] - Get Productes

o

[GET] - Get Productes segons categoria

I com a resposta rebrem un llistat de productes amb la següent informació:

```
[
  {
    "prold": 23,
    "nom": "Producte 23",
    "rutalmg": "url imatge",
    "preuKg": 10.0,
    "preuUn": 0.20,
    "pes": 1.0, // grams
    "descripcio": "descripció del producte",
    "subcategoria": {
      "subld": 1,
      "nom": "Xuxes",
      "subCatld": 1
    },
    "alergens": [
      {
        "aleld": 1,
        "nom": "Nom al·lergen",
        "descripcio": "Descripció de l'al·lergen."
      }
    ]
  }
]
```

## Comandes

[POST] - Post Comanda

Adjuntarem la següent informació a la petició:

```
{
  "nomComanda": "Comanda API",
  "preuAprox": 15,
  "telefon": "777666777",
  "codiPostal": "17003",
  "envNoDetails": false, // enviament gestionat per whatsapp
  "linies": [
    {
      "linProld": 3,
      "quantitat": 12
    },
    {
      "linProld": 5,
      "quantitat": 1
    }
  ]
}
```

I rebrem un booleà indicant si s'ha pogut guardar o no.

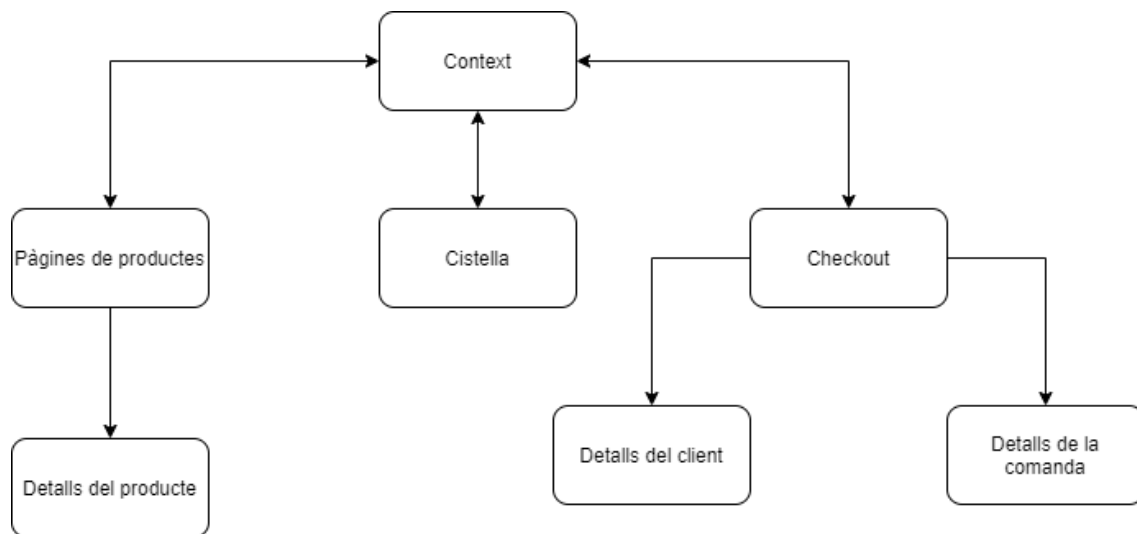


## 8.1.4. Front office

### 8.1.4.1. Gestió de la informació local (estat)

Utilitzem el context com a punt central d'emmagatzemament de dades i variables de la web, des de la qual hi accediran els altres components.

A continuació podem veure com es transmet aquesta informació.



### 8.1.4.3. Fitxes d'ús

En tots els casos l'actor serà l'usuari estàndard (qualsevol persona)

Fitxes d'ús Navegació

Codi: N-01

**Nom:** Navegar entre pàgines

Visió: Usuari

Descripció: Permet canviar de pàgina i veure la informació que conté cadascuna.

Pre: —

Post: L'usuari ha canviat de pàgina

Escenari bàsic:

1. Clic al menú desplegable (part superior esquerra de la pàgina).
2. Clic a l'opció desitjada [Productes, pastissos, etc]

Codi: N-02

**Nom: Accedir a la cistella**

Visió: Usuari

Descripció: Permet obrir el menú de la cistella i visualitzar-ne el contingut.

Pre: —

Post: L'usuari pot visualitzar el menú i interactuar amb el llistat.

Escenari bàsic:

1. Clic a la icona d'una cistella (part superior dreta de la pàgina)
2. Es desplega la vista de cistella i apareix a la part dreta de la pantalla (o ocupant-la tota si s'accedeix des d'un telèfon mòbil).

Codi: N-03

**Nom: Finalitzar encàrrec**

Visió: Usuari

Descripció: Permet a l'usuari finalitzar el procés d'encàrrec.

Pre: —

Post: L'usuari ha enviat l'encàrrec al negoci.

Escenari bàsic:

1. Clic al botó "Enviar comanda" (part superior dreta de la pàgina)
2. Canvia la pàgina (/checkout).
3. L'usuari omple el formulari amb les dades requerides.
4. Clic al botó "Següent"
5. Si (dades vàlides)
  - Canvia el contingut de la pàgina al següent estat del procés.
  - Revisió de la comanda i clic a "Enviar".
  - Comanda realitzada. Canvia el contingut de la pàgina notificant el nom de comanda a l'usuari.
6. Altrament
  - Apareix un missatge d'error a la part inferior de la pantalla, indicant que les dades no són vàlides.

Codi: P-01

**Nom: Afegir producte a la cistella. (taula productes)**

Visió: Usuari

Descripció: S'afegeix un producte a la cistella des de la vista de productes.

Pre: —

Post: El producte s'ha afegit a la cistella.

Escenari bàsic:

1. Si ( aquest producte no existeix a la cistella ):
  - Clic al botó afegir, sota la imatge del producte desitjat
2. Altrament
  - Clic a la icona +, situada a la part dreta de la targeta del producte.

Codi: P-02

**Nom: Treure producte de la cistella (taula productes)**

Visió: Usuari

Descripció: S'extreu una unitat del producte desitjat de la cistella.

Pre: Ha d'haver-hi com a mínim una unitat d'aquest producte a la cistella (P-01).

Post: El producte s'ha extret una unitat del producte. Si ara és 0 s'elimina el producte de la cistella.

Escenari bàsic:

1. Clic a la icona -, situada a la part esquerra de la targeta de producte.

## Fitxes d'ús Cistella

Codi: C-01

**Nom: Modificar quantitat producte des de la cistella.**

Visió: Usuari

Descripció: S'incrementa o redueix la quantitat d'un producte determinat des de la cistella.

Pre: El producte ha d'estar a la cistella (P-01) i aquesta ha d'estar visible (N-02).

Post: S'ha modificat la quantitat del producte des de la cistella.

Escenari bàsic:

1. Clic a la icona + o - depenent de l'acció que es vulgui realitzar.

Codi: C-02

**Nom: Finalitzar encàrrec des de la cistella**

Visió: Usuari

Descripció: A la part inferior de la cistella apareixerà un botó idèntic al de la barra superior, per realitzar el procés de finalització de l'encàrrec.

Pre: La cistella ha d'estar visible (N-02).

Post: Es realitza el procés de finalització d'encàrrec (N-03).

Escenari bàsic:

1. L'usuari fa clic al botó "Enviar comanda", situat a la part inferior de la cistella.

## Altres fitxes d'ús

Codi: A-01

**Nom: Accedir als termes i condicions de la pàgina.**

Visió: Usuari

Descripció: L'usuari accedeix a la pàgina amb la informació legal de l'empresa i el servei.

Pre: L'usuari està a la pàgina de finalització de comanda, /checkout (N-03 fins el punt 2)

Post: L'usuari pot llegir els termes i condicions del servei.

Escenari bàsic:

1. Clic al link de la part inferior del formulari "Termes i condicions"

#### 8.1.4.4. Models de dades (front-office)

Per gestionar el context global de l'aplicació utilitzarem una funció de React anomenada Context

| AppProvider                             |
|-----------------------------------------|
| + AppContext: Context                   |
| + productes: List<Producte>             |
| + cistella: List<ProducteCistella>      |
| + preuEstimat: float                    |
| + navTitol: String                      |
| + useEffect(): void                     |
| + fetchProductes(tipus): List<Producte> |
| + postComanda(comanda): bool/void       |
| + afegirProducte(int): void             |
| + extreureProducte(int): void           |

Els productes tindran la mateixa estructura que hem vist a l'API.

La cistella guardarà la següent informació:

| Cistella                      |
|-------------------------------|
| + producte: Producte          |
| + quantitat: int              |
| + extreureProducte(int): void |

## 8.1.5. Back office

### 8.1.5.2. Fitxes d'ús

#### Fitxes d'ús productes

Codi: P-01

**Nom: Alta producte**

Visió: Usuari

Descripció: Es dona d'alta un nou producte al sistema.

Pre: —

Post: S'ha afegit el producte al sistema.

Escenari bàsic:

1. Clic al botó "Afegir Producte".
2. Omplir el formulari amb la informació corresponent.
3. Clic "Guardar"
4. Si (camps vàlids)
  - Es tenca el modal.
5. Altrament s'indica que hi ha un problema amb l'alta del producte.

Codi: P-02

**Nom: Modificació producte / Veure detalls**

Visió: Usuari

Descripció: Es modifica un producte existent.

Pre: —

Post: S'ha modificat el producte.

Escenari bàsic:

1. Clic al botó "Modificar" al producte corresponent dins la vista d'element de la llista.
2. Modificar les dades del formulari amb la informació corresponent.
3. Clic "Guardar".
4. Si (camps vàlids)
  - Es tenca el modal.
5. Altrament s'indica que hi ha un problema amb l'alta del producte.

Codi: P-03

**Nom: Activar / Desactivar producte**

Visió: Usuari

Descripció: Es dona d'alta un nou producte al sistema.

Pre: Estem a l'opció de filtre corresponent (si volem activar als actius, si volem desactivar als inactius).

Post: S'ha actualitzat l'estat del producte.

Escenari bàsic:

1. Clic al botó "Activar" (o "Desactivar") al producte corresponent dins la vista d'element de la llista.

Codi: P-04

**Nom: Filtrar la llista de productes**

Visió: Usuari

Descripció: Al canviar d'opció de filtrat la vista es refresca directament, mostrant només aquells productes que corresponen amb el filtre.

Pre: —

Post: Productes filtrats en base als filtres actuals.

Escenari bàsic:

1. Clic en qualsevol dels botons de filtre de la part superior de la vista de productes.

Codi: E-01

**Nom: Veure els detalls d'un encàrrec**

Visió: Usuari

Descripció: Permet veure els detalls del client i dels productes seleccionats per aquests en un encàrrec concret.

Pre: —

Post: —

Escenari bàsic:

1. Clic a l'opció del menú superior "Comandes".
2. Clic a "Veure detalls" sobre l'element (Comanda) de la llista que volem visualitzar.

Codi: E-02

**Nom: Modificar l'estat d'un encàrrec**

Visió: Usuari

Descripció: Permet modificar l'estat de la comanda entre [Pendent — En curs — Finalitzada]

Pre: Haver accedit als detalls de la comanda (E-01)

Post: S'actualitza la vista del llistat i la comanda apareix on correspon.

Escenari bàsic:

1. Clic al botó de la part inferior esquerra "Passar a {nom del següent estat}".

Codi: E-03

**Nom: Copiar un missatge predefinit.**

Visió: Usuari

Descripció: Permet copiar al porta-retalls un dels missatges definits des de la pàgina de configuració.

Pre: —

Post: —

Escenari bàsic:

1. Clic a l'opció del menú superior "Comandes".
2. Clic al botó amb el missatge que volem copiar (part superior de la vista de comandes).



Codi: E-04

**Nom: Contctar amb el client d'un encàrrec.**

Visió: Usuari

Descripció: Permet obrir Whatsapp Desktop amb el xat del nº de contacte seleccionat , així com amb el missatge predefinit copiat (si és el cas) escrit i llest per ser enviat.

Pre:

Post:

Escenari bàsic:

1. Clic a la icona + o - depenent de l'acció que es vulgui realitzar.

Codi: E-04

**Nom: Filtrar el llistat d'encàrrecs.**

Visió: Usuari

Descripció: Permet filtrar el llistat d'encàrrecs segons les opcions que desitgi l'usuari.

Pre: —

Post: —

Escenari bàsic:

1. Clic a l'opció del menú superior "Comandes".
2. Clic a "Contactar" sobre l'element (Comanda) de la llista que volem contactar.

Fitxes d'ús configuració

Codi: C-01

**Nom: Alta al·lergen, categoria o subcategoria**

Visió: Usuari

Descripció: S'afegeix al sistema un nou al·lergen, categoria o subcategoria.

Pre: —

Post: Element afegit al sistema.

Escenari bàsic:

1. Clic a la icona de configuració per obrir la pàgina.
2. Clic a "Afegir" dins la secció corresponent.
3. Omplir formulari de la nova finestra emergent (modal).
4. Clic a "Guardar".

Codi: C-02

**Nom: Baixa al·lergen, categoria o subcategoria.**

Visió: Usuari

Descripció:

Pre: --

Post: S'ha eliminat l'element de la base de dades.

Escenari bàsic:

1. Clic a la icona de configuració per obrir la pàgina.
2. Seleccionar element a eliminar del llistat.
3. Clic a "Eliminar" dins la secció corresponent.
4. Clic a "Acceptar" de la nova finestra emergent (modal), que verifica que no sigui un error.
5. Si (l'element es pot eliminar)
  - Fi
6. Altrament
  - Mostra l'error per pantalla.

Codi: C-03

**Nom: Modificació missatge predefinit.**

Visió: Usuari

Descripció: Modifica un missatge predefinit.

Pre: —

Post: Missatge modificat

Escenari bàsic:

1. Clic a la icona de configuració per obrir la pàgina.
2. Modificar el text del missatge que es vulgui actualitzar.

### 8.1.5.3. Models de dades (back-office)

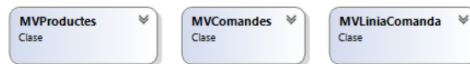
#### Models

Els models seran iguals que els de base de dades i tindran constructors i getters / setters.



#### ViewModels

Implementaré els següents ViewModels per mantenir la informació de la UI a diferents nivells actualitzada.



#### Emmagatzemament i personalitzacions

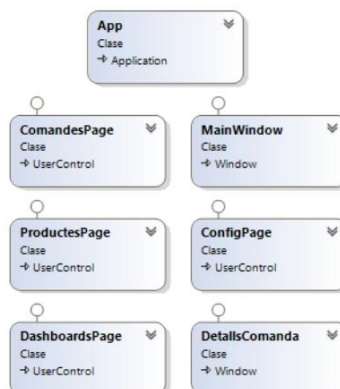
Utilitzaré DadesGlobals per emmagatzemar informació obtinguda de la base de dades gestionar-la en local.

Utilitzaré Styles i Resources per emmagatzemar informació de l'estil de l'aplicació i imatges/icones.



#### Pàgines i finestres

Necessitarem les següents finestres per mostrar tota la informació i llistats.



També necessitarem modals (finestres emergents) per certes parts de l'aplicatiu.

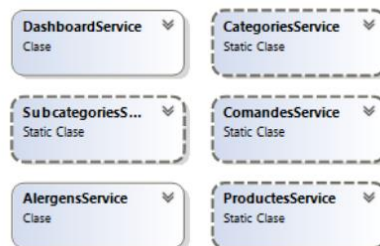


Finalment necessitarem vistes personalitzades per mostrar cada element d'una llista.



### Interacció amb altres serveis

Necessitarem certes classes que emmagatzemin consultes a base de dades i en gestionin la connexió.



## 8.2. Disseny dels diferents sistemes

Aquí comentaré les particularitats de cada part del projecte i adjuntaré imatges del prototip que vaig preparar a l'inici del projecte.

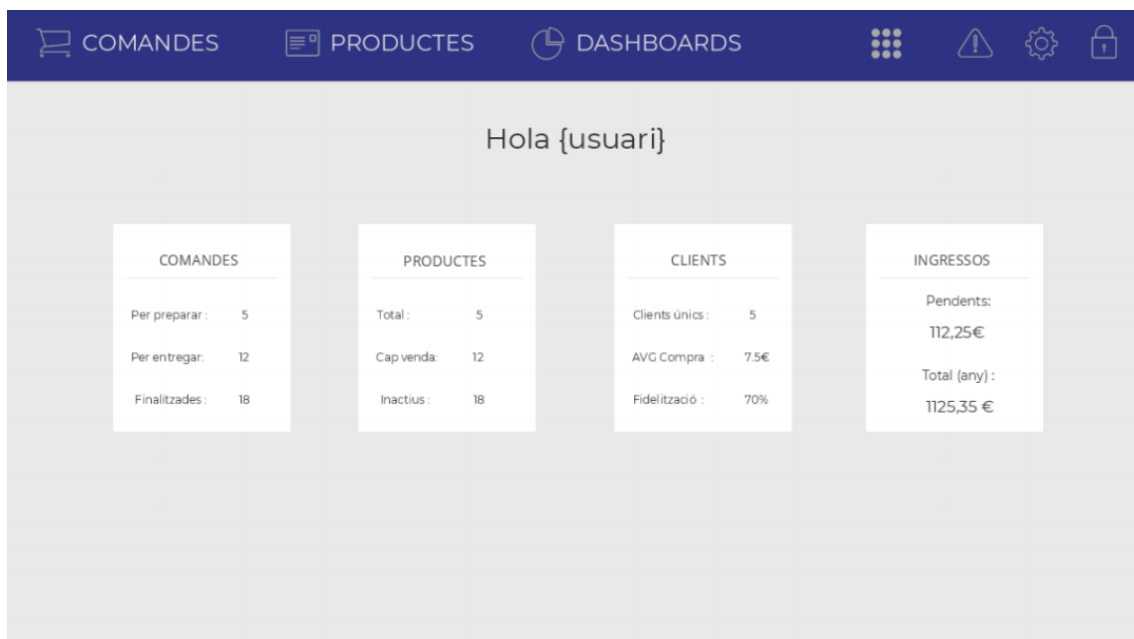
Com ja s'ha esmentat anteriorment, hi han coses que han anat canviant durant el desenvolupament d'aquests (ja sigui per canvi de plantejament en la funcionalitat o per temps).

Per tant, el prototip no determina 100% la presentació final de la web.

### 8.2.1. Back-office

En aquesta part volem centrar-nos només en el que és important per un usuari no especialitzat en botigues online. Per això la idea és fer una pantalla inicial molt visual, amb varis dashboards amb informació de les comandes, clients i productes. Els únics apartats inicialment seran els de gestió de productes i gestió de comandes.

#### 8.2.3.1. Dashboards



### 8.2.3.2. Gestió de comandes

A la pantalla de comandes la intenció és permetre visualitzar de forma ràpida les pendents, veure'n el detall i permetre modificar-ne l'estat [Pendent, en curs, Finalitzada].

| NOM COMANDA    | DATA         | PREU APROXIMAT | PREU REAL   | ESTAT     | TIPUS     | INFO RELLEVANT | ACCIONS                                             |
|----------------|--------------|----------------|-------------|-----------|-----------|----------------|-----------------------------------------------------|
| Name Last name | Company name | Company name   | Lorem ipsum | Pendent   | Recollida | ⓘ ⓘ ⓘ          | <a href="#">Detall</a><br><a href="#">Contactar</a> |
| Name Last name | Company name | Company name   | Lorem ipsum | Pendent   | Recollida | ⓘ              | <a href="#">Detall</a><br><a href="#">Contactar</a> |
| Name Last name | Company name | Company name   | Lorem ipsum | Pendent   | Recollida |                | <a href="#">Detall</a><br><a href="#">Contactar</a> |
| Name Last name | Company name | Company name   | Lorem ipsum | Pendent   | Recollida |                | <a href="#">Detall</a><br><a href="#">Contactar</a> |
| Name Last name | Company name | Company name   | Lorem ipsum | Enviament | Enviament |                | <a href="#">Detall</a><br><a href="#">Contactar</a> |

### 8.2.3.3. Gestió de productes

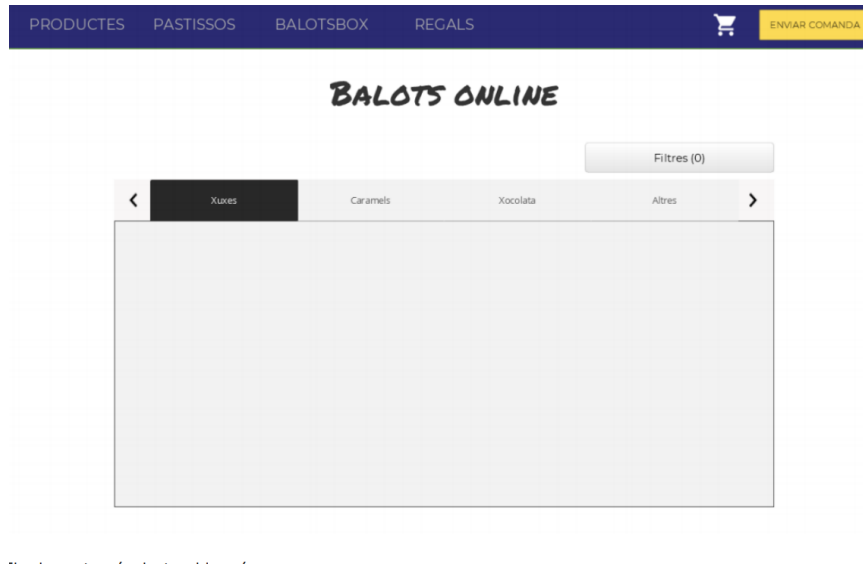
A la pantalla de productes es podrà gestionar els existents o afegir-ne de nous.

| SELECCIONATS                        | PRODUCTE       | CATEGORIA    | PREU UNITAT | PREU KG     | ESTAT |                                                        |
|-------------------------------------|----------------|--------------|-------------|-------------|-------|--------------------------------------------------------|
| <input checked="" type="checkbox"/> | Name Last name | Company name | Lorem ipsum | Lorem ipsum | Actiu | <a href="#">EDITAR</a> <a href="#">ACTIUA / DESACT</a> |
| <input checked="" type="checkbox"/> | Name Last name | Company name | Lorem ipsum | Lorem ipsum | Actiu | <a href="#">EDITAR</a> <a href="#">ACTIUA / DESACT</a> |
| <input checked="" type="checkbox"/> | Name Last name | Company name | Lorem ipsum | Lorem ipsum | Actiu | <a href="#">EDITAR</a> <a href="#">ACTIUA / DESACT</a> |
| <input checked="" type="checkbox"/> | Name Last name | Company name | Lorem ipsum | Lorem ipsum | Actiu | <a href="#">EDITAR</a> <a href="#">ACTIUA / DESACT</a> |
| <input checked="" type="checkbox"/> | Name Last name | Company name | Lorem ipsum | Lorem ipsum | Actiu | <a href="#">EDITAR</a> <a href="#">ACTIUA / DESACT</a> |
| <input checked="" type="checkbox"/> | Name Last name | Company name | Lorem ipsum | Lorem ipsum | Actiu | <a href="#">EDITAR</a> <a href="#">ACTIUA / DESACT</a> |

## 8.2.2. Front-office

### 8.2.4.1. Pàgines de productes

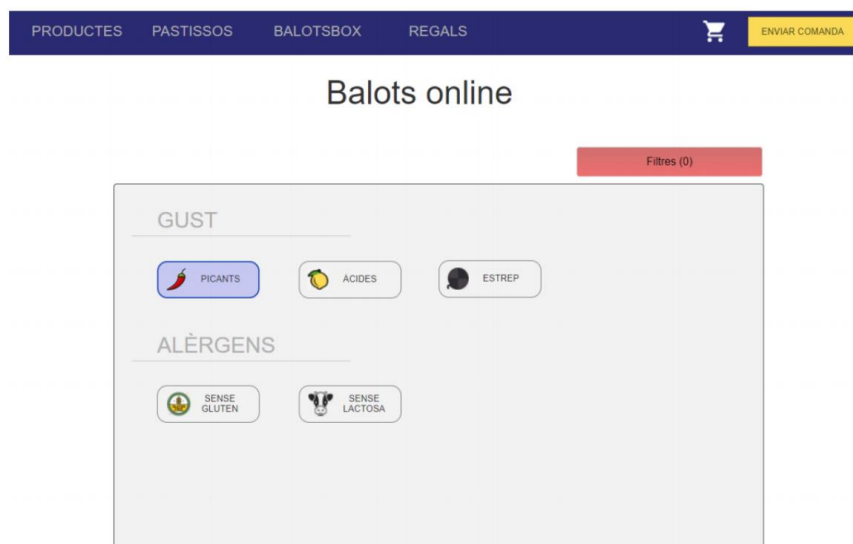
Des de la barra de navegació es podrà navegar entre [Laminadures, Pastissos, BalotsBox i Regals] Les pàgines seran similars (petites diferències en els filtres i la forma de selecció).



Els elements més destacables són:

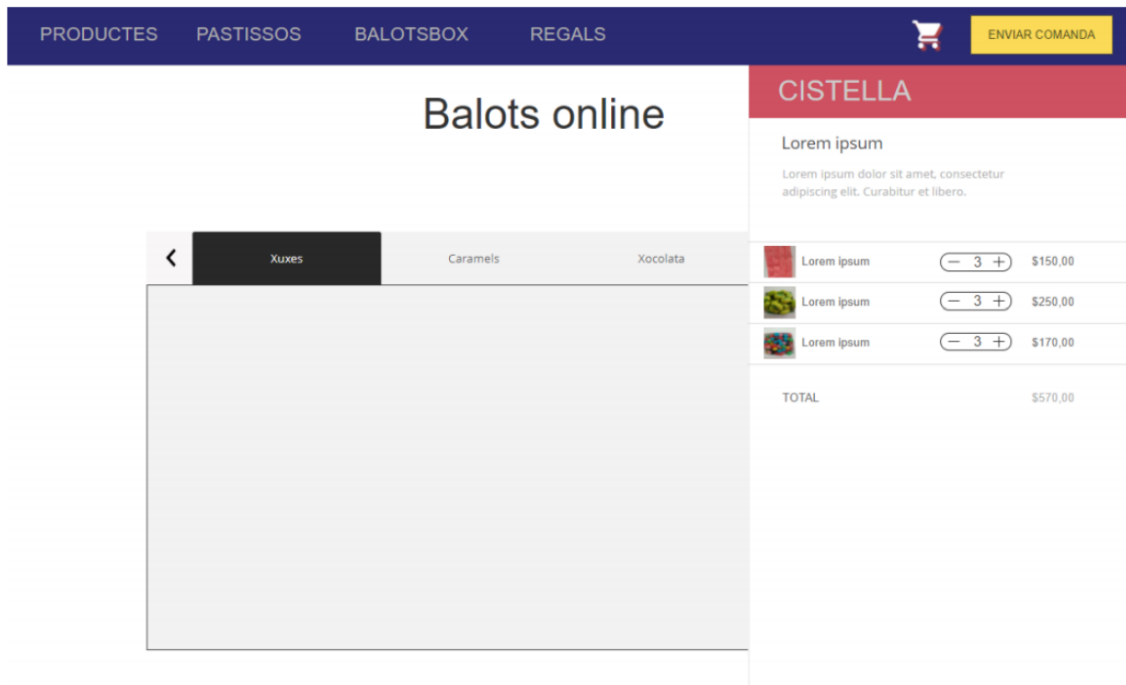
- El panell tabular
- El Grid (taula) amb scroll infinit que contindrà els productes.

A cada una de les pàgines apareixerà el botó de filtres que permetrà a l'usuari veure només aquells productes que li interessin.



### 8.2.4.2. Cistella de compra

La cistella de la compra apareixerà a la barra superior de navegació (al costat del botó de realitzar comanda). Al clicar-lo, desplegarà una pantalla que permetrà veure els productes afegits en format llista i també informarà del preu total aproximat. Al detall de cada producte hi sortiran les unitats i permetrà modificar-les.

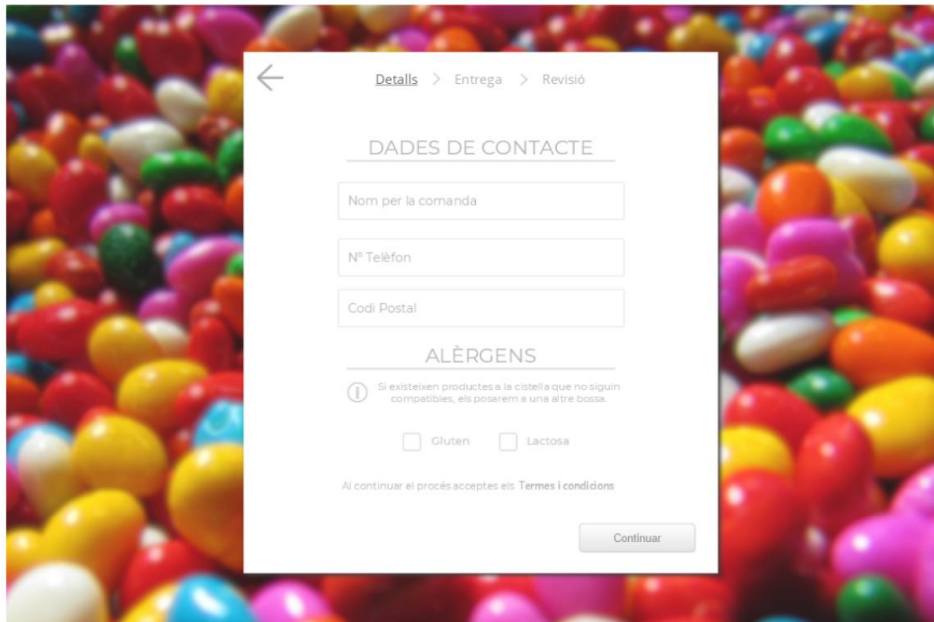




### 8.2.4.3. Checkout

El procés de checkout permetrà escollir quina opció vol, recollida a botiga o entrega a domicili (tot i que aquesta ultima no estarà disponible fins que la web hagi agafat una mica de tracció).

La idea és que sigui un procés senzill, i en l'opció de recollida no volem saturar a l'usuari demanant gaires dades. Demanarem la informació justa per poder processar la comanda i comprovar que és vàlida.



← [Detalls](#) > [Entrega](#) > [Revisió](#)

**DADES DE CONTACTE**

Nom per la comanda

Nº Telèfon

Codi Postal

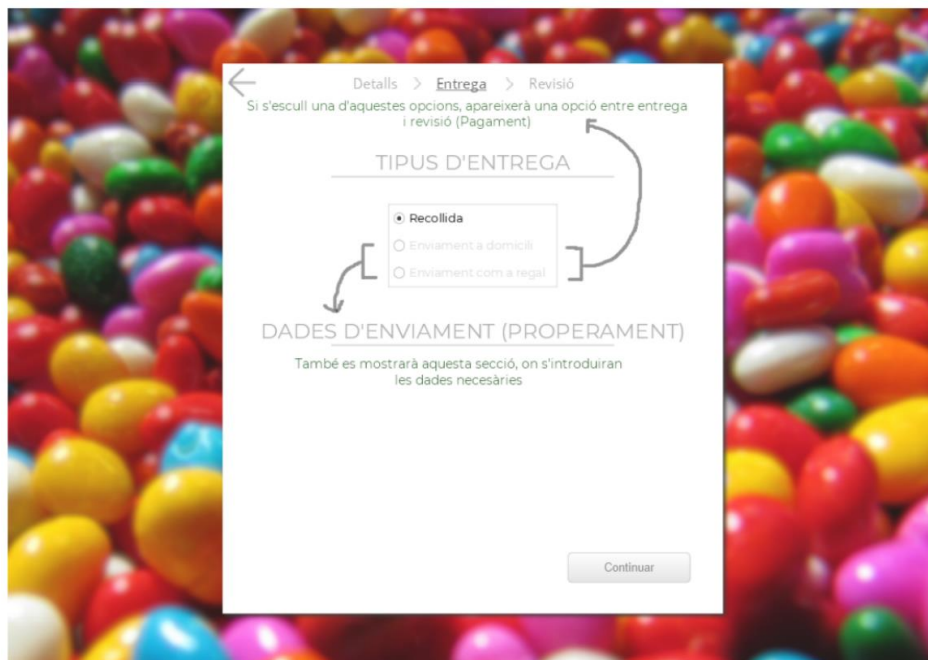
**ALÈRGENS**

**i** Si existeixen productes a la cistella que no siguin compatibles, els posarem a una altre bossa.

Gluten  Lactosa

Al continuar el procés acceptes els [Termes i condicions](#)

Continuar



← [Detalls](#) > [Entrega](#) > [Revisió](#)

Si s'escull una d'aquestes opcions, apareixerà una opció entre entrega i revisió (Pagament)

**TIPUS D'ENTREGA**

Recollida

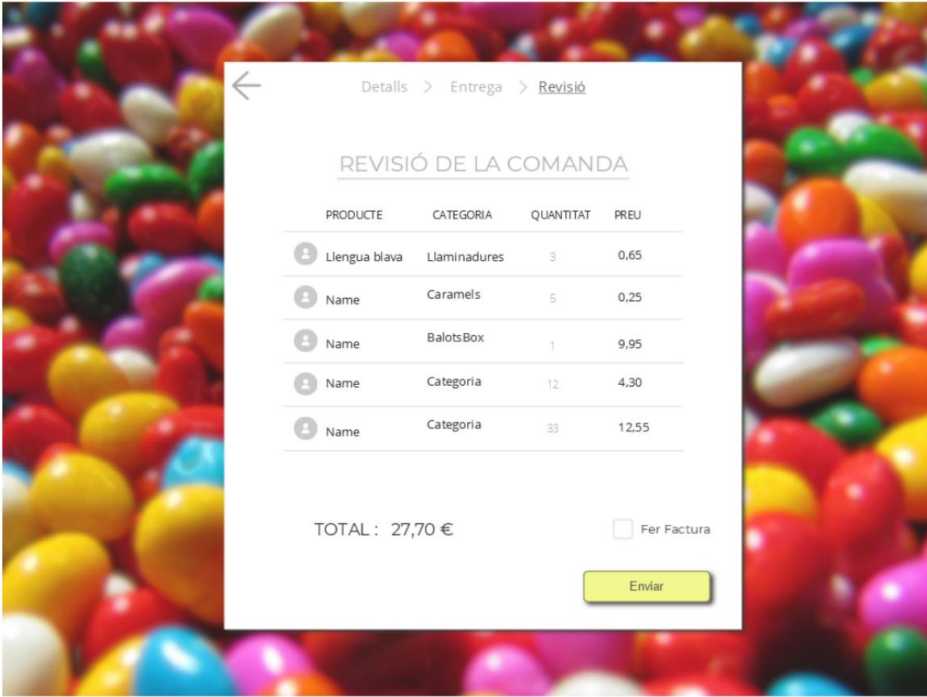
Enviament a domicili

Enviament com a regal

**DADES D'ENVIAMENT (PROPERAMENT)**

També es mostrarà aquesta secció, on s'introduiran les dades necessàries

Continuar



# 9. Implementació i proves

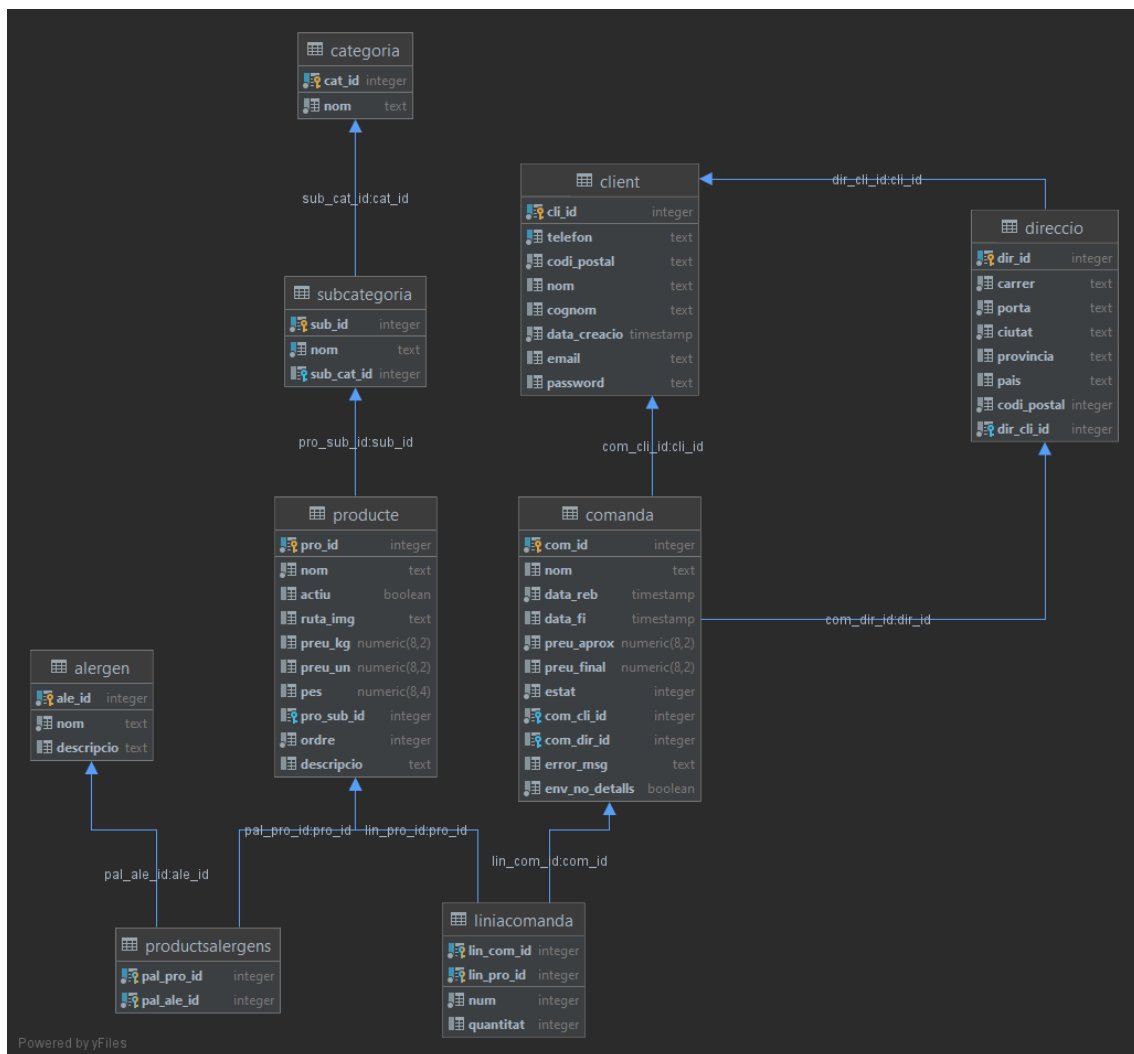
## 9.1. Base de dades

### 9.1.1. Problemes i solucions

He realitzat certs canvis al model plantejat inicialment.

1. He eliminat la relació entre Categoria i producte, ja que em generava una relació circular innecessària.
2. La taula direcció actualment no s'utilitza, i s'hauria d'eliminar (ho faré després de l'entrega del TFG).
3. S'han eliminat les taules relacionades amb el back-office, ja que al utilitzar una aplicació de Windows aquestes propietats que necessitava es poden guardar com a recursos dins el programa.

### 9.1.2. Model actual



## 9.2. API

### 9.2.1. Problemes i solucions

#### 9.2.1.1. Problema amb la vinculació de les línies de comanda amb la comanda durant la creació

La meua idea era pujar el següent objecte com a body a la request de creació de comanda:

```
{
  "nom": "Prova comanda API",
  "dataReb": "2021-05-20T18:30:25.737+00:00",
  "preuAprox": 5.25,
  "comCliId": 1,
  "comDirId": null,
  "envNoDetalls": false,
  "liniesComanda": [
    {
      "linProId": 1,
      "num": 1,
      "quantitat": 1
    },
    {
      "linProId": 1,
      "num": 1,
      "quantitat": 1
    }
  ]
}
```

Però el resultat era aquest:

```
{
  "comId": 4,
  "nom": "Prova comanda API",
  "dataReb": "2021-05-20T18:30:25.737+00:00",
  ...
  "liniesComanda": [
    {
      "linComId": 0,
      "linProId": 1,
      "num": 1,
      "quantitat": 1
    },
    {
      "linComId": 0,
      "linProId": 1,
      "num": 1,
      "quantitat": 1
    }
  ]
}
```

No s'estaven vinculant les línies amb la comanda al moment de crear-les (com es pot veure en la imatge superior, s'assignava 0 a linComId).

## Solució

Gestionar manualment la creació des de ComandaService, guardant primer l'objecte comanda, i després inserir les línies, tal i com es pot veure a la següent imatge:

```
@Transactional
public boolean saveComanda(ComandaRequest comReq) {
    // Veure si existeix client
    try {
        Client client = clientsService.getByPhone(comReq.getTelefon());
        if(client == null) {
            client = clientsService.save(new Client(comReq.getTelefon(), comReq.getCodiPostal()));
        }
        // Guardar la comanda
        Comanda c = new Comanda(comReq.getNomComanda(), comReq.getPreuAprox(), client.getClientId(), comReq.getEnvNoDetails());
        c = comandaRepository.save(c);
        // Guardar les línies de la comanda.
        int i = 1;
        for (LiniaComanda lc : comReq.getLinies()) {
            lc.setLinComId(c.getId());
            lc.setNum(i);
            i++;
        }

        linesComandaRepository.saveAll(comReq.getLinies());

        return true;
    } catch (Exception ex) {
        return false; // Si peta per aquí el client enviarà la comanda per correu.
    }
}
```

Al comentar aquest problema amb el tutor, em va indicar que de la forma que ho feia no estava aprofitant el potencial de Spring.

Volia canviar-ho, però per no invertir massa temps, i tenint en compte ho vaig poder solucionar, no ho he fet.

Com que després del TFG passaré a utilitzar node amb expressjs seria una inversió de temps poc amortitzable.

### *9.2.1.2. Problema amb la CORS Policy*

El problema és simple, a la capçalera hi falta l'element "Access-Control-Allow-Origin", que s'utilitza per determinar si el recurs pot (o no) ser accedit pel contingut dins l'origen actual de la petició.

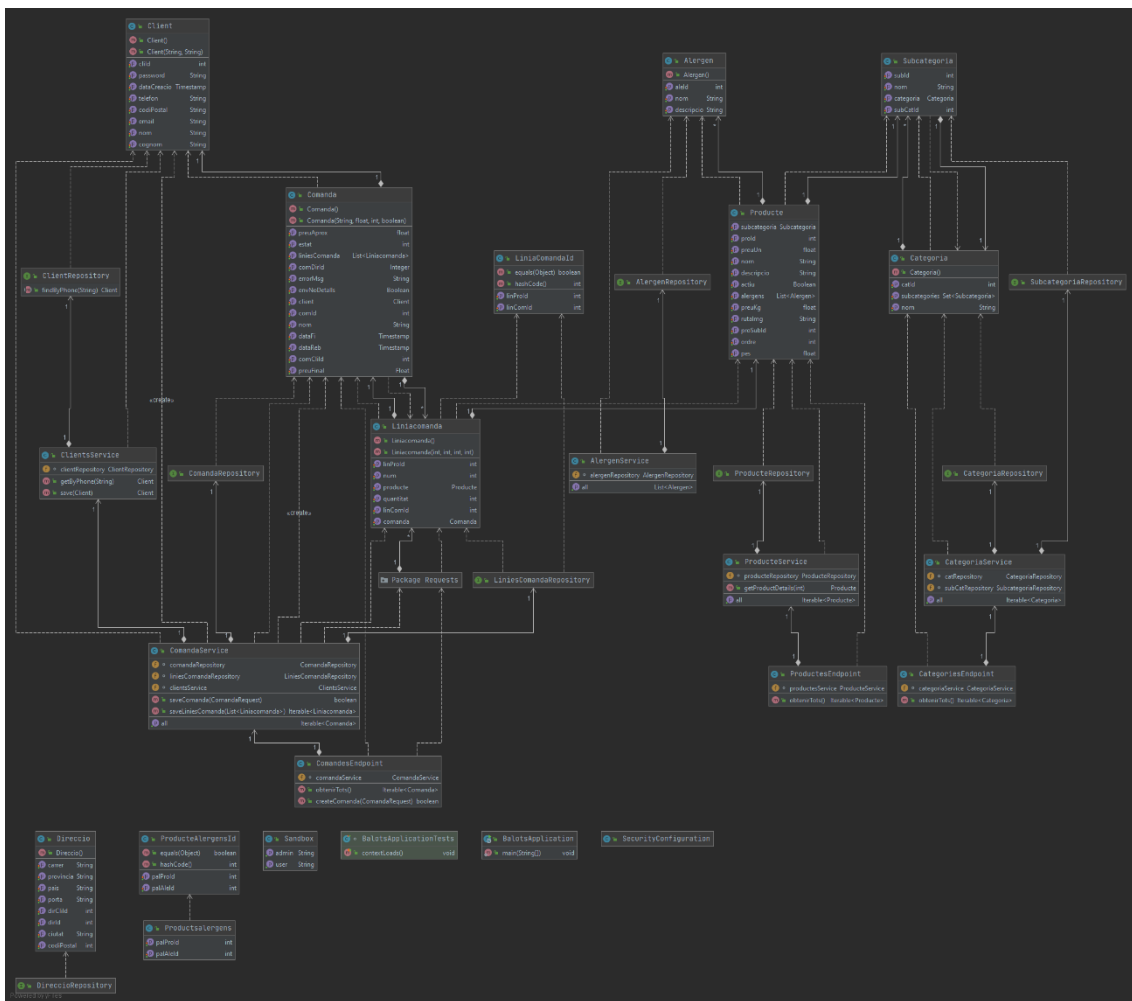
## Solució

Afegir la capçalera des del projecte (API).

```
@CrossOrigin("http://localhost:3000")
@PostMapping(
    consumes = "application/json", produces = "application/json")
public boolean createComanda(@RequestBody ComandaRequest comanda) { return comandaService.saveComanda(comanda); }
```

En aquest cas s'ha habilitat només per la direcció local d'execució del client (localhost:3000), però al passar-ho a producció s'hauria de modificar per la direcció del DNS.

## 9.2.2. Diagrama de classes



## 9.3. Front office

S'ha utilitzat Javascript + React per crear la part funcional de l'aplicació. Explicaré alguns conceptes bàsics per situar el lector, i després mostraré com s'ha utilitzat a la implementació.

### 9.3.1. Conceptes de React

#### React Hooks

Els Hooks són una nova funcionalitat de React 16.8. Permeten interactuar amb l'estat d'un component sense necessitar utilitzar una classe. Són funcions que "s'enllacen" amb el cicle de vida d'un component funcional.

Hi ha disponibles els següents hooks de forma predeterminada:

#### Bàsics

- [useState]
- [useEffect]
- [useContext]

#### Avançats

- [useReducer]
- [useCallback]
- ...

Adicionalment, el programador pot crear-ne de personalitzats si és necessari.

#### Classes vs Components funcionals

La forma més simple de definir un component a React és mitjançant una funció:

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

Aquesta funció és un exemple de component funcional (literalment funcions de JavaScript).

La forma alternativa de fer-ho és utilitzant una classe de ES6:

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

Els dos components anterior són iguals des del punt de vista de React.

La diferència més important és que els primers utilitzen Hooks per controlar el cicle de vida, mentre que els segons utilitzen funcions dins la classe.

React recomana utilitzar el primer mètode a l'hora de desenvolupar nous components, tot i que no deixaran de mantenir l'opció d'utilitzar classes, veuen més correcte utilitzar components funcionals.

### **React Router DOM**

És una eina que permet gestionar rutes en una web app, utilitzant routing dinàmic.

Ens proveeix d'una col·lecció de components de navegació que faciliten la gestió de la navegació.

## **9.3.2. Problemes i solucions**

### 9.3.2.1. Problema amb Context.Api i els components funcionals

Durant la realització del MVP (quan desconeixia totalment el framework), vaig barrejar varis conceptes diferents i no feia res d'una forma unificada.

Un d'aquests casos va ser el Context.Api que ens permet guardar informació en una espècie de contenidor global.

Utilitzava una forma d'implementar-lo que és correcte, però orientada a una estructura amb classes.

Com que la meva estructura actual està basada en funcions (recomanat per la gent de React), aquest sistema anterior no hi encaixava.

A continuació adjunto el codi de la versió antiga :

```
import React, { Component } from 'react'
import { storeProducts, detailProduct } from './data'

const ProductContext = React.createContext();
// PRIMERA DIFERÈNCIA
class ProductProvider extends Component {
  state = {
    products: [],
    pastissos: [],
    detailProduct: detailProduct,
    cart: [],
    estimatedPrice: 0
  };
  // SEGONA DIFERÈNCIA (componentDidMount)
  componentDidMount(){
    this.setProducts();
  };
  // TERCERA DIFERÈNCIA (this.setState en els dos mètodes)
  addToCart = id => {
    ...
    this.state.cart.push(product);
    this.setState({
      products: tempProducts,
      estimatedPrice: estPrice += product.priceKg !== 0 ? (product.priceKg
/ 1000) * product.weight : product.priceUnit
      //cart: this.state.cart.concat(product)
    }));
  };
  removeFromCart = id => {
    ...
```



```

        this.setState(()=>{
            return {products: tempProducts, cart: tempCart, estimatedPrice: estPrice
-
= product.priceKg !== 0 ? (product.priceKg / 1000) * product.weight : product.priceUn
it});
        });});
    render() {
        return (
            <ProductContext.Provider value={{
                ...this.state,
                handleDetail: this.handleDetail,
                addToCart: this.addToCart,
                removeFromCart: this.removeFromCart,
                randomizeCommandName : this.randomizeCommandName
            }}>
                {this.props.children}
            </ProductContext.Provider>
        );
    }
}

// QUARTA DIFERÈNCIA
const ProductConsumer = ProductContext.Consumer;

export { ProductProvider, ProductConsumer };

```

\*He eliminat parts del codi per deixar el més rellevant per aquest problema.

## Solució

Utilitzar la versió pensada per components funcionals proposada per la gent de React.

```

import React, { useContext, useState, useEffect, createContext } from "react";
import axios from "axios";

const AppContext = createContext();
const URLProductes = `http://localhost:9090/api/productes`;

export function AppProvider({ children }) {
    const [productes, setProductes] = useState([]);
    const [cart, setCart] = useState([]);
    const [estimatedPrice, setEstimatedPrice] = useState(0);
    const [navTitol, setNavTitol] = useState();
    //
    // Inicialització
    //
    // useEffect is a lifecycle method for function components, run once after mount (=
= componentDidMount)
    useEffect(() => {
        async function fetchProductes() {
            ...
        };
        setProductes(tempProducts);
    }
    fetchProductes();
}, []);
    const addToCart = id => {
        ...
        cart.push(product);
        setProductes(tempProducts);
        setEstimatedPrice(estPrice += product.preuKg !== 0 ? (product.preuKg / 1000
) * product.pes : product.preuUn);
    }
}

```

```

    };
    const removeFromCart = id => {
        ...
        setCart(tempCart);
        setProductes(tempProducts);
        setEstimatedPrice(estPrice -
= product.preuKg !== 0 ? (product.preuKg / 1000) * product.pes : product.preuUn);
    };

    return (
    <AppContext.Provider
    value={{
        navTitol,
        productes,
        cart,
        estimatedPrice,
        setNavTitol, // TODO : Llest per quan volguem posar titol al NAV
        addToCart,
        removeFromCart,
    }}
    >
    {children}
    </AppContext.Provider>
    );
}

```

// Creem un hook per utilitzar aquest Context, patró de Kent C. Dodds.

```

export function useAppContext() {
    const context = useContext(AppContext);
    if (context === undefined) {
        throw new Error("Context must be used within a Provider");
    }
    return context;
}

```

\*He eliminat parts del codi per deixar el més rellevant per aquest problema.

## Diferències respecte la primera versió

- ComponentDidMount → useEffect()

Els dos s'executen una sola vegada quan s'inicialitza aquesta classe. El primer funciona només dins de classes (és un mètode del cicle de vida de la classe), i el segon és un Hook (introduïts per React a la versió 16.8), i es pot utilitzar des d'un component funcional.

- this.state → useState()

En el primer escenari la classe guarda l'estat com un objecte i es pot modificar directament accedint amb `this.state`.

A la nova versió aquests atributs estan separats i utilitzen Hooks (`useState()` i la funció de set corresponent).

- ProductConsumer → useAppContext()

Aquest canvi és molt rellevant, ja que abans el `ProductConsumer` s'utilitzava dins la renderització d'una classe:

```
<TabPanel value={value} index={0} dir={theme.direction}>
  <div className="row">
    <ProductConsumer>
      {value => {
        return value.products.map( product => {
          if(product.category === "xuxes" && product.active)
            return <Product key={product.id} product={product} />
          else return "";
        })
      }}
    </ProductConsumer>
  </div>
</TabPanel>
```

Ara, en canvi la funció `useAppContext()` utilitza els hook `useContext()` per retornar el context de l'aplicació.

Això ens permet inicialitzar-lo a l'inici del component funcional i utilitzar-lo on tant durant la renderització com en funcions auxiliars.

```
export default function FullwidthTabs() {
  const classes = useStyles();
  const theme = useTheme();
  const [value, setValue] = React.useState(0);

  const context = useAppContext();
```

```
  const handleChangeIndex = (index) => {
    setValue(index);
    console.log(context.productes);
  };
```

```
<TabPanel value={value} index={1} dir={theme.direction}>
  {context.productes.map((product) => {
    if (product && product.proSubId === 2)
      return <Product key={product.proId} product={product} />;
  })}
</TabPanel>
```

### 9.3.2.2. Problema amb la renderització del producte dins la taula principal

Al renderitzar la taula de productes, la web donava un error i un warning (i els productes no es mostraven):

- Object is not valid as a React child [...]
- Warning: Each child in a list should have a unique "key" prop

#### **Solució**

Vaig perdre unes quantes hores buscant informació sobre què podia estar passant. Finalment va resultar ser una "tonteria":

```
{context.productes.map((product) => {  
  if (product && product.proSubId === 2)  
    return <Product key={product.proId} product={product} />;  
  else  
    return "";  
})}
```

Aquest else return "" era el causant del drama.

La nova versió és la següent:

```
{context.productes.map((product) => {  
  if (product && product.proSubId === 2)  
    return <Product key={product.proId} product={product} />;  
})}
```

### 9.3.2.3. Pèrdua del context al canviar de pàgina

Quan canviava de pàgina la cistella (i la resta d'informació) es perdia.

Va portar una bona estona i va resultar ser una tonteria un altre cop:

Al no estar utilitzant el component <Link> de React Router Dom, al canviar de pàgina es tornava a executar tot el cicle de vida de la pàgina i perdia la informació.

#### **Solució**

Es va solucionar canviant les següents línies de codi:

Versió antiga:

```
import Link from '@material-ui/core/Link';  
<Link href="/pastissos" >  
  Link  
</Link>
```

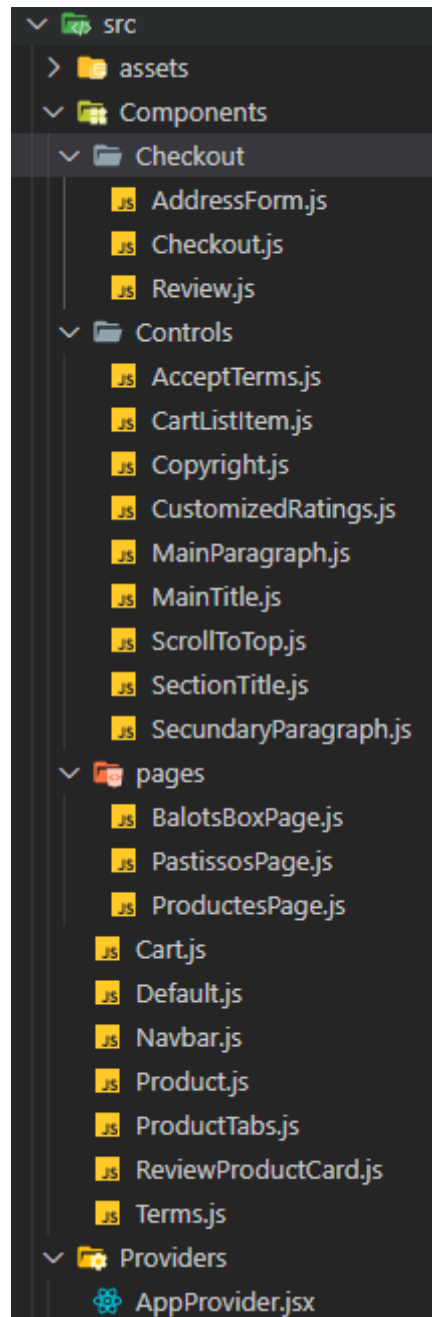
Versió actual:

```
import { Link } from "react-router-dom";  
<Link to="/pastissos">  
  <ListItemText primary='Pastissos' />  
</Link>
```

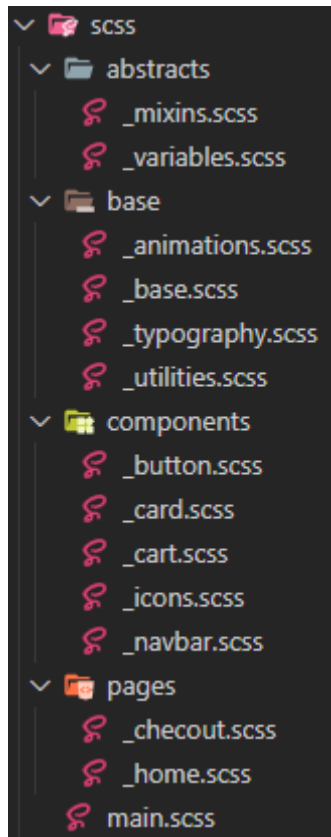
### 9.3.3. Estructura dels components

No utilitzo un sistema de tipat per Javascript (com podria ser TypeScript), més aviat es basa en components (funcions) que s'insereixen fins un html per mostrar la informació desitjada.

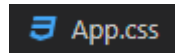
A continuació llistaré l'estructura del codi:



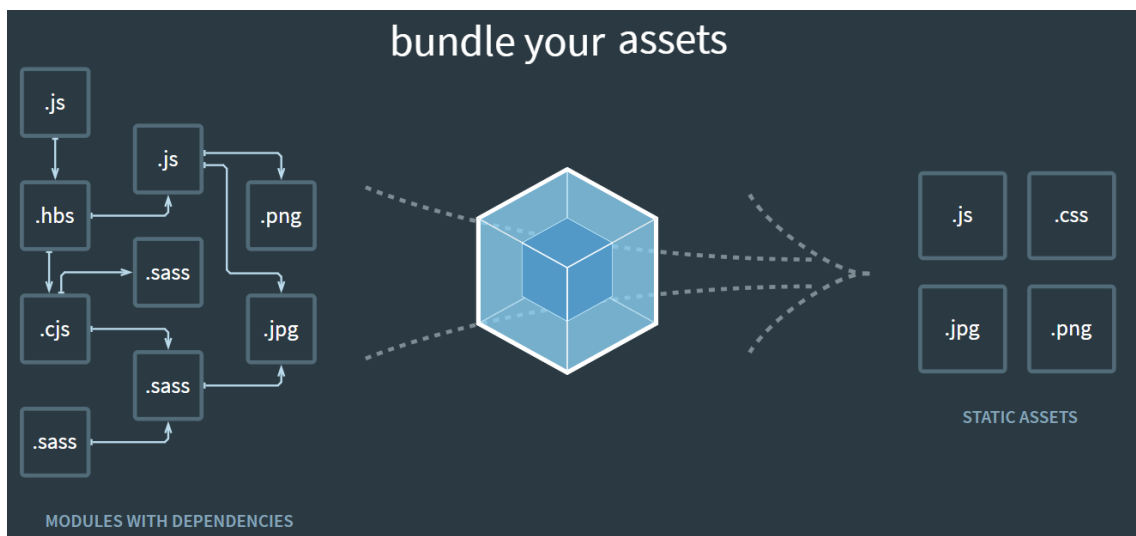
I els estils:



Aquests diferents fitxers d'estil s'unifiquen a un de sol durant la compilació, que acaba "important" tot el que es defineix en aquests.



Al compilar també s'utilitza "Webpack", que s'encarrega d'unificar tots els fitxers d'un tipus en un de sol i "minificar-lo" (reduir-lo al màxim modificant noms de variables i funcions per tal que el fitxer resultant ocupi el menys possible).



### 9.3.4. Codi més rellevant

#### 9.3.4.1. Media queries

No és gens complicat, però trobo que és molt rellevant per una correcta visualització de la web.

Les media queries es van afegir a CSS3 i permeten renderitzar continguts basant-se en unes regles que indiquen com ha d'actuar la web per diferents grups de resolucions (PC i mòbil per exemple).

Es va convertir en una recomanació estàndard de W3C l'any 2012.

Definició:

```
// MEDIA QUERY MANAGER
/*
  Mesura      Param      Descripció
  -----
  0 -- 600 px : [phone]   phone
  600 -- 900 px : [tab-port] tablet portrait
  900 -- 1200 px : [tab-land] tablet landscape
  1200 -- 1800 px : [ ----- ] Desktop normal (default)
  + 1800 px : [big-desk] Desktop big
*/
@mixin respond($breakpoint) {
  @if $breakpoint == phone {
    @media(max-width: 600px) {@content};
  }
  @if $breakpoint == tab-port {
    @media(max-width: 900px) {@content};
  }
  @if $breakpoint == tab-land {
    @media(max-width: 1200px) {@content};
  }
  @if $breakpoint == big-desk {
    @media(min-width: 1800px) {@content};
  }
}
```

```
.cart {
width: 45vw; // TODO: Media Query --> Mobils 90%, Ipad --> 50% , PC --> 40%
height: 100%;
display: flex;
flex-direction: column;

  @include respond(tab-land) { ←
width: 70vw;
  }

  @include respond(phone) { ←
width: 100vw;
  }

  &_header { ...
}

  &_list { ...
}

  &_total { ...
}

  &_cta { ...
}

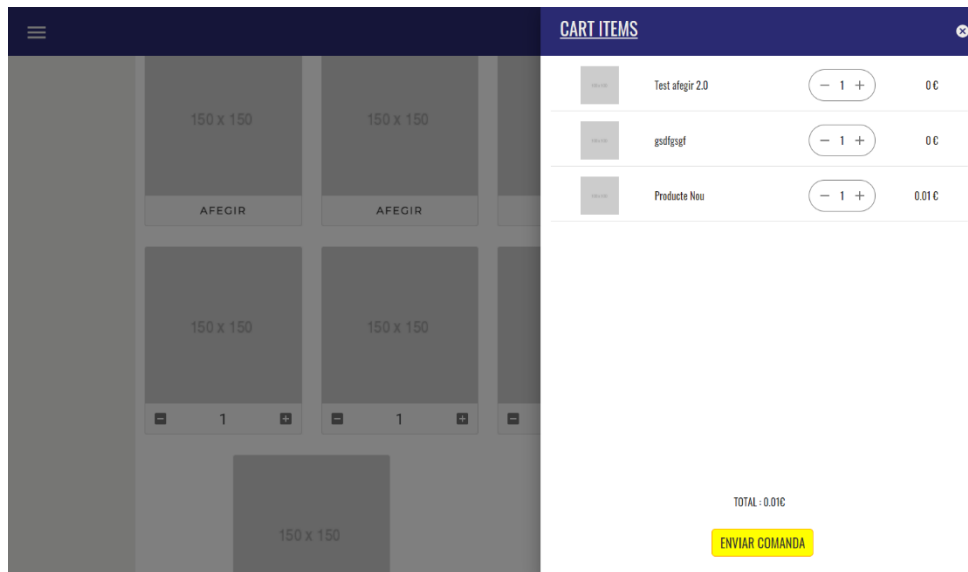
  &_item {
display: flex;
flex: 1 1 auto;
flex-direction: column;
align-items: center;
vertical-align: middle;
justify-content: space-around;

    & p {
margin: 0 0 10px;
width: 30%;

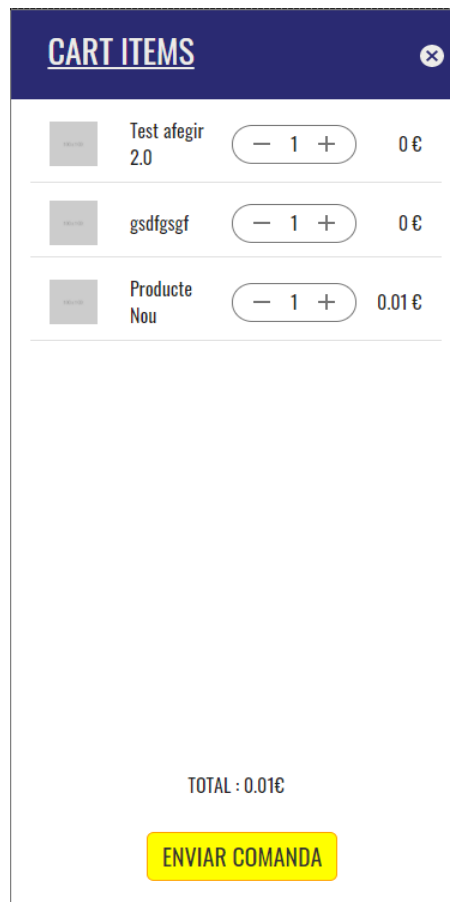
      @include respond(phone) { ←
width: 20%;
      }
    }
  }
}
```

Aquestes parts de codi CSS permeten el següent:

Vista des d'un ordinador:



Vista des d'un mòbil del mateix component:



Com podem veure, en el cas del mòbil ocupa tota la pantalla per tal de poder visualitzar bé el llistat.



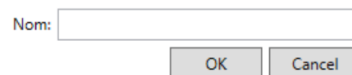
## 9.4. Back office

### 9.4.1. Problemes i solucions

#### 9.4.1.1. Reutilització de modals

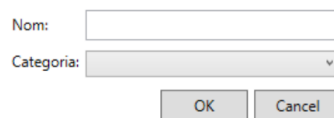
En el procés de creació de categories, subcategories i al·lèrgens s'utilitza un modal pràcticament igual:

#### Categories



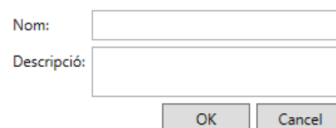
Modal dialog for creating a category. It features a single text input field labeled "Nom:" and two buttons: "OK" and "Cancel".

#### Subcategories



Modal dialog for creating a subcategory. It features a text input field labeled "Nom:", a dropdown menu labeled "Categoria:", and two buttons: "OK" and "Cancel".

#### Al·lèrgens



Modal dialog for creating an allergen. It features two text input fields: "Nom:" and "Descripció:", and two buttons: "OK" and "Cancel".

Per això vaig decidir implementar-ne un de genèric que continguéss la informació dels tres, i es mostrés o no en base a un paràmetre. (ConfigItemDialog)

#### Vista

```
<Window x:Class="balots_office.UserControls.Windows.ConfigItemDialog"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:local="clr-namespace:balots_office.UserControls.Windows"
mc:Ignorable="d"
Title="Alta"
MinHeight="120"
MinWidth="300"
SizeToContent="WidthAndHeight"
ResizeMode="NoResize"
ShowInTaskbar="False"
WindowStartupLocation="CenterOwner"
Height="180" Width="300">
<Grid x:Name="grid" Margin="10">
<Grid.Resources>
<!-- Default settings for controls -->
<Style TargetType="{x:Type Label}">
<Setter Property="Margin" Value="0,3,5,5" />
<Setter Property="Padding" Value="0,0,0,5" />
</Style>
<Style TargetType="{x:Type TextBox}">
```

```

        <Setter Property="Margin" Value="0,0,0,5" />
    </Style>
    <Style TargetType="{x:Type ComboBox}">
        <Setter Property="Margin" Value="0,3,0,10" />
        <Setter Property="Padding" Value="0,0,0,5" />
    </Style>
    <Style TargetType="{x:Type Button}">
        <Setter Property="Width" Value="70" />
        <Setter Property="Height" Value="25" />
        <Setter Property="Margin" Value="5,0,0,0" />
    </Style>
</Grid.Resources>

<Grid.ColumnDefinitions>
    <ColumnDefinition Width="Auto" />
    <ColumnDefinition />
</Grid.ColumnDefinitions>

<Grid.RowDefinitions>
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition />
</Grid.RowDefinitions>

<!-- Form-->
<Label Name="labNom" Grid.Column="0" Grid.Row="0">Nom:</Label>
<TextBox Name="tbNom" Grid.Column="1" Grid.Row="0" VerticalContentAlignment="
Center" />

    <Label x:Name="labDescripcio" Grid.Column="0" Grid.Row="1" Visibility="Collap
sed">Descripció:</Label>
    <TextBox x:Name="tbDescripcio" Grid.Column="1" Grid.Row="1" Grid.RowSpan="2"
MinHeight="40" VerticalContentAlignment="Center" Visibility="Collapsed"/>

    <Label x:Name="labCategoria" Grid.Column="0" Grid.Row="3" Visibility="Collaps
ed">Categoria:</Label>
    <ComboBox x:Name="comCategoria" Grid.Column="1" Grid.Row="3" HorizontalAlignm
ent="Stretch" Visibility="Collapsed" />

    <!-- Accept or Cancel -->
    <StackPanel Grid.Column="0" Grid.ColumnSpan="2" Grid.Row="4" Orientation="Hor
izontal" HorizontalAlignment="Right">
        <Button Name="okButton" Click="okButton_Click" IsDefault="True">OK</Butto
n>
        <Button Name="cancelButton" IsCancel="True" Click="cancelButton_Click">Ca
ncel</Button>
    </StackPanel>
</Grid >
</Window>

```

Va aparèixer un problema amb aquesta decisió en el moment d'utilitzar "Visibility.Hidden" pels elements de la vista que no s'havien de mostrar.

## Solució

Vaig descobrir la funció del valor "Visibility.Collapsed", que permet no reservar espai al layout (finestra) en cas que no es mostri aquest element.

Gràcies a això he pogut reutilitzar aquest component i no haver-ne de crear un per cada cas.

El codi:

```
using balots_office.Infrastructure;
using balots_office.Models;
using balots_office.utils;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Threading.Tasks;
using System.Windows;

namespace balots_office.UserControls.Windows
{
    /// <summary>
    /// Lògica de interacció para ConfigItemDialog.xaml
    /// </summary>
    public partial class ConfigItemDialog : Window
    {
        private const string NOM = "NOM";
        private const string CAT = "CAT";

        List<Categoria> categories = null;
        EConfigType tipusEntitat;
        object newObject;

        public ConfigItemDialog(EConfigType configType, List<Categoria> categories =
null)
        {
            InitializeComponent();
            tipusEntitat = configType;
            newObject = new object();

            switch (configType)
            {
                case EConfigType.ALLERGEN:
                    labDescripcio.Visibility = Visibility.Visible;
                    tbDescripcio.Visibility = Visibility.Visible;
                    break;
                case EConfigType.SUBCATEGORY:
                    labCategoria.Visibility = Visibility.Visible;
                    comCategoria.Visibility = Visibility.Visible;
                    this.categories = categories;
                    comCategoria.ItemsSource = getCatNames();
                    break;
                default:
                    break;
            }
        }

        private IEnumerable getCatNames()
        {
            List<string> col = new List<string>();
            categories.ForEach((cat) => { col.Add(cat.nom); });
            return col;
        }
    }
}
```

```

public object Return
{
    get { return newObject; }
}

private async void okButton_Click(object sender, RoutedEventArgs e)
{
    labNom.Foreground = Styles.black;
    labCategoria.Foreground = Styles.black;

    List<string> invalids = validForm();
    if (invalids.Count.Equals(0))
    {
        switch (tipusEntitat)
        {
            case EConfigType.ALLERGEN:
                Alergen all = new Alergen();
                all.nom = tbNom.Text;
                all.descripcion = tbDescripcion.Text;
                all = await AlergensService.Add(all);
                newObject = all;
                break;
            case EConfigType.CATEGORY:
                Categoria cat = new Categoria();
                cat.nom = tbNom.Text;
                cat = await CategoriesService.Add(cat);
                newObject = cat;
                break;
            case EConfigType.SUBCATEGORY:
                Subcategoria sub = new Subcategoria();
                sub.nom = tbNom.Text;
                sub.subCatId = categories[comCategoria.SelectedIndex].catId;
                sub = await SubcategoriesService.Add(sub);
                sub.categoria = categories[comCategoria.SelectedIndex];
                newObject = sub;
                break;
            default:
                break;
        }
        DialogResult = newObject != null;
        Close();
    }
    else
    {
        foreach (string item in invalids)
        {
            if(item == NOM)
            {
                labNom.Foreground = Styles.negative;
            }
            else if(item == CAT)
            {
                labCategoria.Foreground = Styles.negative;
            }
        }
    }
}

private async Task<Subcategoria> AddSubCat(Subcategoria sub)
{
    await Task.Run(() =>
    {
        return SubcategoriesService.Add(sub);
    }).ContinueWith((result) => {
        return result.Result;
    });
}

```

```

    });
    return null;
}

private List<string> validForm()
{
    List<string> invalids = new List<string>();

    if (tbNom.Text.Length < 2)
    {
        invalids.Add(NOM);
    }
    if(tipusEntitat == EConfigType.SUBCATEGORY)
    {
        if(comCategoria.SelectedIndex.Equals(-1))
        {
            invalids.Add(CAT);
        }
    }

    return invalids;
}

private void cancelButton_Click(object sender, RoutedEventArgs e)
{
    Close();
}
}
}
}

```

#### 9.4.1.2. Problema al utilitzar ADO.NET Entity Data Model (Database First)

Aquesta tecnologia permet generar models i totes les classes necessàries per la interacció de l'aplicació amb la base de dades a partir d'un esquema de la base de dades.

L'havia utilitzat en un altre projecte (amb una base de dades de Microsoft SQL Server) i m'havia funcionat genial.

Per fer-ho servir amb PostgreSQL vaig tenir que instal·lar un connector de tercers, cosa que ja no em va agradar gaire, però els problemes van venir al veure que no em permetia editar les relacions many to many.

Quan intentava assignar un al·lergen a un producte (en teoria guardant les dades a la taula "productsalergens"), em generava un al·lergen nou cada cop.

#### **Solució**

Vaig perdre bastant temps investigant aquest problema, però al final (i sumat a que tenia un rendiment bastant pèssim) vaig decidir fer un refactor de tota aquesta part; realitzar la connexió manualment i preparar les queries jo mateix. Vaig utilitzar una llibreria anomenada "npgsql" que permet realitzar aquestes operacions de forma prou senzilla.

Va ser una gran decisió, ja que amb unes 6-8h vaig tenir-ho tot llest i va passar a funcionar perfecte.

La implementació és la següent:

```
/// ProductesService.cs : 233

private static async Task InsertAlergensProducte(int proId, List<Alergen> alergens)
{
    Try {
        if (alergens != null && alergens.Count > 0)
        {
            conn.Open();

            // Afegim la relació amb els al.lergens seleccionats.
            string alergensQuery = "INSERT INTO productsalergens (pal_pro_id, pa
l_ale_id) VALUES ";
            for (int i = 0; i < alergens.Count; i++)
            {
                if (i != 0)
                {
                    alergensQuery += ", ";
                }
                alergensQuery += "(" + proId + "," + alergens[i].aleId + ")";
            }

            using (NpgsqlCommand comand = new NpgsqlCommand(alergensQuery, conn))
            {
                NpgsqlDataReader results = await comand.ExecuteReaderAsync();
            }
            conn.Close();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

I la forma de cridar-ho és la següent:

```
await InsertAlergensProducte(producte.proId, producte.alergens);
```

## 9.4.2. Algorismes més rellevants

A continuació mostraré un exemple de la implementació del patró MVVM esmentat anteriorment.

El llistat de comandes s'ha d'actualitzar automàticament quan es realitza alguna acció sobre aquestes, per això s'ha implementat MVComandes, que desenvolupa la tasca de ViewModel per aconseguir aquesta comunicació "reactiva".

Els atributs d'aquesta classe són els següents:

```
public ObservableCollection<ListItemComanda> Nodes { get; set; }
private ICollectionView _comandesView;
```

ListItemComanda és la vista que conté una comanda. Guardem una llista d'aquests elements de la interfície (un per cada comanda a la base de dades) a la variable Nodes.

\_comandesView, una Col·lecció de vistes que es filtra automàticament en base a certes condicions sobre els elements de la llista "Nodes", i actualitza directament què s'està veient per pantalla.

S'invoca utilitzant el següent mètode:

```
public void AplicarFiltre(bool pendants = true, bool recollida = true)
{
    if(recollida && pendants) // Recollides pendants
    {
        _comandesView.Filter = new Predicate<object>(obj => {
            ListItemComanda item = obj as ListItemComanda;
            return Recollida(item.ComandaVinculada) && Pendent(item.ComandaVinculada)
&& !Error(item.ComandaVinculada.estat);
        });
    }
    else if (recollida && !pendants) // Recollides finalitzades
    {
        _comandesView.Filter = new Predicate<object>(obj => {
            ListItemComanda item = obj as ListItemComanda;
            return Recollida(item.ComandaVinculada) && !Pendent(item.ComandaVinculada)
&& !Error(item.ComandaVinculada.estat);
        });
    }
    else if (!recollida && pendants) // Enviaments pendants
    {
        _comandesView.Filter = new Predicate<object>(obj => {
            ListItemComanda item = obj as ListItemComanda;
            return Enviament(item.ComandaVinculada) && Pendent(item.ComandaVinculada)
&& !Error(item.ComandaVinculada.estat);
        });
    }
    else // Enviaments finalitzats
    {
        _comandesView.Filter = new Predicate<object>(obj => {
            ListItemComanda item = obj as ListItemComanda;
            return Enviament(item.ComandaVinculada) && !Pendent(item.ComandaVinculada)
&& !Error(item.ComandaVinculada.estat);
        });
    }

    _comandesView.Refresh();
}
```

l es basa en els següents criteris:

```
private bool Pendent(Comanda comanda)
{
    return comanda.estat == 0 || comanda.estat == 1;
}

private bool Recollida(Comanda comanda)
{
    return comanda.comDirId == 0;
}

private bool Enviament(Comanda comanda)
{
    return comanda.comDirId.HasValue && comanda.comDirId.Value > 0;
}

private bool Error(int estat)
{
    return estat == 3;
}
```

Quan canvia l'estat d'un element de la llista "Nodes", també informa als "subscriptors", en aquest cas el llistat de comandes, perquè renderitzi els canvis.

Aquesta subscripció a MVComandes la podem veure a la part d'interfície:

```
<ItemsControl x:Name="llistaComandes" Margin="25 0 25 15" HorizontalAlign
ment="Stretch" VerticalAlignment="Stretch" MinWidth="1300" MinHeight="600
" Background="WhiteSmoke" ItemsSource="{Binding ComandesView}">
```



## **9.5. Testing**

No he tingut prou temps com per implementar un sistema de tests elaborat. He realitzat molt testing seguint la tècnica del "Monkey testing", que es basa en què l'usuari provi l'aplicació introduint inputs aleatoris i comprovant el comportament de l'aplicació. (intentant fer que falli atacant els punts febles coneguts).

També he fet ús de les eines de debug per comprovar línia per línia les funcionalitats principals.

### **9.5.1. API**

En aquest cas he preparat diferents consultes amb informació variada per veure què retorna l'API.

### **9.5.2. Front-office**

En aquest cas he fet ús dels ReactDev Tools (esmentat anteriorment) per poder veure què està passant en cada moment, i modificar l'estat a temps real per veure com reacciona la web.

### **9.5.2. Back-office**

En aquest cas he fet un ús exhaustiu de l'eina de debug que proporciona Visual Studio 2019 (molt senzilla i potent).

## 10. Implantació i resultats

Aquest apartat ha de descriure amb detall el procés de desenvolupament que ha calgut dur a terme per implantar el sistema desenvolupat. L'apartat de resultats ha de mostrar clarament el grau d'assoliment dels objectius, mitjançant exemples del funcionament d'allò que s'ha fet. Es revisarà i valorarà que el sistema desenvolupat compleixi amb la legislació i normatives vigents, especialment en el que fa referència a la llei orgànica de protecció de dades de caràcter personal (LOPD) i a la llei de serveis de la societat de la informació i comerç electrònic (LSSICE). A més, ha de mostrar que l'aplicació resultant del projecte fa les tasques especificades originalment.

Inicialment volia implantar els sistemes a AWS, tal i com he explicat anteriorment, però vaig descobrir Heroku, que simplifica bastant tota la part de deploy; i he optat per publicar-ho en aquesta plataforma.

Encara hi ha feina per arribar a una versió estable, però permetrà tenir un entorn de producció on anar aplicant els canvis i provant.

Necessitarem dues aplicacions pel projecte:

| Nom aplicació Heroku | Responsabilitats    |
|----------------------|---------------------|
| Balots-api           | API + Base de dades |
| Balots               | Front-office        |



A continuació veurem com publicar aquests diferents serveis.

### Resum de tasques a realitzar

- Crear DB
- Restaurar Backup
- Canviar connexió a l'API perquè apunti a la DB.
- Deploy API
- Modificar front perquè ataquí a la url de l'api a Heroku
- Deploy Front
- Modificar API per el cross-origin amb la IP del front-office
- Deploy API
- Compilació del programa back-office en mode 'release'.

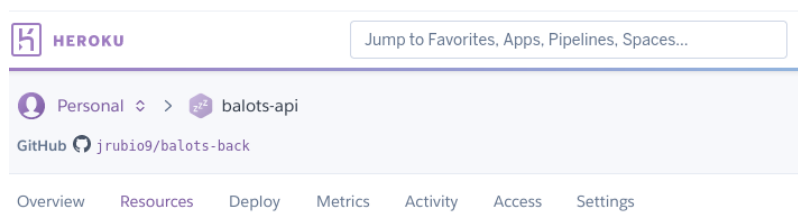
## 10.1. Implantació backend

El primer pas és crear una aplicació.

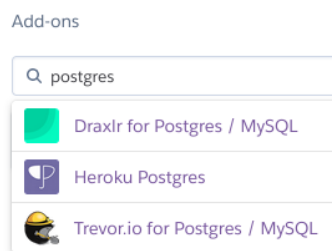
Un cop hem fet login a Heroku i estem al perfil d'usuari, veurem un botó a la part dreta per crear una nova aplicació (només ens demana un nom vàlid i una ubicació). Seguim el procés i finalitzem.

### 10.1.1. Base de dades

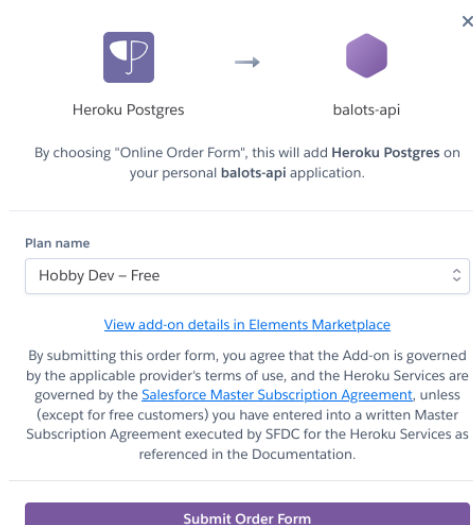
Accedim a l'aplicació balots-api, i ens movem a la segona opció de la barra superior (Resources)



Ens situem a l'apartat d'Add-ons, i busquem "PostgreSQL"

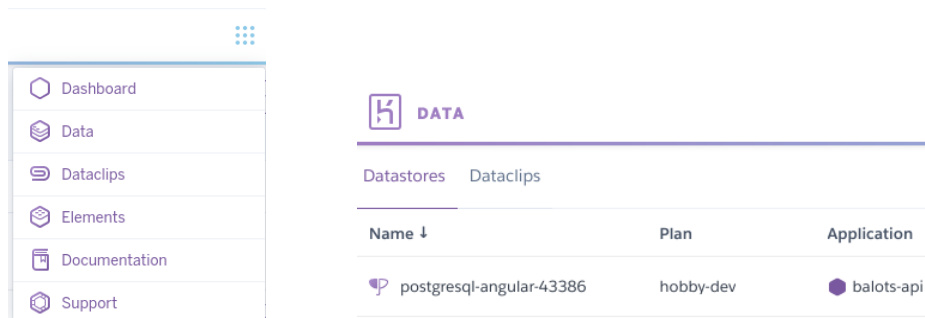


Al seleccionar la segona opció s'obrirà un modal de configuració.

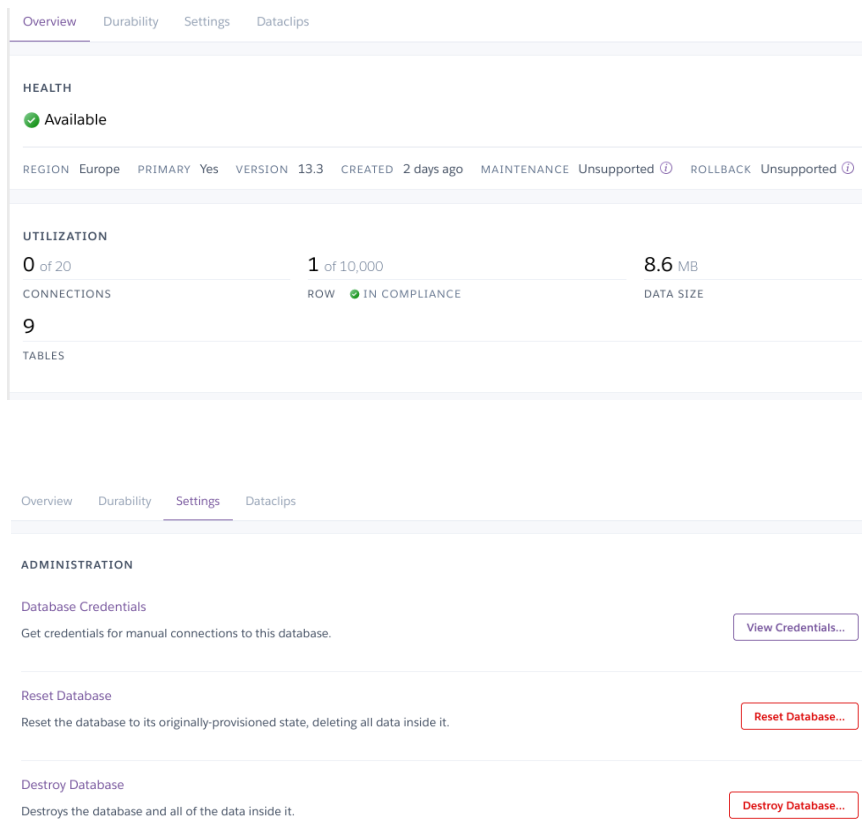


Esperem uns minuts a que es registri i llestos! Ja tenim la base de dades creada!

Podem veure'n els detalls accedint a la secció de Data (des del menú situat a la part superior dreta de la pantalla)



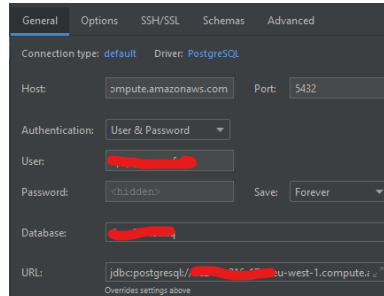
Si hi fem clic podem veure'n els detalls.



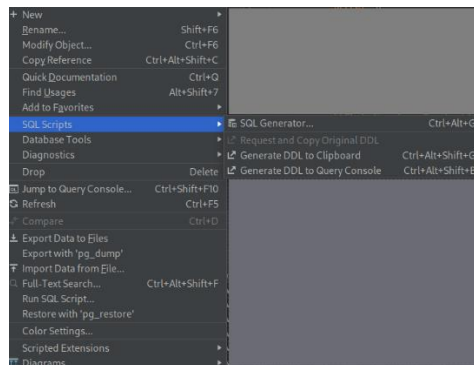
L'únic que ens queda per acabar aquesta secció és crear l'estructura de taules dins la base de dades.

Per fer-ho, he realitzat els següents passos:

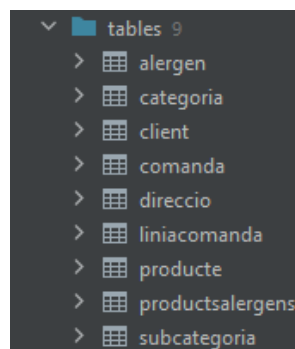
1. Configurar la connexió amb aquesta base de dades des de IntelliJ DataGrip (SGBD).



2. Exportar els scripts SQL de creació de taules a partir de la base de dades que tinc en local (utilitzant l'eina SQL Generator).



3. Executar aquest script a la connexió amb la base de dades al núvol.



Ja tenim la base de dades creada i llesta per començar a rebre informació.

## 10.1.2. API

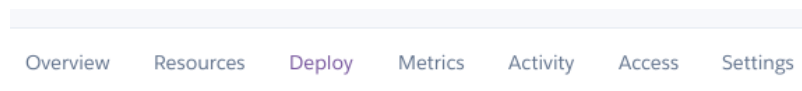
El primer pas és modificar la configuració de connexió entre l'API i la base de dades, per tal que apunti a la nova base de dades al núvol.

Ho farem des del fitxer *application.properties* de Spring.

```
# Postgres
spring.datasource.url= jdbc:postgresql://ec2-5
spring.datasource.username= [redacted]
spring.datasource.password= [redacted]
spring.datasource.sql-script-encoding=UTF-8
spring.datasource.initialization-mode=always
```

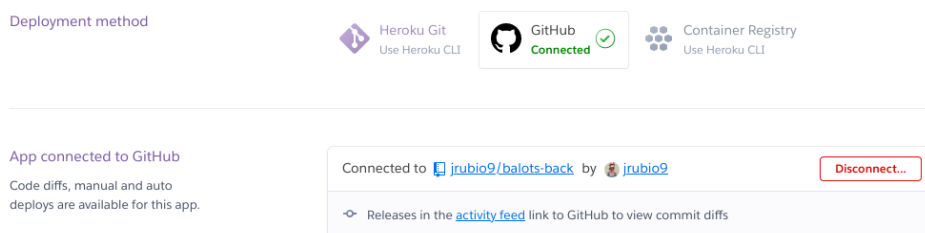
En aquest punt seguirem utilitzant l'aplicació de Heroku (balots-api), ja que ens interessa que la base de dades i l'API estiguin en el mateix contenidor.

Tornem al Dashboard (panell d'usuari) i seleccionem l'opció Deploy:



Aquí vincularem aquesta aplicació amb el nostre repositori a Github.

Això ens permetrà seleccionar una branca determinada i dir-li a Heroku que faci deploy (així de fàcil; al menys amb els llenguatges que Heroku integra per defecte).



Un cop connectats, baixem fins al final de la secció i realitzem el deploy:

### Manual deploy

Deploy the current state of a branch to this app.

### Deploy a GitHub branch

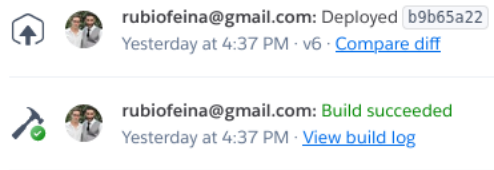
This will deploy the current state of the branch you specify below. [Learn more.](#)

### Choose a branch to deploy

↕ Deploy Branch

Ens apareixerà una consola notificant l'estat, i si tot ha funcionat, ens ho indicarà al acabar.

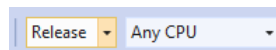
Podem veure un llistat dels deploys que hem anat fent des del panell principal de l'usuari.







### 10.1.3. Back-office

El back-office és l'única aplicació d'escriptori del projecte, i connecta directament amb la base de dades.

Per posar-la en "producció", hem hagut de modificar el connectionString (adreça de connexió amb la base de dades), i compilar l'aplicació amb mode "release".



Això generarà una carpeta amb l'executable, i varis fitxers de configuració i/o llibreries.

|                                                                                                              |                  |                     |        |
|--------------------------------------------------------------------------------------------------------------|------------------|---------------------|--------|
|  Assets                   | 10/06/2021 17:33 | Carpeta de archivos |        |
|  Resources                | 10/06/2021 17:33 | Carpeta de archivos |        |
|  balots-office            | 10/06/2021 17:31 | Aplicación          | 989 KB |
|  balots-office.exe.config | 10/06/2021 17:28 | XML Configuratio... | 23 KB  |

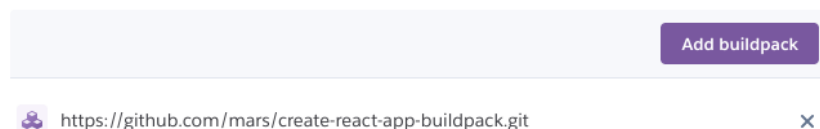
## 10.2. Implantació frontend

Des de Heroku accedirem a l'aplicació "balots" i repetirem el procés realitzat per pujar l'API.

Tot i això, JavaScript (amb React) no és un llenguatge suportat per defecte per Heroku, per tant necessitem indicar un buildpack des de l'apartat de settings.

Els buildpacks són scripts que s'executen al fer deploy de l'aplicació, s'utilitzen per instal·lar dependències necessàries i configurar l'entorn perquè s'adapti a l'aplicació.

Per sort, Heroku ja ofereix un buildpack preparat per aplicacions amb React.



(*GitHub - Nhutphuongit/Create-React-App-Buildpack: Heroku Buildpack for Create-React-App: Static Hosting for React.Js Web Apps, n.d.*)

Un cop configurat el buildpack ja podem seguir els passos de la mateixa forma que hem fet amb l'API.

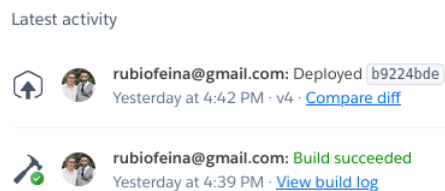
Però encara ens falta un detall.

Abans de pujar-la hem de modificar les direccions guardades dins l'aplicació que apunten a l'API.

Passem de:

- localhost:9090/productes a <https://balots-api.herokuapp.com/api/productes>
- localhost:9090/comandes a <https://balots-api.herokuapp.com/api/comandes>

Després de pujar els canvis a Github, fem el deploy.



En aquest moment ja podem accedir a <https://balots.herokuapp.com> i veurem com es mostra la pàgina correctament (però sense informació).



## 10.3. Ajustaments finals

Faltarà fer una última modificació al projecte d'api i tornar a fer deploy.

Hem de canviar la configuració de política cors perquè permeti realitzar crides des de el DNS balots.herokuapp.com.

```
@CrossOrigin("https://balots.herokuapp.com")
@GetMapping
public Iterable<Producte> obtenirTots() { return productesService.getAll(); }
```

Un cop fet aquest canvi fem deploy, i ja tindrem tot el sistema funcionant.

Ara, cada cop que fem canvis en el codi, podem publicar a Github i fer el deploy manual per actualitzar el servidor.

## 10.4. Resultats

### 10.4.1. Back-office

#### 10.4.1.1. Tasques

| Name                                                                     | Projecte               | Status |
|--------------------------------------------------------------------------|------------------------|--------|
| Pujada d'imatges a AWS S3 i emmagatzemament del link al producte         | Back-office WPF        | To Do  |
| Actualitzar Informació Comandes Regularment                              | Back-office WPF DB     | To Do  |
| [DashboardsPage]                                                         | Back-office WPF        | Doing  |
| [ConfigPage]                                                             | Back-office WPF        | Done   |
| [ProductesPage]                                                          | Back-office WPF        | Done   |
| Afegir la info del client al fer GET de la comanda.                      | Back-office DB         | Done   |
| Comanda Fer que demani el preu final al finalitzar-la                    | Back-office WPF DB     | Done   |
| Error al actualitzar els alérgens d'un producte                          | Back-office WPF DB     | Done   |
| [ModalProducte]                                                          | Back-office WPF        | Done   |
| [ComandesDetails]                                                        | Back-office WPF        | Done   |
| Millorar el modal perquè sigui més reaprofitable                         | Back-office WPF        | Done   |
| Mirar això (OnPropertyChanged)                                           | Back-office            | Done   |
| [ComandesPage]                                                           | Back-office WPF        | Done   |
| Producte hauria de poder incloure descripció (si és pastis, per exemple) | DB Back-office Estils  | Done   |
| Revisar relacions DB                                                     | DB                     | Done   |
| Canvis DB                                                                | DB                     | Done   |
| OJO: Permetre URL de consulta (CORS)                                     | API                    | Done   |
| Funcionalitats API                                                       | API                    | Done   |
| Actualitzar Models Springboot                                            | API                    | Done   |
| MVVM NO FUNCIONA                                                         |                        | Done   |
| [ProductesPage] v1                                                       |                        | Done   |
| Fer MVLiniesComanda com MVComandes (no a fora)                           |                        | Done   |
| MVVM Productes                                                           |                        | Done   |
| [ProductesPage] v2                                                       | WPF Back-office Estils | Futurs |
| Problema: Històric de preus de producte                                  | Back-office DB         | Futurs |
| Millorar estil de Detalls Comanda                                        | Back-office Estils     | Futurs |
| [ComandesPage 2.0]                                                       |                        | Futurs |

COUNT 27

Hem de tenir en compte que moltes d'aquestes tasques contenen sub-tasques.

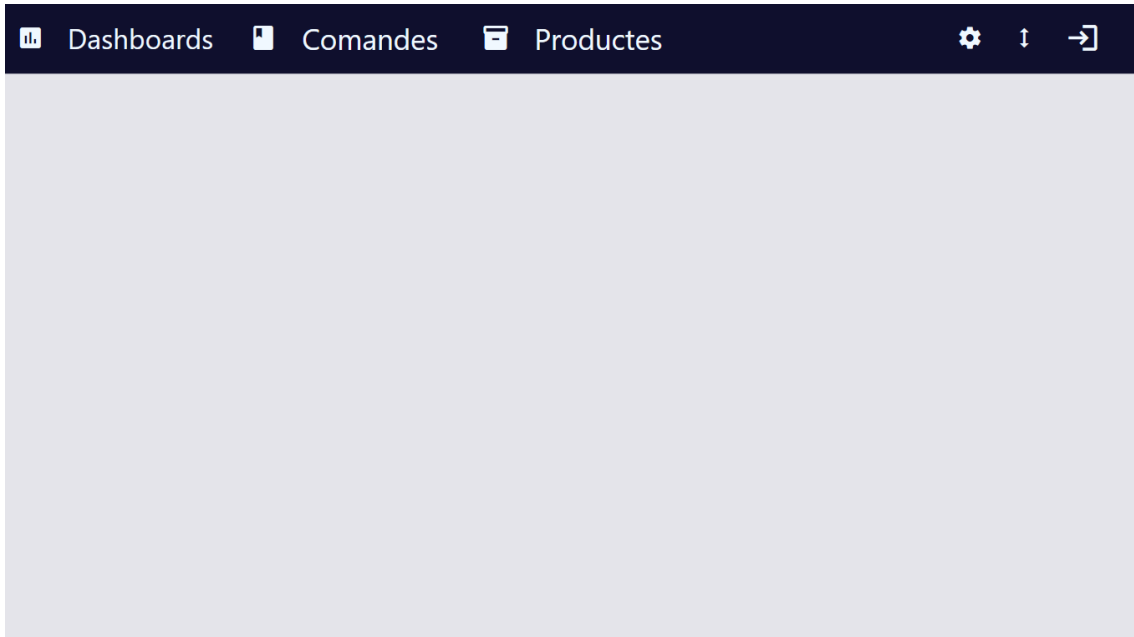
Però com a visió general podem veure que s'han realitzat 20 de les 23 tasques que hi havia (tenint en compte que n'hi ha 4 "Futures" que no entren dins el marc del TFG).

Equival a un ~87% de la feina.

### 10.4.1.2. Resultat implementació

Seguidament desglossarem l'aplicació en components, per tal que el lector tingui una visió clara de l'estat actual del projecte de back-office.

#### Pantalla d'inici






#### Barra de navegació



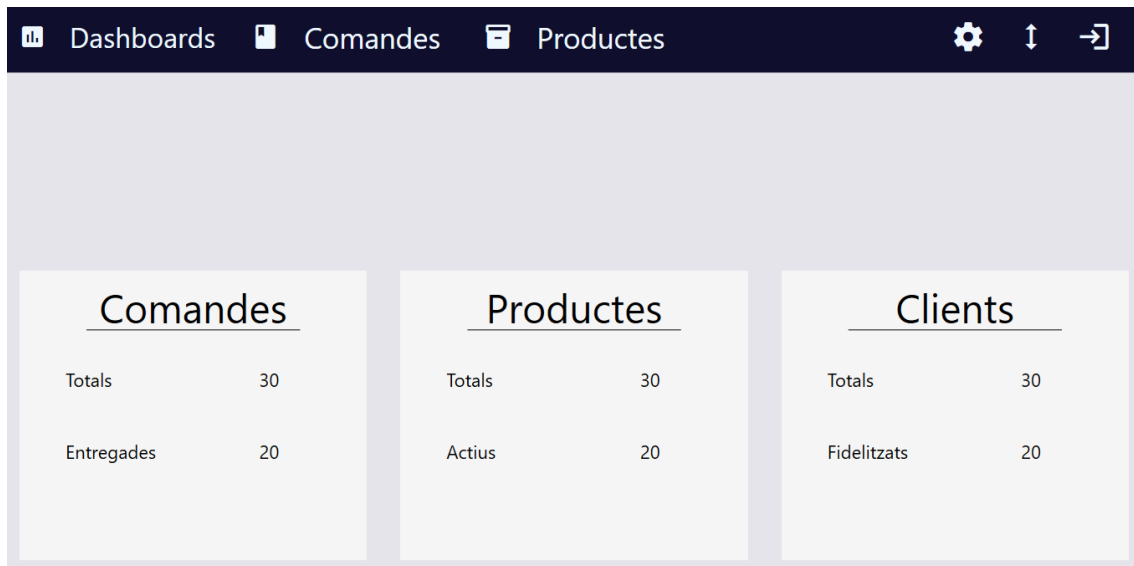
Des d'aquesta barra superior podem accedir als diferents apartats del programa.

La part dreta del menú ens permet realitzar les següents accions:

|                                                                                     |  |                                                       |
|-------------------------------------------------------------------------------------|--|-------------------------------------------------------|
|  |  | Ens permet accedir a les configuracions del programa. |
|  |  | Maximitza l'aplicació o la torna al seu estat normal. |
|  |  | Utilitzant aquesta opció sortim del programa.         |

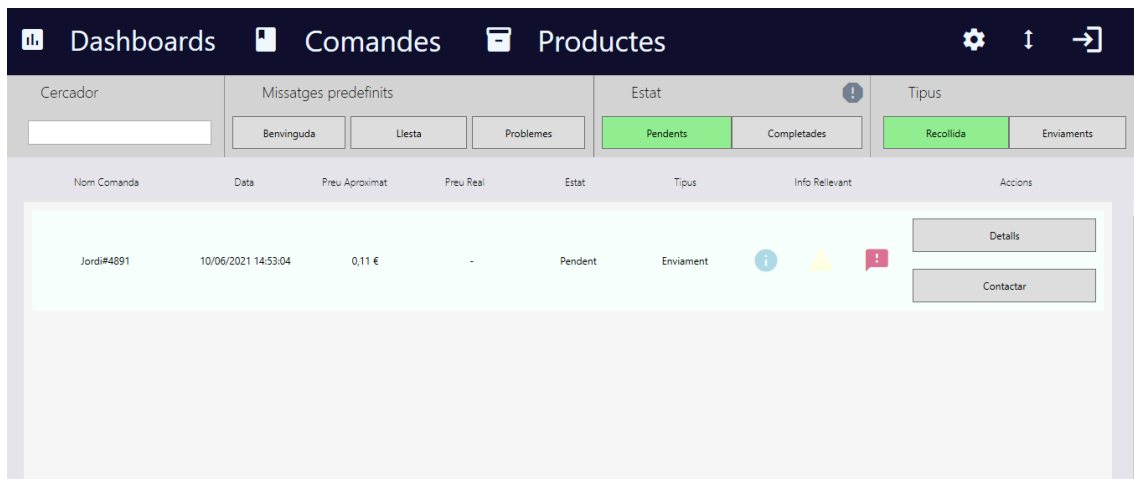
## Mòduls principals

### - Dashboards

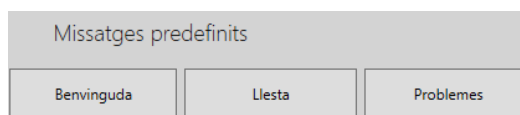


La part de pantalla està preparada, però faltaria connectar-la amb la base de dades.  
(Actualment són estàtics)

### - Comandes



#### ○ Missatges preestablerts



Al fer clic sobre algun dels missatges preestablerts el copia al porta-retalls de Windows.

*\*Aquests missatges es poden personalitzar des de la finestra de configuració.*

## ○ Secció de filtres



Des d'aquesta secció podem seleccionar quins filtres volem habilitar. Al fer-ho es filtrarà el llistat automàticament.

Només se'n pot seleccionar un de cada. [Tipus, Secció]

### Descripció filtres

Estat:

- Pendants → Aquelles comandes que encara no s'han gestionat.
- Completades → Comandes finalitzades
- Error (icona) → Comandes que han tingut algun error.

Tipus:

- Recollida → Comandes que es recolliran a la botiga.
- Enviament → Comandes que s'enviaran.

## ○ Vista d'una comanda dins el llistat

| Nom Comanda | Data                | Preu Aproximat | Preu Real | Estat   | Tipus     | Info Relevant                                                                                                                                                                                                                                                    | Accions                                                |
|-------------|---------------------|----------------|-----------|---------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| Jordi#4891  | 10/06/2021 14:53:04 | 0,11 €         | -         | Pendent | Enviament |    | <button>Details</button><br><button>Contactar</button> |

Ens permet veure els detalls de les diferents comandes de forma ràpida.

El botó veure detalls ens permet accedir a una vista detallada de la comanda que veurem a continuació.

El botó contactar obrirà Whatsapp Desktop (obrint directament el xat amb el contacte, fins i tot si no està registrat), o ens redirigirà a la web de descàrrega si no el tenim instal·lat.

○ **Detalls de la comanda**

Jordi#4891



Pendent → En curs → Finalitzada

Preu aproximat: 0,11 €

Preu final: -

Començar

Error

| Num. | Producte   | Quantitat | Alergens |
|------|------------|-----------|----------|
| 1    | Producte 1 | 9         |          |

Des d'aquí podem veure el llistat de productes que sol·licita el client, així com dades pròpies de la comanda (nom, estat, preu aproximat).

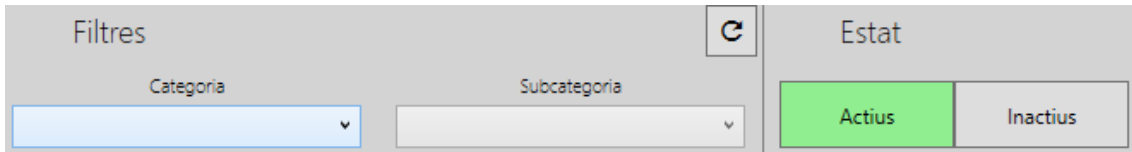
Amb els botons inferiors podem avançar d'estat la comanda, o marcar-la com a comanda amb errors.

Al finalitzar el procés (avançant des de l'estat "en curs" a "finalitzada"), sortirà una finestra emergent que ens demanarà el preu final que ha tingut.

- **Productes**

| Seleccionats | Producte   | Categoria    | Subcategoria | Preu                   | Alergens | Accions           |
|--------------|------------|--------------|--------------|------------------------|----------|-------------------|
|              | Producte 1 | Llaminadures | Xuxes        | 12,50 €/kg<br>0,20 €/u |          | Editar Desactivar |
|              | Producte 2 | Llaminadures | Caramels     | -<br>0,50 €/u          |          | Editar Desactivar |

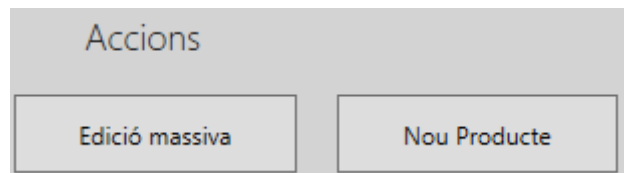
- **Filtres**



En aquest cas podem filtrar per categoria / subcategoria (o combinació de les dues) i per estat del producte.

Actiu implica que es vegi a la web, mentre que al estar inactiu no apareixerà disponible pels clients.

- **Accions generals**



Al fer clic a “Nou Producte” s’obrirà una finestra de creació que veurem més endavant.

L’opció d’edició massiva permet modificar els atributs no únics de tots els productes seleccionats (\*).

- **Vista d’un producte dins el llistat**

- 

| Seleccionats | Producte   | Categoria    | Subcategoria | Preu                   | Al·lèrgens | Accions |
|--------------|------------|--------------|--------------|------------------------|------------|---------|
|              | Producte 1 | Llaminadures | Xuxes        | 12,50 €/kg<br>0,20 €/u |            | <br>    |

Ens permet veure els detalls dels diferents productes de forma ràpida.

El botó “Editar” ens permet accedir a la vista de creació / modificació de productes amb les dades carregades i llestes per ser modificades.

El botó “Desactivar” permet passar ràpidament aquest producte al grup dels inactius.

Per tornar-lo a activar haurem de filtrar el llistat per “Inactius” i fer clic al mateix botó, que s’anomenarà “Activar”.

\* La opció d’edició massiva encara no està disponible.

- **Finestra de creació / modificació de producte**

Aquesta finestra ens permet omplir en la creació, o modificar durant l'edició, la informació sobre un producte.

- Selecció de la imatge (\*)

Al fer clic sobre la imatge s'obrirà un selector de Windows que ens demanarà un fitxer. Al seleccionar-lo pujarà la imatge al servidor i quedarà vinculada amb aquest producte.

*\* La selecció d'imatges encara no està disponible.*



- **Configuració**

Configuració ?

|                                                                    |                                                                                                                                                 |
|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Categories</p> <p>Laminadures</p> <p>Afegir</p> <p>Eliminar</p> | <p>Subcategories</p> <p>Xuxes Laminadures</p> <p>Caramels Laminadures</p> <p>Afegir</p> <p>Eliminar</p>                                         |
| <p>Al·lèrgens</p> <p>Afegir</p> <p>Eliminar</p>                    | <p>Missatges Predefinitos</p> <p>Benvinguda: <input type="text"/></p> <p>Lista: <input type="text"/></p> <p>Problemes: <input type="text"/></p> |

Aquesta pàgina ens permet crear categories, al·lèrgens, subcategories, i modificar missatges preestablerts.

## 10.4.2. Front-office

### 10.4.2.1. Tasques

| Name                                                       | Status      | Priority |
|------------------------------------------------------------|-------------|----------|
| Millorar el poc Copywriting que hi ha                      | Post-TFG    | Medium   |
| Scroll més guapo (a alguns navegadors)                     | Post-TFG    | Low      |
| Pagament via Stripe                                        | Post-TFG    | Low      |
| Millorar footer                                            | Post-TFG    | Low      |
| Analytics                                                  | Post-TFG    | Low      |
| Millorar estils Checkout                                   | Post-TFG    |          |
| TAGs a productes                                           | Post-TFG    |          |
| Multiples fotos d'un producte                              | Post-TFG    |          |
| HTML Font size a 62.5%                                     | Post-TFG    |          |
| Navbar + Cart                                              | Completed   | High 🔥   |
| Comunicació amb l'API                                      | Completed   | High 🔥   |
| [Dades] Store                                              | Completed   | High 🔥   |
| ProductsPage (home)                                        | Completed   | High 🔥   |
| Actualitzar els packages                                   | Completed   | High 🔥   |
| Media Queries Bàsiques                                     | Completed   | Medium   |
| Checkout                                                   | Completed   | Medium   |
| Unificar Estils                                            | Completed   | Low      |
| Overflow: scroll a Tabs i/o Scrollable Tabs                | Completed   |          |
| Mirar que faci bé el càlcul del preu estimat               | Completed   |          |
| Fer una pàgina com producttabs però genèrica per la pàgina | In Progress | Medium   |
| Vista d'al·lèrgens des de la targeta del producte          | Next Up     | Medium   |
| PastissosPage                                              | Next Up     | Medium   |
| BalotsBoxPage                                              | Next Up     | Medium   |
| Filtres                                                    |             | Medium   |
| Redisseny del Checkout                                     |             | Low      |
| Label amb el nom de la Categoria en la que estem al co     |             | Low      |
| + New                                                      |             |          |

COUNT 26

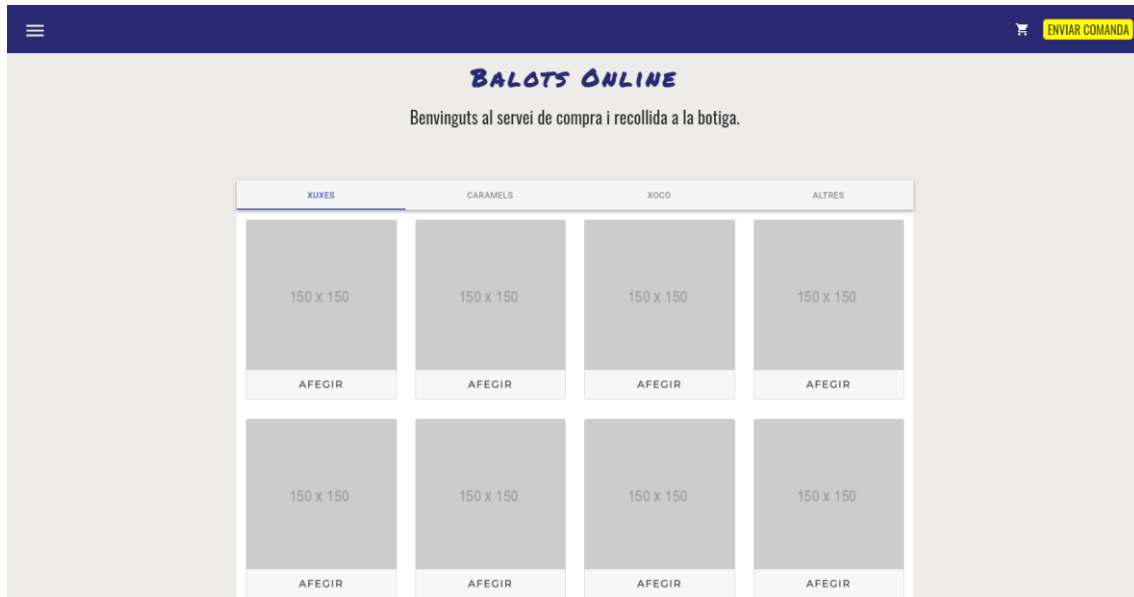
Hem de tenir en compte que moltes d'aquestes tasques contenen sub-tasques.

Però com a visió general podem veure que s'han realitzat 10 de les 17 tasques que hi havia (tenint en compte que n'hi ha 9 "Futures" que no entren dins el marc del TFG).

Equival a un ~60% de la feina.

### 10.4.2.2. Resultat d'implementació

#### Pàgina inicial (home)



#### Barra de navegació

##### Menú de navegació

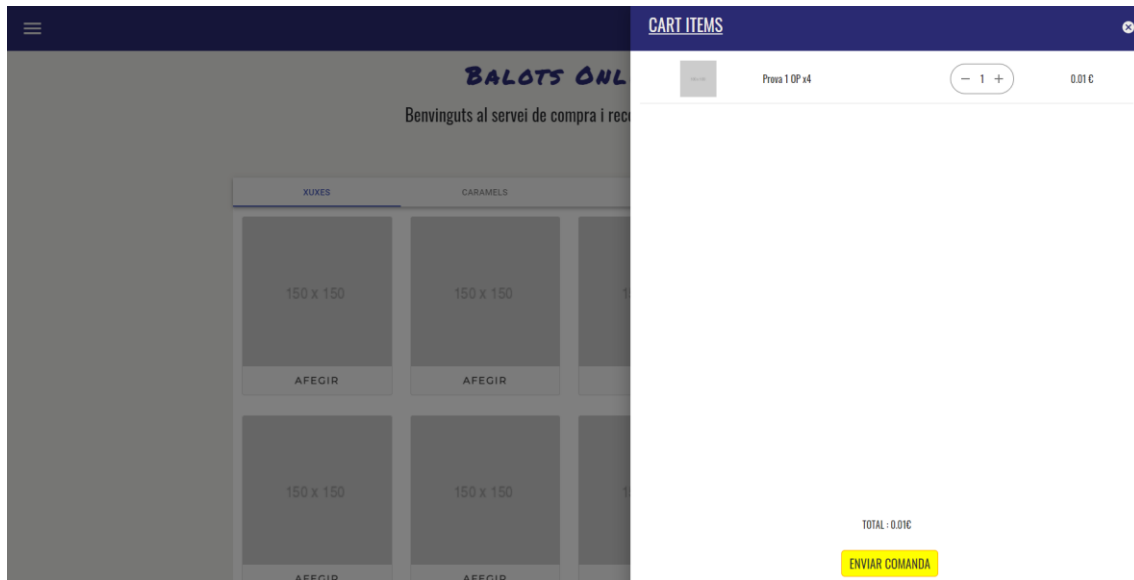
Al fer clic a la icona de la part esquerra de la barra de navegació s'obrirà un menú lateral amb les pàgines disponibles.



## Cistella de compra

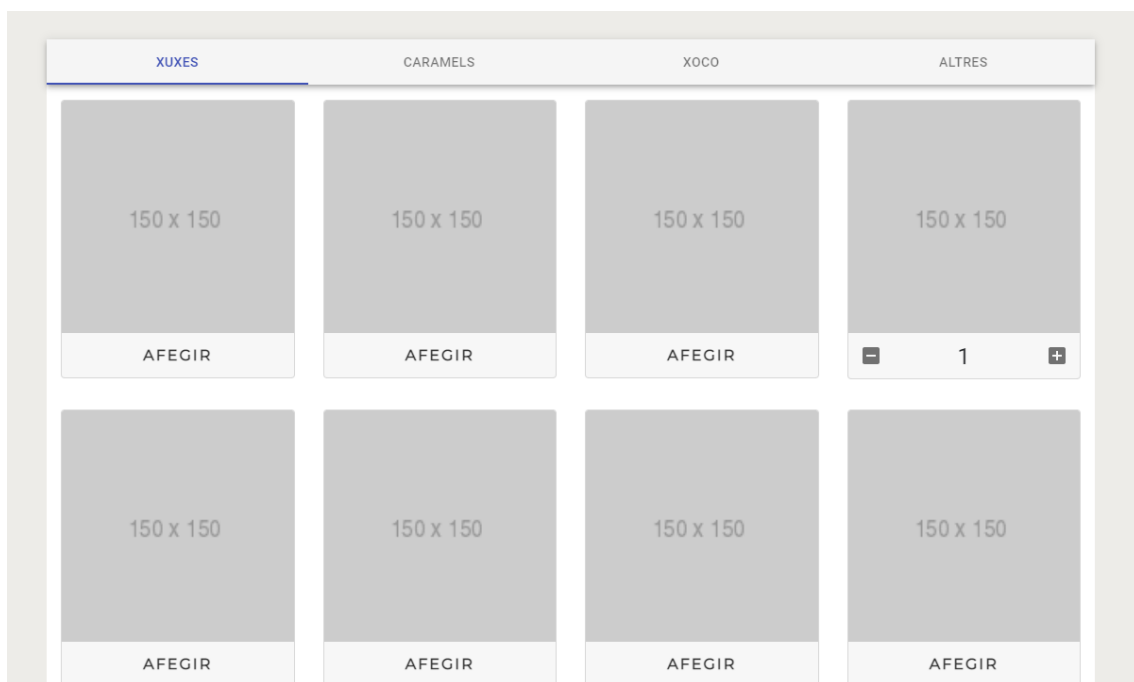
Al fer clic sobre la icona de la cistella a la part dreta de la barra de navegació, s'obrirà aquesta visualització dels productes seleccionats.

Des d'aquí es podrà interactuar amb la cistella.



## Pàgines productes

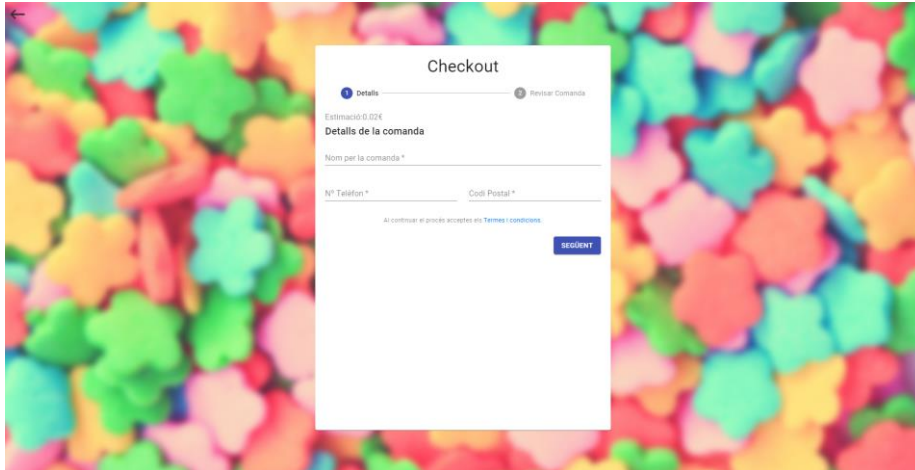
Les pàgines de productes consten d'una taula central amb targetes dels productes disponibles. (De moment només hi és a la pàgina de "Productes", la inicial)



## Checkout

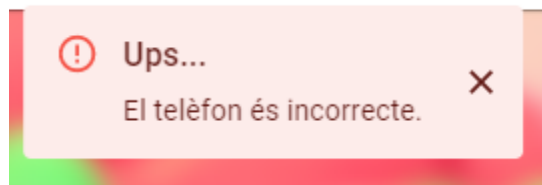
La primera pàgina del checkout demana a l'usuari certes dades bàsiques per realitzar la comanda.

A la part superior esquerra tenim una icona que ens permet tornar al lloc web (hauria de ser més visible).

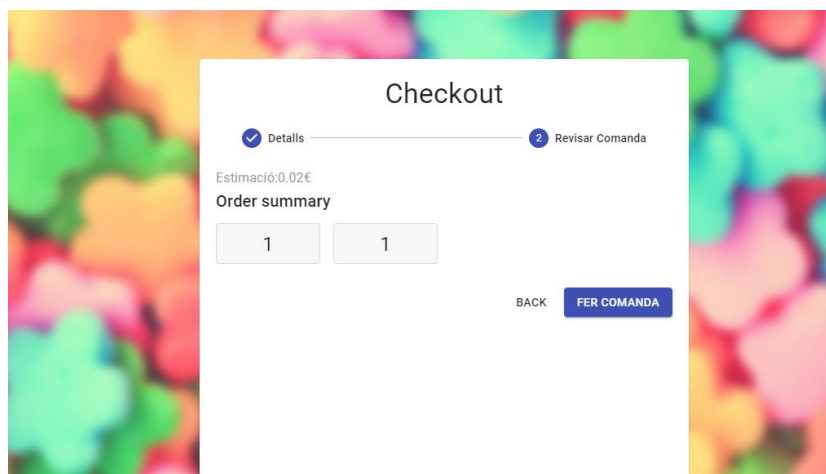


The screenshot shows a mobile checkout form titled 'Checkout'. At the top, there is a progress bar with two steps: '1 Detalls' (active) and '2 Revisar Comanda'. Below the progress bar, the estimated amount is 'Estimació: 0.02€'. The section is titled 'Detalls de la comanda'. It contains three input fields: 'Nom per la comanda \*', 'Nº Telèfon \*', and 'Codi Postal \*'. Below these fields, there is a small link: 'Al continuar el procés acceptes els Termes i condicions.' and a blue button labeled 'SEGUIR'. A back arrow is visible in the top left corner of the screen.

Al introduir malament alguna de les dades necessàries, la pàgina ens notificarà amb un error.

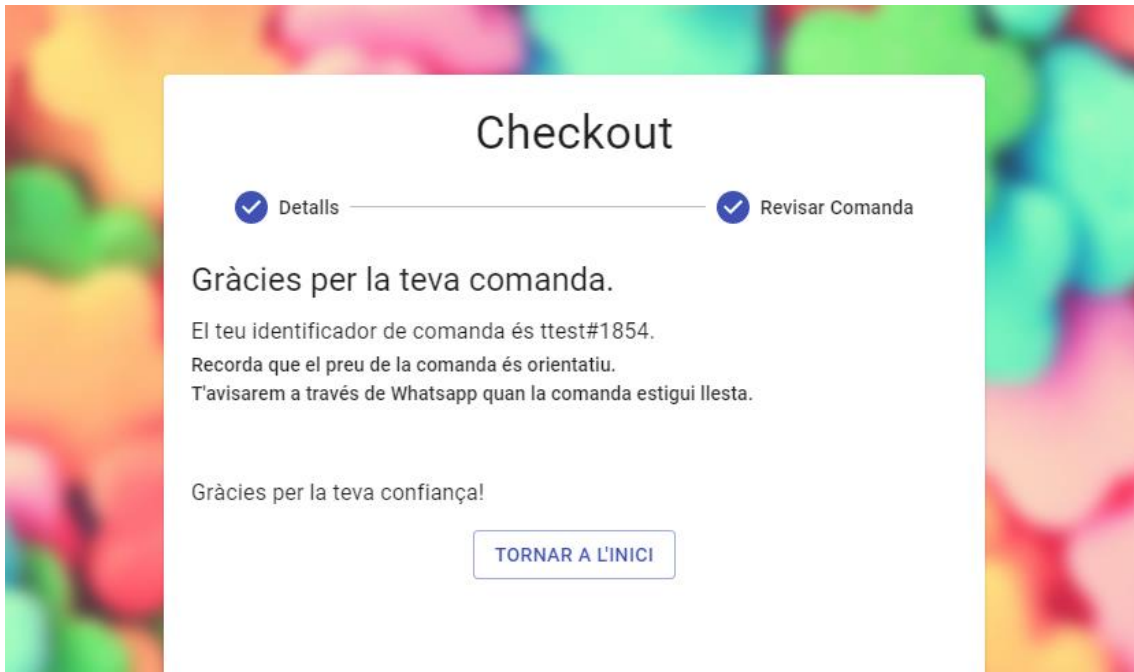


Al passar a la següent fase de compra ens mostrarà un llistat amb els productes sol·licitats i les seves quantitats (haig de fer que mostri la imatge).



The screenshot shows the checkout form at the 'Revisar Comanda' step. The progress bar now shows '1 Detalls' (completed) and '2 Revisar Comanda' (active). The estimated amount remains 'Estimació: 0.02€'. The section is titled 'Order summary'. It displays two items, each with a quantity of '1' in a grey box. At the bottom right, there are two buttons: a grey 'BACK' button and a blue 'FER COMANDA' button.

Al finalitzar la comanda, informarà al client que ja ha estat enviada.



# 11. Conclusions

Construir una aplicació que utilitza diferents serveis no és una tasca simple. I utilitzar llenguatges que estàs aprenent potser no és la millor decisió en un projecte del món laboral; però en el context d'aquest treball m'ha servit molt.

He vist i après tecnologies que feia temps que volia utilitzar.

I el camí només acaba de començar, la meva intenció és continuar desenvolupant aquest lloc web, com podrem veure a l'apartat 12.

Després de presentar el full de projecte vaig canviar de feina, passant de 6h en remot a 8h presencials. Per desgràcia això ha afectat al compliment de les tasques proposades inicialment.

Vaig subestimar la tasca de buscar informació i aprendre unes bases mínimament sòlides dels llenguatges que volia utilitzar.

També em vaig saturar una mica amb la varietat de formes de realitzar el mateix amb Spring, sobretot amb la part de seguretat.

Aquests són els motius pels quals la part de back-office, que inicialment volia que fos web, i utilitzés l'API, finalment va ser un programa d'escriptori que connecta directament amb la base de dades. WPF és una eina amb la que he treballat professionalment i vaig pensar que fer aquest canvi era l'única forma d'arribar als objectius que m'havia marcat.

En següents versions unificaré la web, permetent accedir al back-office des d'un subdomini, i utilitzant l'API per comunicar-me amb la base de dades.

Després de comentar aquests aspectes "negatius", vull afegir que considero que he prioritzat correctament les tasques, i que la major part de tasques importants estan realitzades.

Com ja hem comentat anteriorment, si ens basem en les tasques definides durant l'anàlisi podem extreure els següents percentatges d'acompliment:

**Front-office: 60%**

**Back-office: 89%**

També vull puntualitzar que estic molt satisfet amb el resultat de la part de front-office, tot i que encara quedi bastant per fer.

És la part que veuran els clients, i per tant "la que ha de rebre més amor".

Un últim punt a destacar és que tinc una versió MVP totalment funcional i disponible pels clients (<https://can-balots.web.app>), i una versió de prova amb els canvis aplicats durant el projecte final.

M'interessava que això estigués fet perquè el tribunal pugui provar totes les parts de l'aplicació.

Per accedir al front-office, utilitzar l'enllaç: <https://balots.herokuapp.com>

Per obrir el programa de back-office: Seguir les instruccions de l'apartat 14.

Cal destacar que al ser un pla gratuït de Heroku, la web entra en repòs després d'una estona sense visites, i degut a això l'usuari que hi accedeix per primer cop quan està en repòs, experimentarà una càrrega de la pàgina bastant lenta.

Per acabar, una mica d'informació sobre les hores (aproximades) invertides a cada part, sense contar la investigació i proves:

- Backoffice : 58h
- API : 16h
- front-office : 36h
- Deployment: 8h



## 12. Treball futur

Tot i tenir una base sòlida per avançar, encara queden bastants punts a implementar abans de poder substituir la versió actual per aquesta a producció:

### Back-office

Aquesta part seria ideal passar-la a web, però d'entrada acabaré de millorar l'aplicació actual.

Tenint en compte que està pràcticament llesta, no seria gaire intel·ligent tornar-ho a començar tot amb una versió web.

| Name                                                             | Projecte               | Status |
|------------------------------------------------------------------|------------------------|--------|
| Pujada d'imatges a AWS S3 i emmagatzemament del link al producte | Back-office WPF        | To Do  |
| Actualitzar Informació Comandes Regularment                      | Back-office WPF DB     | To Do  |
| [DashboardsPage]                                                 | Back-office WPF        | Doing  |
| [ProductesPage] v2                                               | WPF Back-office Estils | Futurs |
| Problema: Històric de preus de producte                          | Back-office DB         | Futurs |
| Millorar estil de Detalls Comanda                                | Back-office Estils     | Futurs |
| [ComandesPage 2.0]                                               |                        | Futurs |

+ New

COUNT 7

### Front-office

Aquesta serà la part que prioritzaré, deixant un producte acabat i polit per una bona experiència d'ús del client.

| Name                                                                              | Status      | Priority |
|-----------------------------------------------------------------------------------|-------------|----------|
| Millorar el poc Copywriting que hi ha                                             | Post-TFG    | Medium   |
| Scroll més guapo (a alguns navegadors)                                            | Post-TFG    | Low      |
| Pagament via Stripe                                                               | Post-TFG    | Low      |
| Millorar footer                                                                   | Post-TFG    | Low      |
| Analytics                                                                         | Post-TFG    | Low      |
| Millorar estils Checkout                                                          | Post-TFG    |          |
| TAGs a productes                                                                  | Post-TFG    |          |
| Multiples fotos d'un producte                                                     | Post-TFG    |          |
| HTML Font size a 62.5%                                                            | Post-TFG    |          |
| Fer una pàgina com producttabs però genèrica per la part de pastissos i balotsbox | In Progress | Medium   |
| Vista d'al·lèrgens des de la targeta del producte                                 | Next Up     | Medium   |
| PastissosPage                                                                     | Next Up     | Medium   |
| BalotsBoxPage                                                                     | Next Up     | Medium   |
| Filtres                                                                           |             | Medium   |
| Redisseny del Checkout                                                            |             | Low      |
| Label amb el nom de la Categoria en la que estem al costat del DrawerIcon         |             | Low      |

+ New

COUNT 16

## 13. Bibliografía

Ndaw, Ibrahima (2020). *Redux VS React Context: Which one should you choose?*. [Consulta: 23 d'abril de 2021]. <https://dev.to/ibrahima92/redux-vs-react-context-which-one-should-you-choose-2hhh>

Hall, Mars (2016). *Deploying React with Zero Configuration* [Consulta: 3 de juny de 2021]. <https://blog.heroku.com/deploying-react-with-zero-configuration>

Rodríguez García, Juan David (s.d.). *Programación asíncrona* [Consulta: 13 de maig de 2021]. <https://programamos.es/unidad-7-programacion-asincrona/>

Corzo, Giancarlo (2017). *Programación Asíncrona* .[Consulta: 13 de maig de 2021]. <https://medium.com/laboratoria-how-to/programacion-asincrona-cea3bad7c3c6>

Chanphakeo, Robert (2017). *Getting Started with Entity Framework Core (PostgreSQL)* [Consulta: 22 de març de 2021]. <https://medium.com/@RobertKhou/getting-started-with-entity-framework-core-postgresql-c6fa09681624>

Orús, Abigail (2020, desembre 9). *Número de usuarios de Internet que compraron online en España en 2020, por grupos de edad*. [Consulta: 3 de juny de 2021]. <https://es.statista.com/estadisticas/496509/porcentaje-de-compradores-online-por-edad-espana/>

Orús, Abigail (2021, març 10). *Productos adquiridos a través de internet por los consumidores online en España durante 2020, por categoría*. [Consulta: 3 de juny de 2021]. <https://es.statista.com/estadisticas/499435/productos-online-comprados-online-espana/>

Segura Aguilar, Paula (2021, febrer 2). *El eCommerce en España - Edición 2021*. [Consulta: 3 de juny de 2021]. <https://business.trustedshops.es/blog/e-commerce-espania/>

García Romeralo, Puri (2019, maig 29). *Tipos de APIs*. [Consulta: 12 de març de 2021]. <https://solidgargroup.com/tipos-de-apis/>

Rodríguez Patiño, Eduardo (2019, desembre 6). *¿QUÉ ES LA PROGRAMACIÓN ASÍNCRONA Y SÍNCRONA?*. [Consulta: 28 d'abril de 2021]. <https://anexsoft.com/que-es-la-programacion-asincrona-y-sincrona>

Yamazaki, Shiori (2020, agost 18). *Understanding Functional Components vs. Class Components in React* [Consulta: 23 d'abril de 2021]. <https://www.twilio.com/blog/react-choose-functional-components>

Zaharie, Marius (2020, octubre 13). *React route refresh without page reload* [Consulta: 23 d'abril de 2021]. <https://dev.to/zbmarius/react-route-refresh-without-page-reload-1907>

*Elogia* (s.d.). [Consulta: 22 de maig de 2021]. <https://elogia.net/estudio-covid19-consumo-online-ecommerce/>

*Github* (s.d.). [Consulta: 5 de juny de 2021]. <https://github.com/nhutphuongit/create-react-app-buildpack>

*Github* (s.d.). [Consulta: 5 de juny de 2021]. <https://github.com/mars/create-react-app-buildpack#usage>

*Material UI* (s.d.). [Consulta: 3 de maig de 2021]. <https://material-ui.com/api/list/>

*CSS Tricks* (s.d.). [Consulta: 5 de maig de 2021]. <https://css-tricks.com/almanac/properties/g/grid-rows-columns/>

*Blendo* (s.d.). [Consulta: 3 de juny de 2021]. <https://www.blendo.co/documents/setup-heroku-postgres/>

*Heroku Dev Center* (2020, maig 27). [Consulta: 5 de juny de 2021]. <https://devcenter.heroku.com/articles/deploying-spring-boot-apps-to-heroku>

*Heroku Dev Center* (2020, maig 27). [Consulta: 3 de juny de 2021]. <https://devcenter.heroku.com/articles/heroku-postgresql-credentials>

*Institut Europeu de postgrau* (s.d.). [Consulta: 27 de maig de 2021]. <https://www.iep.edu.es/que-es-el-ecommerce/>

*Lluvia Digital* (2020, febrer 17). [Consulta: 27 de maig de 2021]. <https://www.lluviadigital.com/e-commerce-que-es-y-como-empezar/>

*Santander Becas* (d.). [Consulta: 29 de maig de 2021]. <https://blog.becas-santander.com/es/metodologias-desarrollo-software.html>

*Kanbanize* (s.d.). [Consulta: 29 de maig de 2021]. <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban>

*RedHat* (s.d.). [Consulta: 25 de maig de 2021]. <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>

*MDN Web Docs* (s.d.). [Consulta: 1 de maig de 2021]. [https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/async\\_function](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/async_function)

*MDN Web Docs* (s.d.). [Consulta: 14 de maig de 2021]. [https://developer.mozilla.org/es/docs/Web/CSS/Media\\_Queries/Using\\_media\\_queries](https://developer.mozilla.org/es/docs/Web/CSS/Media_Queries/Using_media_queries)

*Reactjs Docs* (s.d.). [Consulta: 3 de maig de 2021]. <https://reactjs.org/docs/components-and-props.html>

*Reactjs Docs* (s.d.). [Consulta: 3 de maig de 2021]. <https://reactjs.org/docs/hooks-reference.html>

*JavaPoint* (s.d.). [Consulta: 3 de maig de 2021]. <https://www.javatpoint.com/react-hooks>

*React Router* (s.d.). [Consulta: 13 de maig de 2021]. <https://reactrouter.com/web/guides/quick-start>

*Educative.io* (s.d.). [Consulta: 13 de maig de 2021]. <https://www.educative.io/edpresso/what-is-a-react-router>

*JetBrains* (s.d.). [Consulta: 1 de juny de 2021]. <https://www.jetbrains.com/help/idea/class-diagram-toolbar-and-context-menu.html#toolbar>

*Npgsql* (s.d.). [Consulta: 22 de març de 2021]. <https://www.npgsql.org/doc/basic-usage.html>

*Npgsql* (s.d.). [Consulta: 22 de març de 2021]. <https://www.npgsql.org/doc/security.html>

*Npgsql* (s.d.). [Consulta: 22 de març de 2021]. <https://www.npgsql.org/doc/types/basic.html>

*PostgreSQL* (s.d.). [Consulta: 24 de març de 2021]. <https://www.postgresql.org/docs/9.1/libpq-ssl.html>

*BezKoder* (2021, juny 2). [Consulta: 12 de març de 2021]. <https://bezcoder.com/spring-boot-security-postgresql-jwt-authentication/>

*PostgreSQL* (s.d.). [Consulta: 10 de març de 2021]. <https://www.postgresql.org/docs/9.1/libpq-ssl.html>

Maksim Ivanov. (2020, maig 20). *HOW TO STRUCTURE YOUR REACT APPLICATIONS*. [Vídeo]. <https://www.youtube.com/watch?v=UfoM49p3WCw>

Java Guides. (2020, novembre 28). *Create and Deploy React App on Heroku - Step by Step Guide* [Vídeo]. <https://www.youtube.com/watch?v=tA4Fl1dZgjc>

Material-ui & TouchTap: Getting key of ListItem (2016, juny 22). A *Stackoverflow*. 15 de maig de 2021 <https://stackoverflow.com/questions/32457480/material-ui-touchtap-getting-key-of-listitem>

How to round to at most 2 decimal places, if necessary? (2012, agost 6). A *Stackoverflow*. 15 de maig de 2021 <https://stackoverflow.com/questions/11832914/how-to-round-to-at-most-2-decimal-places-if-necessary>

Versionado de software (s.d.). En *Wikipedia*. 22 de març de 2021 [https://es.wikipedia.org/wiki/Versionado\\_de\\_software](https://es.wikipedia.org/wiki/Versionado_de_software)

# 14. Manual d'instal·lació

El back-office és la única part del projecte que requereix tenir el programa en local.

S'ha generat en format portable. Per tant no és necessari realitzar cap instal·lació al sistema.

Simplement s'ha de descomprimir el fitxer "portable.zip". Al fer-ho es generarà una carpeta anomenada "balots-back-office".

Dins aquesta carpeta podrem trobar-hi els següents fitxers.

|                                   |                  |                       |          |
|-----------------------------------|------------------|-----------------------|----------|
| Assets                            | 10/06/2021 17:33 | Carpeta de archivos   |          |
| Resources                         | 10/06/2021 17:33 | Carpeta de archivos   |          |
| balots-office                     | 10/06/2021 17:31 | Aplicación            | 989 KB   |
| balots-office.exe.config          | 10/06/2021 17:28 | XML Configuration...  | 23 KB    |
| balots-office.pdb                 | 10/06/2021 17:31 | Program Debug D...    | 378 KB   |
| EntityFramework.dll               | 16/04/2020 20:38 | Extensión de la ap... | 4.875 KB |
| EntityFramework.SqlServer.dll     | 16/04/2020 20:38 | Extensión de la ap... | 578 KB   |
| EntityFramework.SqlServer         | 16/04/2020 20:38 | Documento XML         | 160 KB   |
| EntityFramework                   | 16/04/2020 20:38 | Documento XML         | 3.651 KB |
| EntityFramework6.Npgsql.dll       | 17/02/2020 12:12 | Extensión de la ap... | 188 KB   |
| EntityFramework6.Npgsql           | 17/02/2020 12:12 | Documento XML         | 61 KB    |
| MaterialDesignColors.dll          | 13/02/2021 5:46  | Extensión de la ap... | 295 KB   |
| MaterialDesignColors.pdb          | 13/02/2021 5:46  | Program Debug D...    | 18 KB    |
| MaterialDesignThemes.Wpf.dll      | 13/02/2021 5:46  | Extensión de la ap... | 7.980 KB |
| MaterialDesignThemes.Wpf.pdb      | 13/02/2021 5:46  | Program Debug D...    | 99 KB    |
| MaterialDesignThemes.Wpf          | 13/02/2021 5:46  | Documento XML         | 101 KB   |
| Microsoft.Bcl.AsyncInterfaces.dll | 19/10/2020 18:40 | Extensión de la ap... | 21 KB    |
| Microsoft.Bcl.AsyncInterfaces     | 19/10/2020 18:40 | Documento XML         | 18 KB    |
| Microsoft.Win32.Primitives.dll    | 08/07/2020 20:38 | Extensión de la ap... | 21 KB    |
| netstandard.dll                   | 08/07/2020 20:38 | Extensión de la ap... | 97 KB    |
| Npgsql.dll                        | 25/03/2021 11:16 | Extensión de la ap... | 887 KB   |

L'executable balots-office és el que farem servir per iniciar el programa.

## Recomanació

Generar un accés directe a l'escriptori:

Per fer-ho fem clic dret sobre l'executable → Enviar a → Escriptori.