



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica

Universitat de Barcelona

**PLATAFORMA PER A AGRUPACIONS DE
COMERÇOS LOCALS**

Carles Magallón i Triola

Director: Josep Vañó Chic

Realitzat a:

Departament de Matemàtiques i
Informàtica

Barcelona, 26 de Juliol de 2019

Agraïments

Vull agrair al meu tutor Josep Vañó Chic, la dedicació, els consells i el recolzament que m'ha donat durant la realització d'aquest treball.

A tot el professorat del grau d'Enginyeria Informàtica, per facilitar-me l'aprenentatge de tots els coneixements que han fet possible aquest projecte.

Gràcies a la meva família i a la meva parella per creure en mi i donar-me el seu suport incondicional, han estat una peça clau per arribar a on sóc.

Resum

Aquest projecte consisteix en l'anàlisi, el disseny i la implementació d'un lloc web que pretén assentar les bases d'una plataforma per a les agrupacions de comerços locals i als comerços que la formen. Es busca que puguin tenir un lloc web sense molt esforç ni dedicació.

La plataforma s'ha implementat amb el *framework Django* de *Python* al *backend* i al *frontend HTML* i *Bootstrap*.

La plataforma permetrà a l'agrupació tenir un lloc web on donar-se a conèixer als clients i on s'hi recolliran les dades dels comerços que la formen. També permetrà a cada comerç tindre una àrea on exposar als clients la seva informació, actualitzacions i productes.

Hi ha diferents àrees de gestió dins l'aplicació per tal que es puguin modificar i actualitzar les dades. Aquestes àrees les podem dividir entre: la de l'agrupació i la de un comerç. Cal destacar que l'accés, la visualització i la modificació de les dades que s'hi pot trobar a aquestes àrees està restringida en funció dels permisos d'accés que tingui l'usuari.

El disseny de la interfície és *responsive*, és a dir que s'adapta segons el dispositiu des d'on s'està visualitzant. Per això l'aplicació permet ser utilitzada des de qualsevol dispositiu sense perdre funcionalitats.

La finalitat del projecte ha estat conèixer amb més profunditat un *web-framework* molt utilitzat avui en dia. A més de poder aprendre i experimentar com implementar un lloc web des de zero.

Resum (Castellà)

Este proyecto consiste en el análisis, el diseño y la implementación de un sitio web que pretende sentar las bases de una plataforma para las agrupaciones de comercio local y los comercios que la forman. Se busca que puedan tener un sitio web sin mucho esfuerzo ni dedicación.

La plataforma se ha implementado con el *framework Django* de *Python* en el *backend* y en el *frontend HTML* y *Bootstrap*.

La plataforma permitirá a la agrupación tener un sitio web donde darse a conocer a los clientes y donde se recogen los datos de los comercios que la forman. También permitirá a cada comercio tener un área donde exponer a los clientes su información actualizaciones y productos.

Hay diferentes áreas de gestión dentro de la aplicación para que se puedan modificar y actualizar los datos. Estas áreas las podemos dividir entre: la de la agrupación y la de un comercio. Hace falta destacar que el acceso, la visualización y la modificación de los datos que se pueden encontrar en estas áreas estará restringida en función de los permisos de acceso que tenga el usuario.

El diseño de la interface es *responsive*, es decir que se adapta según el dispositivo desde donde se está visualizando. Por eso la aplicación permite ser usada desde cualquier dispositivo sin perder funcionalidades.

La finalidad del proyecto ha sido conocer con más profundidad un *web-framework* muy usado hoy en día. También poder aprender y experimentar como implementar un sitio web desde cero.

Resum (anglès)

The aim of this project is the analysis, design and implementation of a website to settle the basis of an Internet platform addressed to the local trade. Hence, it looks for the reduction of effort and dedication to have a webpage that those businesses hold.

The platform has been implemented using the web *framework Django* of *Python* for the backend and for the frontend *HTML* and *Bootstrap*.

The platform will allow the commercial association to have a webpage to be made known by the customers and collect all their data. It will also permit each business to have an area to expose all its information, updates and products geared towards the customers.

There are multiple management areas within the application where data can be modified and updated. Moreover, there is a section for the trade association and another for each trade. It has to be emphasized that the access, the visualization and modification of data is restricted to the entrance permissions of each user.

The design of the interface is responsive, that is, it adapts to the device from which the webpage is consulted. Thus, the app allows being used from any device without losing functionalities.

The project's purpose is to deeply know a web-framework which is greatly used in the nowadays society. Meanwhile, learnings about how to create a webpage from scratch will be done in this paper.

Índex

Resum	3
Resum (Castellà)	4
Resum (anglès)	5
Índex de figures	9
1. Introducció.....	11
1.1 Motivacions	11
1.2 Descripció del projecte	11
1.3 Objectius.....	12
1.3.1 Objectius generals	12
1.3.2 Objectius específics	12
1.4 Mètode de planificació temporal	12
1.5 <i>Sprints</i>	13
1.5.1 <i>Sprint</i> 1.....	13
1.5.2 <i>Sprint</i> 2.....	13
1.5.3 <i>Sprint</i> 3.....	13
1.5.4 <i>Sprint</i> 4.....	14
1.5.5 <i>Sprint</i> 5.....	14
1.5.6 <i>Sprint</i> 6.....	14
1.5.7 <i>Sprint</i> 7.....	14
1.5.8 <i>Sprint</i> 8.....	15
1.5.9 <i>Sprint</i> 9.....	15
1.6 Dates de lliurament	15
2 Anàlisi.....	16
2.1 Anàlisi i definició de les funcionalitats	16
2.2 Diagrama de casos d'ús	16
2.3 Descripció dels casos d'us textuals.....	17
3 Disseny.....	27
3.1 Arquitectura.....	27
<i>Model</i>	27
<i>Template</i>	27
<i>View</i>	27
Fitxer <i>urls.py</i>	27

3.2 Diagrama de classes	27
3.3 <i>Templates</i>	28
3.3.1 Perfil de l'agrupació	30
3.3.2 Directori de comerços	31
3.3.3 Catàleg de l'agrupació	31
3.3.4 Pàgina d'esdeveniments de l'agrupació	32
3.3.5 <i>Dashboard</i> de l'agrupació	33
3.3.6 <i>Feed</i> de notícies de l'agrupació	34
3.3.7 Perfil del comerç	35
3.3.8 Catàleg del comerç	36
3.3.9 Pàgina d'esdeveniments del comerç	36
3.3.10 <i>Dashboard</i> del comerç	36
3.3.11 <i>Feed</i> de notícies del comerç	37
3.3.12 Pàgina de la notícia	37
3.3.13 Pàgina d'un producte	37
3.3.14 Pàgina del esdeveniment	38
3.3.15 <i>Templates</i> de formularis	39
4 Implementació	40
4.1 Organització dels fitxers	40
4.1.1 Directori <i>TFG</i>	40
4.1.2 Directori <i>elocalcommerce</i>	40
4.1.3 Directori <i>templates</i>	41
4.1.4 Directori <i>media</i>	41
4.2 Model	41
4.2.1 Classes del model	41
4.2.2 <i>ForeignKey</i> . Relació <i>Many-to-one</i>	42
4.2.3 <i>OneToOneField</i> . Relació <i>One-to-one</i>	42
4.2.4 Consultes al model	43
4.3 Formularis de <i>Django</i>	43
4.4 Gestió de les imatges pujades	44
4.5 Interfície	45
4.5.1 <i>HTML</i> i <i>Bootstrap</i>	45
4.5.2 Filtres i etiquetes de templates	46

4.5.3 Llibreria <i>Crispy-Forms</i>	47
4.6 <i>Views</i> , la lògica de l'aplicació.....	49
4.7 <i>URLs</i>	49
4.8 Permisos	50
5. Proves i resultats	52
5.1 Avaluacions del <i>sprints</i>	52
6. Conclusions.....	54
6.1 Conclusions.....	54
7. Futures ampliacions.....	55
7.1 Futures ampliacions.....	55
8. Glossari	56
9. Bibliografia.....	57

Índex de figures

Figura 0, planificació del projecte	15
Figura 1: Diagrama de casos d'ús	17
Figura 2: UC1 Gestionar Directori de Comerços	18
Figura 3: UC2 Afegir un Nou Comerç.....	18
Figura 4: UC3 Gestionar Esdeveniments	19
Figura 5: UC4 Eliminar Esdeveniment	19
Figura 6: UC5 Afegir Esdeveniment.....	20
Figura 7: UC6 Modificar Esdeveniment.....	20
Figura 8: UC7 Gestionar Notícies.....	21
Figura 9: UC8 Eliminar Notícia	21
Figura 10: UC9 Afegir Notícia	21
Figura 11: UC10 Modificar Notícia	22
Figura 12: UC11 Gestionar Perfil	22
Figura 13: UC12 Modificar Dades.....	23
Figura 14: UC13 Gestionar Catàleg Comerç	23
Figura 15: UC14 Afegir Producte	24
Figura 16: UC15 Eliminar Producte	24
Figura 17: UC16 Modificar Producte	25
Figura 18: UC17 Visualitzar Perfil Comerç.....	25
Figura 19: UC18 Visualitzar Perfil Agrupació.....	25
Figura 20: UC19 Visualitzar Notícies.....	26
Figura 21: UC20 Visualitzar Esdeveniments	26
Figura 22: UC21 Visualitzar Catàleg.....	26
Figura 23: UC22 Visualitzar Directori.....	26
Figura 24: Diagrama de classes	28
Figura 25: Taula de templates	30
Figura 26: Template pàgina principal agrupació	30
Figura 27: Template directori de comerços	31
Figura 28: Template catàleg de comerços.....	32
Figura 29: Template pàgina d'esdeveniments de l'agrupació.....	33
Figura 30: Template dashboard de l'agrupació.....	34
Figura 31: Template feed de notícies	35

Figura 32: Template pàgina principal del comerç	36
Figura 33: Pàgina d'una notícia	37
Figura 34: Pàgina d'un producte	38
Figura 35: Pàgina d'un Esdeveniment	39
Figura 36: Formulari nova notícia.....	39
Figura 37: exemple d'una crida a la API	43
Figura 38: perfil de l'agrupació.....	45
Figura 39: perfil de l'agrupació amb pantalla petita	46
Figura 40: Catàleg	47
Figura 41: Directori.....	47
Figura 42: formularis Django per defecte	48
Figura 42: formularis amb Crispy-Forms	48
Figura 43: barra de navegació del Dashboard de l'agrupació.....	50
Figura 44: barra de navegació del Dashboard del comerç.....	50

1. Introducció

1.1 Motivacions

Aquest projecte sorgeix com a una eina que pugui facilitar o millorar la presència *on-line* dels comerços locals.

El sector del comerç està canviant, la forma en que la societat actual consumeix és totalment diferent a la forma com es feia fa anys. Avui en dia els comerços que no tenen lloc web o no tenen presència *on-line* es troben en una posició de desavantatge. Per tant, és important donar facilitats als comerços per a que puguin millorar aquesta situació de desavantatge.

Actualment hi ha una gran varietat d'aplicacions web que generen llocs web, també n'hi ha d'especialitzats en comerços. No obstant, la majoria requereixen d'un mínim de coneixements o tenen una corba d'aprenentatge que en dificulta la iniciació. Totes aquestes aplicacions web acostumen a tenir una gran quantitat d'opcions de personalització que poden arribar a dificultar-ne l'ús a un usuari amb pocs coneixements.

Després d'una recerca, he pogut comprovar que la majoria de comerços locals o bé tenen un lloc web poc atractiu o no en tenen. Una petita part dels comerços locals té un lloc web que permet una navegació senzilla i és atractiu als usuaris.

Durant la recerca, també he constatat que hi ha agrupacions de comerços que tenen un lloc web on es donen a conèixer, i mostren informació sobre ells

1.2 Descripció del projecte

El projecte tracta de la realització d'una aplicació web que ajudi als comerços locals (que solen tenir poca presència a internet) a arribar als clients mitjançant les noves tecnologies.

L'aplicació està plantejada per a agrupacions de comerços, associacions de comerciants o ajuntaments que vulguin donar presència *on-line* als comerços locals.

A l'aplicació web es podran generar dos tipus de perfils. Un perfil (que serà únic) de l'agrupació de comerços i un perfil per a cada comerç de l'agrupació registrat. Es vol que els perfils generats puguin ésser usats com a lloc web. Així doncs, el perfil d'un comerç podrà funcionar com a lloc web d'aquest. Es per això que els perfils tindran l'estètica d'un lloc web.

El perfil de l'agrupació de comerços tindrà les següents seccions:

- Directori amb tots els comerços registrats.

- Notícies.
- Catàleg, on s'hi podran trobar i cercar productes de tots els comerços registrats.
- Esdeveniments, on s'hi podran trobar esdeveniments de dos tipus, esdeveniments que es realitzen a nivell de agrupació i uns altres esdeveniments que es realitzen a nivell de comerç.
- Informació de l'agrupació.

El perfil d'un comerç tindrà:

- Notícies.
- Catàleg, on trobar tots els productes del comerç.
- Esdeveniments, on s'hi podran trobar només esdeveniments a nivell de comerç.
- Informació del comerç.

1.3 Objectius

1.3.1 Objectius generals

L'objectiu d'aquest projecte és crear un programari que permeti que les agrupacions de comerços locals i els comerços que la formen puguin tenir un lloc web amb el que donar-se a conèixer als clients. Es pretén que sigui senzill d'utilitzar i no requereixi de molta dedicació.

1.3.2 Objectius específics

Els objectius en l'àmbit tècnic d'aquest projecte és aprofundir en el desenvolupament d'aplicacions web i ampliar el coneixement sobre el *framework Django de Python*

1.4 Mètode de planificació temporal

La realització d'aquest treball es durà a terme durant el semestre de primavera del curs 2018/2019.

Durant la realització del treball, es seguirà una metodologia basada en *sprints*.

Els *sprints* són períodes curts de temps on es plantegen i posteriorment s'estimen uns objectius. Un cop estimats, es seleccionen els objectius que poden encabir-se dins del temps que es disposa. Al final del *sprint*, s'han d'haver complert les tasques estimades.

Per tal de permetre un desenvolupament sobre uns fonaments sòlids i estables, tots els *sprints* que contenen una part de desenvolupament d'alguna part del software del projecte, contenen implícitament unes hores per a realitzar proves sobre totes aquelles funcionalitats (modificat o afegides) que s'hagin acabat durant el *sprint*.

Per tal de no cometre errors de planificació que causin que no es puguin assolir alguns objectius o requisits marcats, s'ha establert un període de 15 hores lliures al penúltim *sprint*. En el cas de que s'hagi estimat malament alguna tasca i hi hagin tasques del desenvolupament que vagin fora de temps, aquestes hores serviran per acabar-les. En canvi, si s'ha estimat bé el temps, aquestes hores serviran per a realitzar alguna tasca que es cregui convenient.

1.5 Sprints

Es disposen de 18 setmanes de treball per a la realització del projecte. S'ha pres la decisió de que els *sprints* siguin períodes de dues setmanes. Donat que es pretén una dedicació d'unes 5 hores diàries, una duració de 25 hores per cada *sprint* podia ser poca. Així doncs, hi haurà 9 *sprints*.

1.5.1 Sprint 1

Durant aquest *sprint* es pretén escollir un tema pel projecte i definir un entorn per al desenvolupament del projecte.

Així doncs, les tasques que es duran a terme són:

- Pluja d'idees per tal de poder escollir el tema.
- Escollir un entorn de desenvolupament per el projecte.
- Començar amb la documentació i l'aprenentatge del entorn escollit.

1.5.2 Sprint 2

Per al següent *sprint*, és vol establir un pla pel projecte i començar redactant la memòria amb aquest.

- Descripció del projecte
- Requeriments del projecte
- Objectius del projecte
- Seguir amb la documentació i aprendre en profunditat el entorn de desenvolupament escollit. (*Framework Django*)

1.5.3 Sprint 3

Com ja s'ha realitzat una recerca suficient sobre l'entorn de desenvolupament, fer una possible planificació i estimació del desenvolupament del projecte. Les tasques que es realitzaran aquest *sprint* seran:

- Diagrama de casos d'ús.

- Diagrama de classes.
- Casos d'ús textuais.
- Establir i estimar les tasques que s'han de dur a terme durant el desenvolupament del projecte.

1.5.4 *Sprint 4*

Com ja s'ha realitzat una primera estimació, començar amb les primeres tasques que s'han escollit. En aquest cas, s'ha decidit començar el desenvolupament del projecte amb el prototipatge i la creació de les plantilles per les pàgines. Les tasques que es pretén dur a terme seran:

- Realització de varis prototips dels quals finalment, se'n escollirà un.
- Un cop decidit en quin prototip es basarà el lloc web, es començarà a realitzar les plantilles de les pàgines amb seguint als prototips.
- Redactar a la memòria el prototip final escollit.
- Instal·lació i posada en marxa del entorn necessari per començar el desenvolupament.

1.5.5 *Sprint 5*

Un cop es tenen les plantilles per a mostrar les dades, l'objectiu d'aquest *sprint* és començar amb el desenvolupament del projecte.

- Definició i desenvolupament del model de dades de l'aplicació.
- Implementació de vistes i formularis de *Django* per a poder introduir i modificar les dades.

1.5.6 *Sprint 6*

Amb el model de dades creat i els formularis que permeten la introducció i la modificació de les dades, es pretén desenvolupar la lògica de l'aplicació.

- Introducció de les pàgines de *dashboards* amb tot el seu contingut.
- Creació de diferents perfils d'usuaris.

1.5.7 *Sprint 7*

Un cop creats els perfils d'usuaris, establir quines parts són accessibles per cada un dels perfils que hi ha al sistema i implementar un sistema de permisos. Al finalitzar aquest *sprint*, es consideraran acabades les tasques de desenvolupament del projecte.

- Establir permisos per a cada tipus d'usuari de l'aplicació.

Aquest *sprint* només té els permisos ja que un cop s'hagin establert, s'haurà de provar tota l'aplicació varies vegades. En concret s'haurà de provar l'aplicació 3 cops, un per cada perfil d'usuari

1.5.8 *Sprint 8*

Durant aquest *sprint*, es pretén continuar amb el redactat de la memòria. També s'hi inclourà una franja de 15 hores per si hi ha alguna tasca anterior sense finalitzar.

- Finalitzar tasques que no s'hagin finalitzat als *sprints* anteriors. En el cas de que no hi hagi cap tasca a dur a terme, incloure alguna tasca que es consideri.
- Començar amb la redacció de la implementació a la memòria.

1.5.9 *Sprint 9*

Durant aquest *sprint*, es vol acabar amb la redacció de la memòria per a poder entregar-ne un esborrany amb suficient antelació.

1.6 Dates de lliurament

Data de lliurament	Lliurament	Descripció del lliurament
14 de Març	Pla del projecte	Pla de treball en el qual es detalla la descripció, els requeriments previs i la planificació temporal del projecte
4 d'Abril	Anàlisi i disseny	L'anàlisi i el disseny prèvies a la implementació del projecte.
9 de Maig	Implementació	Una versió final i funcional del projecte.
20 de Juny	Memòria	Document de la memòria.
1 al 10 de Juliol	Defensa	Presentació del projecte.

Figura 0, planificació del projecte

2 Anàlisi

2.1 Anàlisi i definició de les funcionalitats

Un cop exposat el funcionament de la pàgina web al apartat d'introducció, caldrà definir els usuaris del sistema i quines accions pot realitzar cada un d'ells. Per això s'ha utilitzat un diagrama de casos d'ús i la descripció de cada cas.

Per a la pàgina web, s'han considerat tres tipus d'usuaris que cal diferenciar:

1. Administrador de l'agrupació.
2. Administrador del comerç.
3. Usuari.

2.2 Diagrama de casos d'ús

Tenint definits els tipus d'usuaris que hi podrà haver al sistema, s'ha d'establir quines accions podrà dur a terme cada un d'ells.

La següent figura mostra el diagrama de casos d'ús, on s'hi pot veure una relació entre els tipus d'usuaris i les accions que poden realitzar.

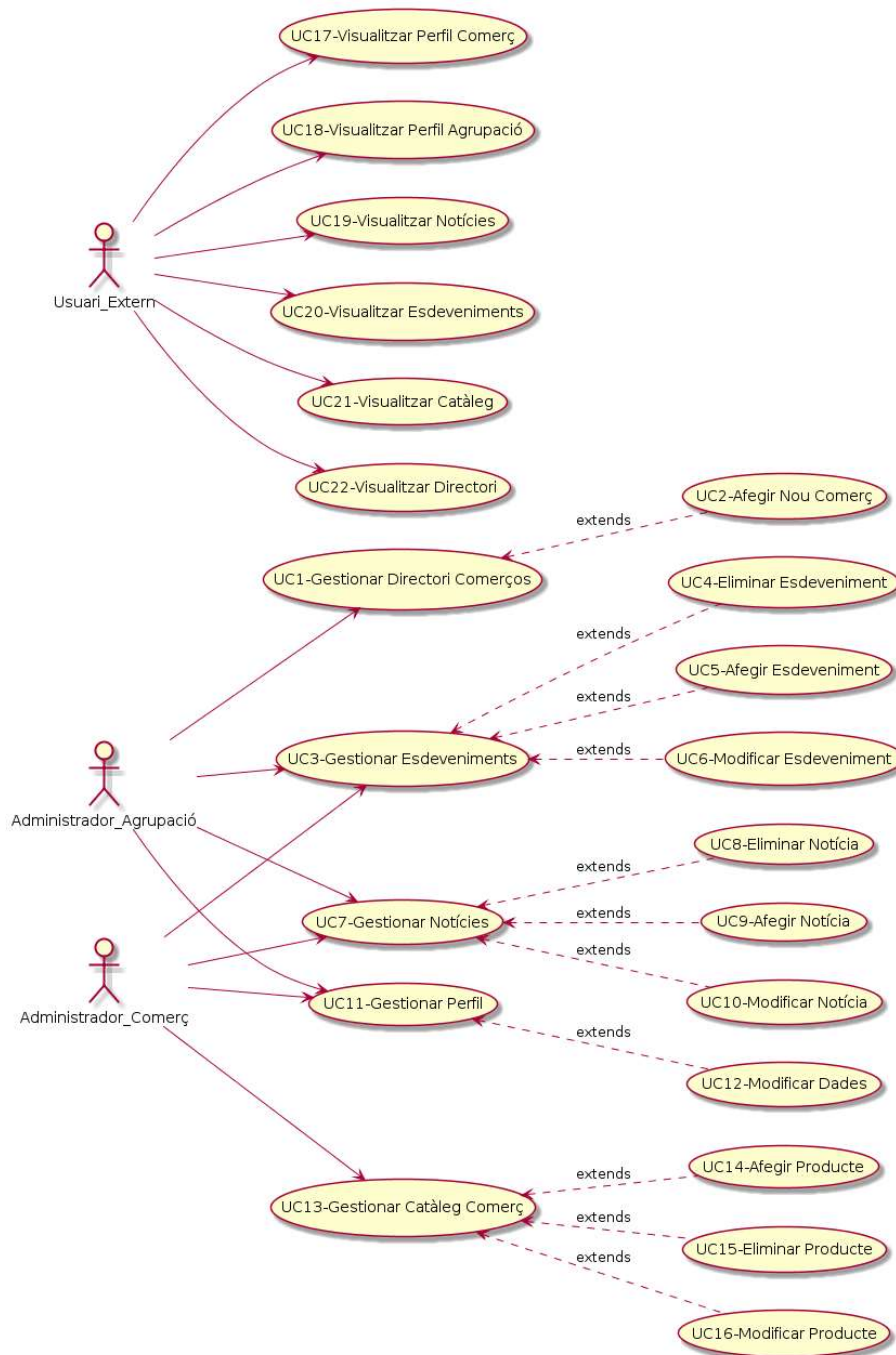


Figura 1: Diagrama de casos d'ús

2.3 Descripció dels casos d'us textuals

Descripció	UC1: Gestionar Directori Comerços
Actors	Administrador de l'agrupació
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació.
Flux bàsic	1. Administrador de l'agrupació: Accedeix al <i>dashboard</i> .

	<ol style="list-style-type: none"> 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador de l'agrupació: Selecciona la opció "Directori" del menú de navegació del <i>dashboard</i>. 4. Sistema: Mostra un llistat amb tots els comerços registrats.
Flux alternatiu	<p>a)</p> <ol style="list-style-type: none"> 4. Sistema: No hi ha comerços registrats, es mostra una llista buida.
Post condicions	-

Figura 2: UC1 Gestionar Directori de Comerços

Descripció	UC2: Afegir Nou Comerç
Actors	Administrador de l'agrupació
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador de l'agrupació: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador de l'agrupació: Selecciona la opció "Directori" del menú de navegació del <i>dashboard</i>. 4. Sistema: Mostra un llistat amb tots els comerços registrats. 5. Administrador de l'agrupació: Selecciona el botó per afegir un comerç nou. 6. Sistema: Mostra un formulari que l'usuari ha d'omplir amb les dades necessàries per afegir un nou comerç. 7. Administrador de l'agrupació: Omple el formulari i selecciona la opció de desar. 8. Sistema: Es validen els camps del formulari i es crea un nou comerç.
Flux alternatiu	<p>a)</p> <ol style="list-style-type: none"> 8. Sistema: Hi ha hagut un problema al validar el formulari, es marquen les dades incorrectes i es torna al punt 6 del flux bàsic.
Post condicions	S'ha creat un nou comerç i un perfil amb permisos per a gestionar aquest nou comerç.

Figura 3: UC2 Afegir un Nou Comerç

Descripció	UC3: Gestionar Esdeveniments
Actors	Administrador Agrupació i Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació o les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona la opció "Esdeveniments" del menú de navegació del <i>dashboard</i>.

	4. Sistema: Mostra un llistat amb tots els esdeveniments que pot gestionar l'usuari.
Flux alternatiu	-
Post condicions	-

Figura 4: UC3 Gestionar Esdeveniments

Descripció	UC4: Eliminar Esdeveniment
Actors	Administrador Agrupació i Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació o les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona la opció "Esdeveniments" del menú de navegació del <i>dashboard</i>. 4. Sistema: Mostra un llistat amb tots els esdeveniments que pot gestionar l'usuari. 5. Administrador: Selecciona el botó eliminar al costat del esdeveniment que vol eliminar. 6. Sistema: Suprimeix totes les dades del esdeveniment.
Flux alternatiu	a) <ol style="list-style-type: none"> 6. Sistema: Hi ha hagut un error al suprimir les dades del esdeveniment. Es notifica l'error i es torna al punt 4.
Postcondicions	S'han eliminat les dades del esdeveniment seleccionat.

Figura 5: UC4 Eliminar Esdeveniment

Descripció	UC5: Afegir Esdeveniment
Actors	Administrador Agrupació i Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació o les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona la opció "Esdeveniments" del menú de navegació del <i>dashboard</i>. 4. Sistema: Mostra un llistat amb tots els esdeveniments que pot gestionar l'usuari. 5. Administrador: Selecciona l'opció afegir un esdeveniment. 6. Sistema: Mostra una formulari que l'usuari ha d'omplir. 7. Administrador: Omple el formulari i selecciona l'opció de desar. 8. Sistema: Es validen els camps del formulari i es crea un nou esdeveniment.
Flux alternatiu	a)

	8. Sistema: Hi ha hagut un problema al validar el formulari, es marquen les dades incorrectes i es torna al punt 6 del flux bàsic.
Postcondicions	S'ha creat un nou esdeveniment al sistema

Figura 6: UC5 Afegir Esdeveniment

Descripció	UC6: Modificar Esdeveniment
Actors	Administrador Agrupació i Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació o les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona la opció "Esdeveniments" del menú de navegació del <i>dashboard</i>. 4. Sistema: Mostra un llistat amb tots els esdeveniments que pot gestionar l'usuari. 5. Administrador: Selecciona l'esdeveniment que es vol modificar del llistat. 6. Sistema: Mostra una formulari (amb dades ja omplertes) que l'usuari ha d'omplir o modificar. 7. Administrador: Omple el formulari i selecciona l'opció de desar. 8. Sistema: Es validen els camps del formulari i es crea un nou esdeveniment.
Flux alternatiu	a) <ol style="list-style-type: none"> 8. Sistema: Hi ha hagut un problema al validar el formulari, es marquen les dades incorrectes i es torna al punt 6 del flux bàsic.
Postcondicions	S'han modificat les dades d'un esdeveniment ja existent.

Figura 7: UC6 Modificar Esdeveniment

Descripció	UC7: Gestionar Notícies
Actors	Administrador Agrupació i Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació o les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona l'opció "Notícies" del menú de navegació del <i>dashboard</i>. 4. Sistema: Mostra un llistat amb totes les notícies que pot gestionar l'usuari.
Flux alternatiu	-
Postcondicions	

Figura 8: UC7 Gestionar Notícies

Descripció	UC8: Eliminar Notícia
Actors	Administrador Agrupació i Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació o les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona l'opció "Notícies" del menú de navegació del <i>dashboard</i>. 4. Sistema: Mostra un llistat amb totes les notícies que pot gestionar l'usuari. 5. Administrador: Selecciona el botó d'eliminar que hi ha al costat de la notícia que vol eliminar. 6. Sistema: Es suprimeixen totes les dades de la notícia.
Flux alternatiu	<p>a)</p> <ol style="list-style-type: none"> 6. Sistema: Hi ha hagut un error al suprimir les dades de la notícia. Es notifica l'error i es torna al punt 4.
Postcondicions	S'han eliminat les dades del esdeveniment seleccionat.

Figura 9: UC8 Eliminar Notícia

Descripció	UC9: Afegir Notícia
Actors	Administrador Agrupació i Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació o les dades d'un comerç..
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona l'opció "Notícies" del menú de navegació del <i>dashboard</i>. 4. Sistema: Mostra un llistat amb totes les notícies que pot gestionar l'usuari. 5. Administrador: Selecciona el botó d'afegir una notícia. 6. Sistema: Mostra un formulari que l'usuari ha d'omplir. 7. Administrador: Omple el formulari i selecciona l'opció de desar. 8. Sistema: Es validen els camps del formulari i es crea una nova notícia.
Flux alternatiu	<p>a)</p> <ol style="list-style-type: none"> 8. Sistema: Hi ha hagut un problema al validar el formulari, es marquen les dades incorrectes i es torna al punt 6 del flux bàsic.
Postcondicions	S'ha creat una nova notícia al sistema

Figura 10: UC9 Afegir Notícia

Descripció	UC10: Modificar Notícia
------------	-------------------------

Actors	Administrador Agrupació i Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació o les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona l'opció Notícies del menú de navegació del <i>dashboard</i>. 4. Sistema: Mostra un llistat amb totes les notícies que pot gestionar l'usuari. 5. Administrador: Selecciona la notícia de la llista que vol modificar. 6. Sistema: Mostra un formulari (amb camps ja omplerts) que l'usuari ha d'omplir o modificar. 7. Administrador: Omple o modifica les dades del formulari i selecciona l'opció de desar. 8. Sistema: Es validen els camps del formulari i es modifica la notícia.
Flux alternatiu	<p>a)</p> <ol style="list-style-type: none"> 8. Sistema: Hi ha hagut un problema al validar el formulari, es marquen les dades incorrectes i es torna al punt 6 del flux bàsic.
Postcondicions	S'han modificat les dades d'una notícia ja existent.

Figura 11: UC10 Modificar Notícia

Descripció	UC11: Gestionar Perfil
Actors	Administrador Agrupació i Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació o les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona l'opció "Dades de l'agrupació" o "Dades del comerç" (depenent dels permisos del usuari apareix una opció o l'altre) del menú de navegació del <i>dashboard</i>. 4. Sistema: Mostra un formulari amb totes les dades del perfil.
Flux alternatiu	
Postcondicions	

Figura 12: UC11 Gestionar Perfil

Descripció	UC12: Modificar Dades
Actors	Administrador Agrupació i Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades de l'agrupació o les dades d'un comerç.

Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona l'opció "Dades de l'agrupació" o "Dades del comerç" (depenent dels permisos del usuari apareix una opció o l'altre) del menú de navegació del <i>dashboard</i>. 4. Sistema: Mostra un formulari amb totes les dades del perfil. 5. Administrador: Modifica les dades del formulari i selecciona l'opció de desar. 6. Sistema: Es validen els camps del formulari i es modifiquen les dades del perfil.
Flux alternatiu	<p>a)</p> <ol style="list-style-type: none"> 6. Sistema: S'ha produït un error, es notifica al usuari i es torna al punt 4 del flux bàsic.
Postcondicions	S'han modificat les dades del perfil.

Figura 13: UC12 Modificar Dades

Descripció	UC13: Gestionar Catàleg Comerç
Actors	Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona l'opció "Productes" del menú de navegació. 4. Sistema: Mostra una llista amb tots els productes que té el comerç que gestiona l'usuari.
Flux alternatiu	<p>a)</p> <ol style="list-style-type: none"> 4. Sistema: No hi ha productes al catàleg, es mostra una llista buida.
Postcondicions	

Figura 14: UC13 Gestionar Catàleg Comerç

Descripció	UC14: Afegir Producte
Actors	Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona l'opció "Productes" del menú de navegació.

	<ol style="list-style-type: none"> 4. Sistema: Mostra un llistat amb tots els productes que té el comerç que gestiona l'usuari. 5. Administrador: Selecciona l'opció d'afegir un nou producte. 6. Sistema: Mostra un formulari que l'usuari ha d'omplir. 7. Administrador: Omple el formulari i selecciona l'opció de desar. 8. Sistema: Es validen els camps del formulari i es crea un nou producte al sistema.
Flux alternatiu	<ol style="list-style-type: none"> a) <ol style="list-style-type: none"> 8. Sistema: Hi ha hagut un problema al validar el formulari, es marquen les dades incorrectes i es torna al punt 5 del flux bàsic.
Postcondicions	S'ha creat un nou producte del comerç al sistema que pertany al comerç que gestiona l'usuari que té la sessió iniciada.

Figura 15: UC14 Afegir Producte

Descripció	UC15: Eliminar Producte
Actors	Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari. 3. Administrador: Selecciona l'opció "Productes" del menú de navegació. 4. Sistema: Mostra un llistat amb tots els productes que té el comerç que gestiona l'usuari. 5. Administrador: Selecciona el botó d'eliminar que hi ha al costat del producte que es vol eliminar. 6. Sistema: Suprimeix les dades del producte.
Flux alternatiu	<ol style="list-style-type: none"> a) <ol style="list-style-type: none"> 6. Sistema: Hi ha hagut un problema al eliminar les dades del producte, es notifica l'error al usuari i es torna al punt 4 del flux bàsic.
Postcondicions	S'han suprimit totes les dades d'un producte.

Figura 16: UC15 Eliminar Producte

Descripció	UC16: Modificar Producte
Actors	Administrador Comerç
Precondicions	L'usuari ha d'haver iniciat sessió amb un usuari que tingui permisos per gestionar les dades d'un comerç.
Flux bàsic	<ol style="list-style-type: none"> 1. Administrador: Accedeix al <i>dashboard</i>. 2. Sistema: Mostra el <i>dashboard</i> amb totes les opcions que té l'usuari.

	<ol style="list-style-type: none"> 3. Administrador: Selecciona l'opció "Productes" del menú de navegació. 4. Sistema: Mostra un llistat amb tots els productes que té el comerç que gestiona l'usuari. 5. Administrador: Selecciona el producte que es vol modificar de la llista. 6. Sistema: Mostra un formulari (amb camps ja omplerts) que l'usuari ha d'omplir. 7. Administrador: Omple el formulari i selecciona l'opció de desar. 8. Sistema: Es validen els camps del formulari i es modifiquen les dades del producte.
Flux alternatiu	<ol style="list-style-type: none"> a) <ol style="list-style-type: none"> 8. Sistema: Hi ha hagut un problema al validar el formulari, es marquen les dades incorrectes i es torna al punt 6 del flux bàsic.
Postcondicions	S'han modificat les dades del producte.

Figura 17: UC16 Modificar Producte

Descripció	UC17: Visualitzar Perfil Comerç
Actors	Usuari Extern
Precondicions	
Flux bàsic	<ol style="list-style-type: none"> 1. Usuari: Accedeix al perfil d'un comerç. 2. Sistema: Mostra el perfil del comerç.
Flux alternatiu	
Postcondicions	

Figura 18: UC17 Visualitzar Perfil Comerç

Descripció	UC18: Visualitzar Perfil Agrupació
Actors	Usuari Extern
Precondicions	
Flux bàsic	<ol style="list-style-type: none"> 1. Usuari: Accedeix al perfil de l'agrupació. 2. Sistema: Mostra el perfil de l'agrupació.
Flux alternatiu	
Postcondicions	

Figura 19: UC18 Visualitzar Perfil Agrupació

Descripció	UC19: Visualitzar Notícies
Actors	Usuari Extern
Precondicions	
Flux bàsic	<ol style="list-style-type: none"> 1. Usuari: Accedeix al perfil del que es vol visualitzar les notícies. 2. Sistema: Mostra el perfil. 3. Usuari Selecciona l'opció de navegació "Notícies". 4. Sistema: Es mostren les notícies del perfil.
Flux alternatiu	

Postcondicions	
----------------	--

Figura 20: UC19 Visualitzar Notícies

Descripció	UC20: Visualitzar Esdeveniments
Actors	Usuari Extern
Precondicions	
Flux bàsic	<ol style="list-style-type: none"> 1. Usuari: Accedeix al perfil del que es vol visualitzar els esdeveniments. 2. Sistema: Mostra el perfil. 3. Usuari: Selecciona l'opció de navegació "Esdeveniments" 4. Sistema: Es mostren els esdeveniments del perfil.
Flux alternatiu	
Postcondicions	

Figura 21: UC20 Visualitzar Esdeveniments

Descripció	UC21: Visualitzar Catàleg
Actors	Usuari Extern
Precondicions	
Flux bàsic	<ol style="list-style-type: none"> 1. Usuari: Accedeix al perfil del que es vol visualitzar el catàleg. 2. Sistema: Mostra el perfil. 3. Usuari: Selecciona l'opció de navegació "Catàleg". 4. Sistema: Es mostra el catàleg del perfil.
Flux alternatiu	
Postcondicions	

Figura 22: UC21 Visualitzar Catàleg

Descripció	UC22: Visualitzar Directori
Actors	Usuari Extern
Precondicions	
Flux bàsic	<ol style="list-style-type: none"> 1. Usuari: Accedeix al perfil de l'agrupació. 2. Sistema: Mostra el perfil de l'agrupació. 3. Usuari: Selecciona l'opció de navegació "Directori". 4. Sistema: Es mostren les notícies del perfil.
Flux alternatiu	
Postcondicions	

Figura 23: UC22 Visualitzar Directori

3 Disseny

3.1 Arquitectura

L'arquitectura que es seguirà en aquest projecte és la MTV (*model-template-view*), és el model en el que es basa el *framework Django*.

Model

El *model* és la capa d'accés a les dades, s'hi pot trobar com estan estructurades i es guardaran les dades a l'aplicació. Tot i que *Django* utilitza per defecte bases de dades SQL, permet definir classes al fitxer ***models.py*** i el propi *framework* s'encarregarà de crear o modificar les taules de la base de dades.

Template

El *template* es la capa de presentació, definirà com es mostra la informació al usuari. Es definirà als fitxers de *templates* de l'aplicació que s'explicaran més endavant.

View

La *view* és on hi trobarem la lògica de l'aplicació, que estarà definit al arxiu ***views.py***.

Fitxer *urls.py*

També cal afegir que fora d'aquests tres components del patró MVT, el *framework Django* fa servir un fitxer anomenat ***urls.py***. Dins del fitxer hi ha declarades les possibles direccions URL que l'aplicació admet i el mètode de la vista que s'executarà al rebre una petició.

Per exemple, quan es rebí una petició, si la direcció URL està dins del fitxer ***urls.py***, s'executarà el mètode de la *view* que hi hagi indicat. El mètode de la *view* farà les operacions necessàries i podrà utilitzar el *model* per a comunicar-se amb la base de dades i obtenir les dades necessàries. Finalment la *view* passarà les dades al *template* per a que les mostri al usuari.

3.2 Diagrama de classes

A continuació es mostra el Diagrama de classes que està definit al fitxer ***models.py***.

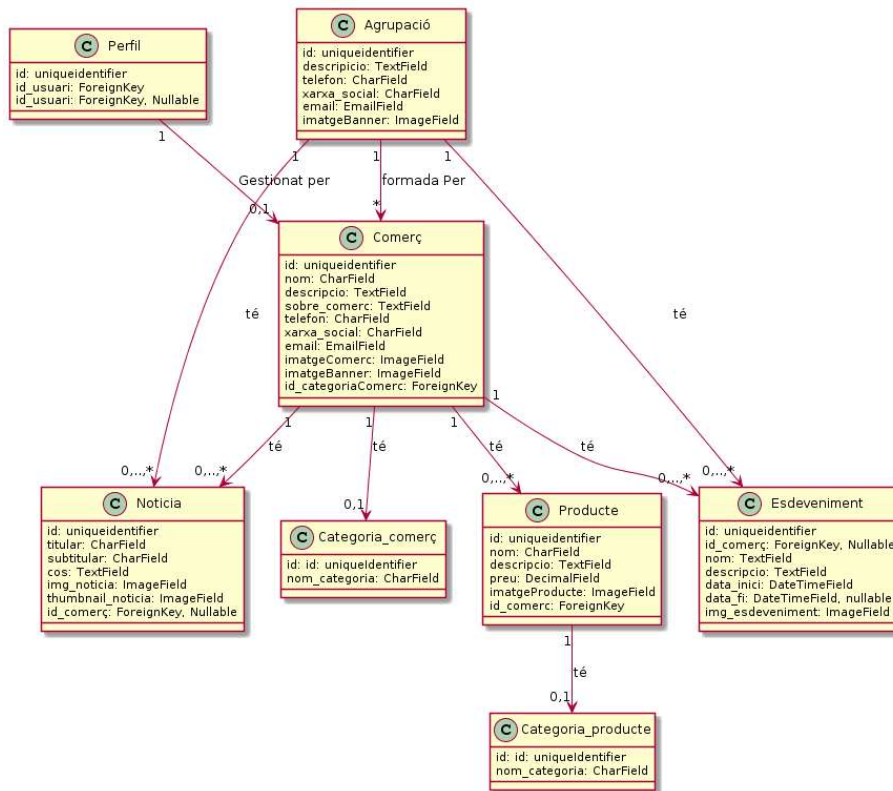


Figura 24: Diagrama de classes

3.3 Templates

En aquest punt del disseny, es tractarà el disseny de l'aplicació, tant en l'aspecte gràfic com en l'aspecte d'interacció amb l'usuari.

S'ha realitzat un prototip mitjançant l'eina online *Balsamiq*. En aquest prototip no s'han creat tots els *templates* que tindrà l'aplicació donat que molts templates comparteixen estructura o bé només varien en contingut. Per exemple, aquelles pàgines que serviran per introduir dades via formulari, només variaran en els camps que hi ha al formulari, tots mantindran la mateixa estètica.

És clau destacar que el prototip que es comenta a continuació és un dels primers prototips que s'ha realitzat durant la fase de disseny. Per tant, el prototip pot no ser totalment fidel amb la versió final.

A la següent figura, podem veure la totalitat dels *templates* que formaran l'aplicació i una breu descripció.

Perfil de l'agrupació	Aquesta serà la pàgina inicial de l'aplicació web. Servirà com una presentació de l'agrupació i mostrarà al usuari les dades més rellevants sobre aquesta.
Director de comerços	Aquesta és la pàgina on s'hi pot trobar una llista de tots els comerços que hi ha a l'aplicació, un resum sobre cada un dels comerços i informació destacable sobre el comerç.

	També hi haurà una barra lateral amb una llista de categories per permetre a l'usuari filtrar.
Catàleg de l'agrupació	Aquesta pàgina mostra tots els productes que hi ha a l'agrupació. És on s'hi poden trobar tots els productes de tots els comerços que hi ha a l'aplicació. També hi haurà una barra lateral amb una llista de categories per permetre a l'usuari filtrar.
Pàgina d'esdeveniments de l'agrupació	Aquesta pàgina mostrarà un llistat amb els esdeveniments que organitzats tant per l'agrupació com per un dels comerços que la formen. Els esdeveniments estaran classificats en tres llistes (en curs, propers i actuals).
Feed de notícies de l'agrupació	En aquesta pàgina s'hi trobaran les notícies de l'agrupació.
Dashboard de l'agrupació	Un panell de control per a l'administrador de la pàgina de l'agrupació. Des d'aquesta pàgina, el administrador podrà accedir a gestionar notícies, esdeveniments, comerços i les dades de l'agrupació.
Perfil del comerç	Aquesta pàgina serà la pàgina principal d'un comerç a l'aplicació. Servirà com una presentació del comerç i mostrarà al usuari les dades més rellevants sobre aquest.
Catàleg del comerç	Aquesta pàgina mostra tots els productes d'un comerç. També hi haurà una barra lateral amb una llista de categories per permetre a l'usuari filtrar.
Pàgina d'esdeveniments del comerç	Aquesta pàgina mostrarà un llistat amb els esdeveniments que organitzats per el comerç. Els esdeveniments estaran classificats en tres llistes (en curs, propers i actuals).
Feed de notícies del comerç	En aquesta pàgina s'hi trobaran les notícies del comerç.
Dashboard del comerç	Un panell de control per a l'administrador de la pàgina del comerç. Des d'aquesta pàgina, el administrador podrà accedir a gestionar notícies, esdeveniments, productes i les dades del comerç.
Pàgina de la notícia	Pàgina per veure una notícia.
Pàgina d'un producte	Pàgina per veure un producte.
Formulari de creació d'un comerç	Un formulari amb totes les dades necessàries per a la creació d'un comerç. Aquesta pàgina serà accessible per l'administrador de l'agrupació. Un cop s'ha introduït les dades d'un comerç al formulari i es trameta, es crea un nou comerç i un perfil que el podrà gestionar.
Formulari dades d'un comerç	Un formulari on l'administrador del comerç podrà editar les dades del comerç.
Formulari producte	Un formulari on s'hi podran editar i crear productes.
Formulari esdeveniment	Un formulari on s'hi podrà editar i crear un esdeveniment.

Figura 25: Taula de templates

3.3.1 Perfil de l'agrupació

A la següent figura s'hi pot veure el prototip del perfil de l'agrupació.

Des d'aquesta pàgina l'usuari podrà veure les dades més rellevants de l'agrupació i les últimes actualitzacions. A la part superior, podrà trobar-hi la barra de navegació que li permetrà navegar per les pàgines de l'agrupació.

Les pàgines accessibles des de la barra de navegació de l'agrupació seran:

- [Directori dels comerços.](#)
- [Catàleg de l'agrupació.](#)
- [Calendari d'esdeveniments de l'agrupació.](#)
- [Feed de notícies de l'agrupació.](#)

També cal afegir que des de la barra de navegació l'usuari podrà:

- Iniciar sessió, si no hi ha cap usuari amb la sessió iniciada.
- Accedir al panell de control, si hi ha un usuari amb la sessió iniciada. Depenent dels permisos dels que disposi l'usuari, veurà el [Dashboard de l'agrupació](#) (si té permisos per gestionar el perfil de l'agrupació) o el [Dashboard del comerç](#) (si té permisos per gestionar el perfil del comerç).
- Tancar la sessió, si hi ha un usuari amb la sessió iniciada.

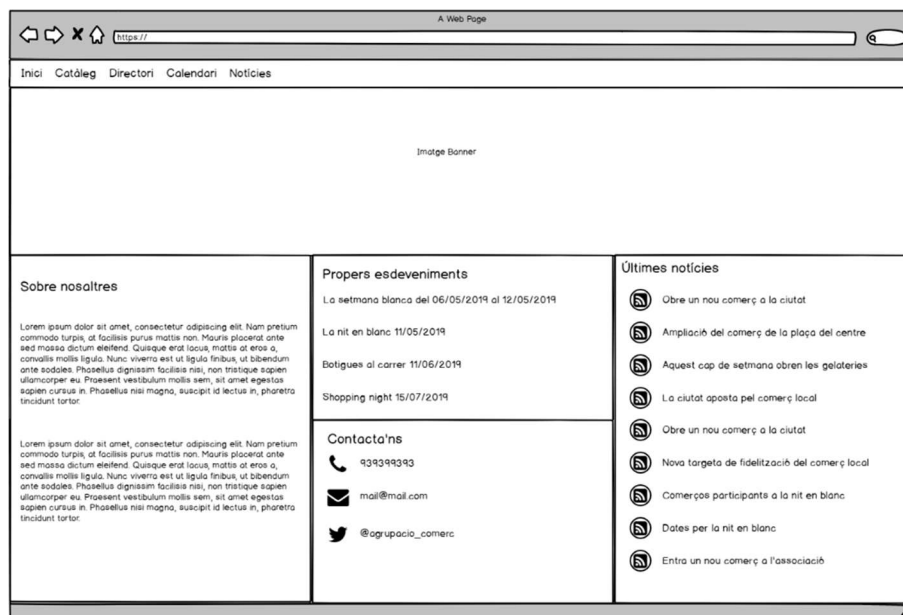


Figura 26: Template pàgina principal agrupació

3.3.2 Directori de comerços

A la següent figura, s'hi pot veure el prototip del directori de comerços.

Aquesta pàgina estarà formada per:

- La barra de navegació de l'agrupació a la part superior.
- Una barra lateral des de on l'usuari podrà filtrar per categoria els comerços.
- Un llistat amb tots els comerços que hi ha a l'agrupació. A part del nom de cada comerç de la llista s'hi podrà veure d'altra informació rellevant com un resum o el numero telèfon.

Si l'usuari fa clic en algun comerç de la llista, serà dirigit a la pàgina del comerç ([apartat 3.3.7](#)).

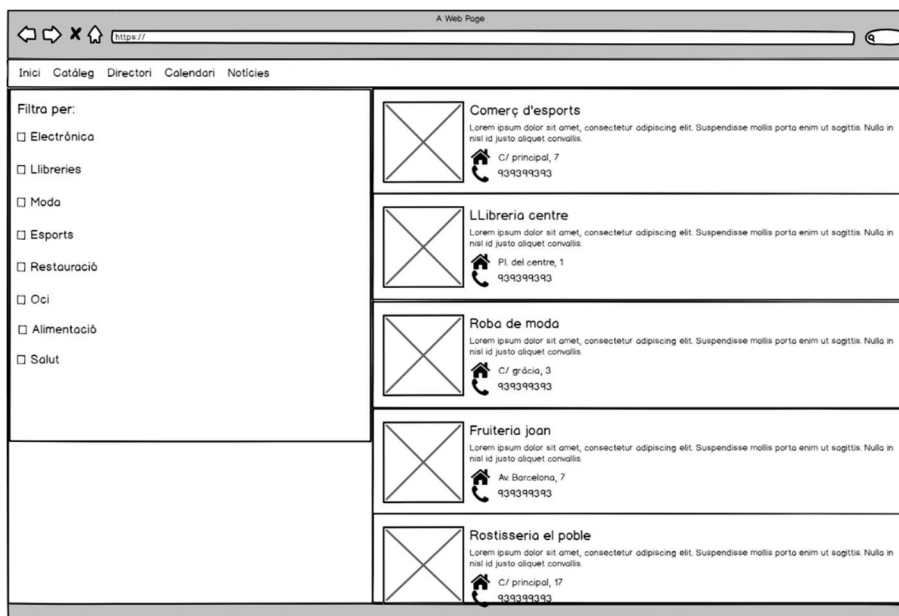


Figura 27: Template directori de comerços

3.3.3 Catàleg de l'agrupació

A continuació està el prototip del catàleg de l'agrupació, des d'on l'usuari podrà veure els productes de tots els comerços de l'agrupació.

Les parts que formen el catàleg de l'agrupació són:

- A la part superior, la barra de navegació de l'agrupació.
- Una barra lateral que permetrà filtrar els productes per categories.
- Un llistat de productes.

Si l'usuari fa clic en algun producte de la llista, serà dirigit a la pàgina del producte ([apartat 3.3.13](#)).

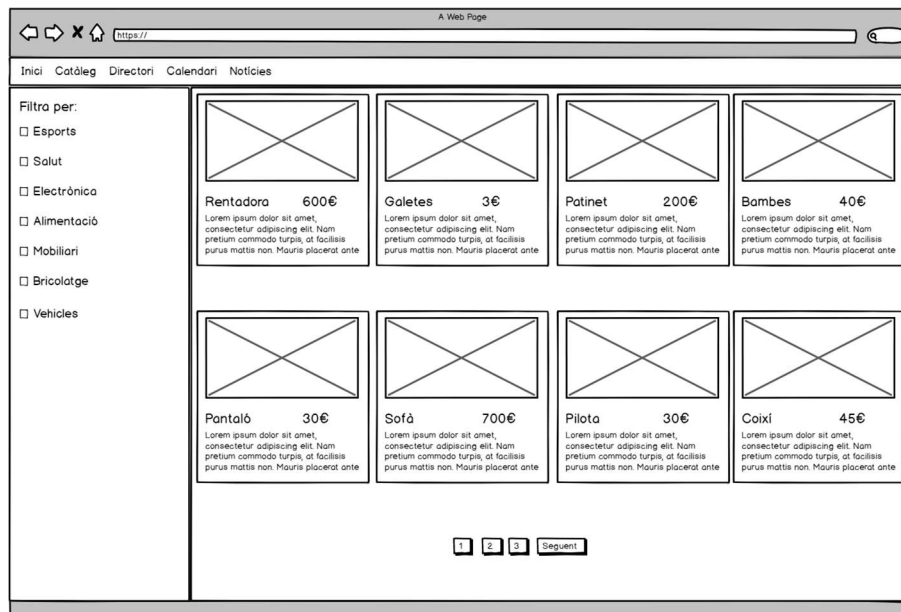


Figura 28: Template catàleg de comerços

3.3.4 Pàgina d'esdeveniments de l'agrupació

La següent figura mostra la pàgina d'esdeveniments de l'agrupació. Aquesta pàgina servirà perquè l'usuari pugui veure de manera senzilla els esdeveniments actuals, els que es duran a terme i els que ja s'han dut a terme.

Aquesta pàgina estarà formada per:

- La barra de navegació de l'agrupació.
- Els esdeveniments que hi ha al sistema, classificats en tres columnes.
 - La primera columna la formaran els esdeveniments en curs.
 - La segona columna la formaran els esdeveniments que es duran a terme pròximament.
 - La tercera columna la formaran els esdeveniments que ja s'han dut a terme.

Si l'usuari selecciona un esdeveniment, serà dirigit a la pàgina del esdeveniment ([apartat 3.3.14](#)).

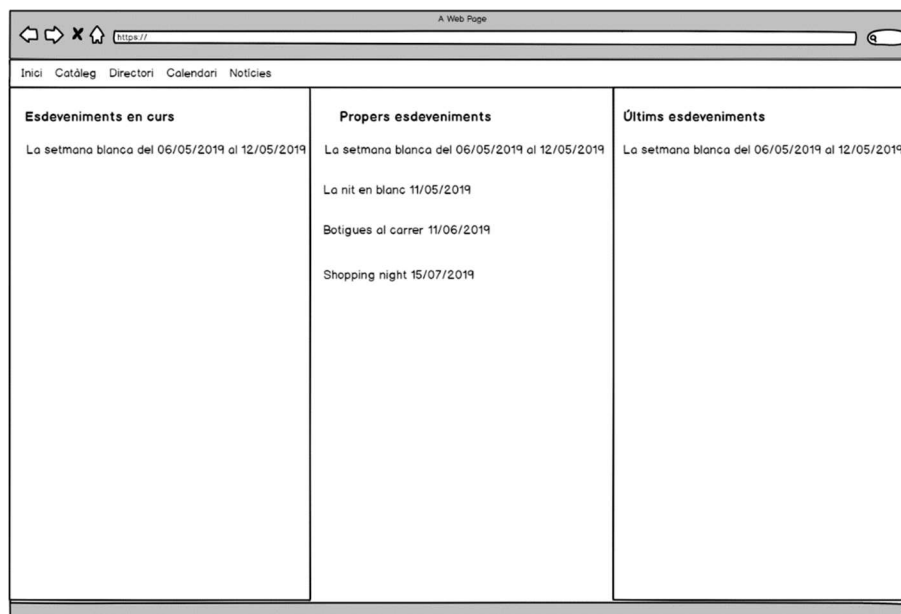


Figura 29: Template pàgina d'esdeveniments de l'agrupació

3.3.5 Dashboard de l'agrupació

La figura que hi ha a continuació representa el prototip del *dashboard* de l'agrupació.

El *dashboard* de l'agrupació és el panell de control amb el que un usuari amb permisos per a gestionar el perfil de l'agrupació podrà editar-ne les dades, gestionar els comerços que hi ha al sistema i les notícies i esdeveniments de l'agrupació.

El *dashboard* estarà format per:

- Una barra de navegació per les opcions del *dashboard*.
- El contingut de la pàgina, que variarà segons la opció de la barra de navegació seleccionada.

Les opcions que l'administrador de l'agrupació podrà seleccionar des de la barra de navegació del *dashboard* seran:

- **Dades de l'agrupació.** Si es selecciona aquesta opció, el contingut de la pàgina serà un formulari amb les dades de l'agrupació per a que l'administrador les pugui modificar.
- **Directori.** Si es selecciona aquesta opció, el contingut de la pàgina serà un llistat amb tots els comerços que hi ha al sistema. També mostrarà un botó que permetrà afegir un nou comerç.
- **Notícies.** Si es selecciona aquesta opció, el contingut de la pàgina serà un llistat amb totes les notícies de l'agrupació. Al costat de cada notícia hi haurà un botó per eliminar-la.

- **Esdeveniments.** Si es selecciona aquesta opció, el contingut de la pàgina serà un llistat amb tots els esdeveniments de l'agrupació. Cada esdeveniment tindrà un botó al costat que permetrà eliminar-lo.

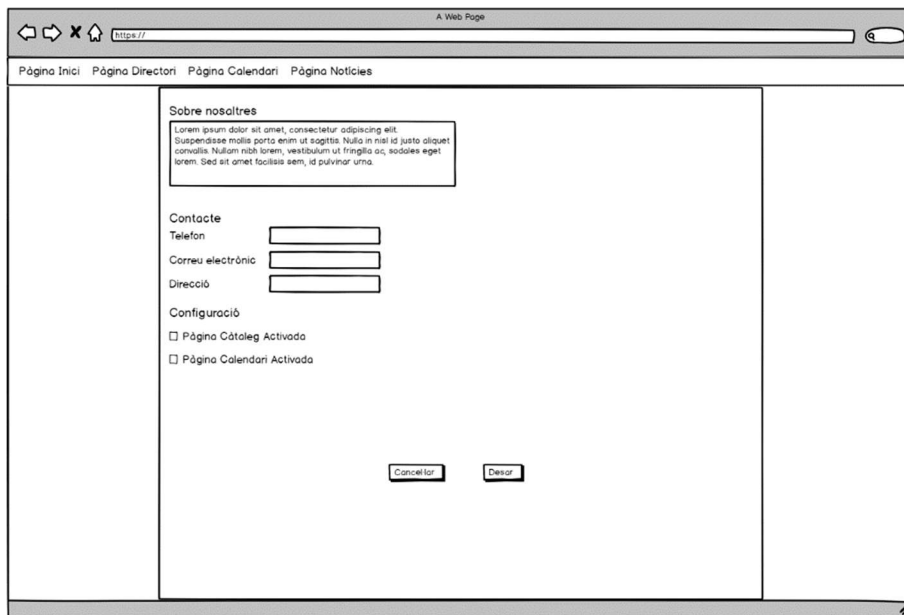


Figura 30: Template dashboard de l'agrupació

3.3.6 Feed de notícies de l'agrupació

La següent figura representa el prototip de la pàgina *feed* de notícies de l'agrupació.

A la *feed* de notícies de l'agrupació l'usuari podrà trobar les notícies tant de l'agrupació com de cada comerç.

A la *feed* de notícies, l'usuari hi podrà trobar:

- La barra de navegació de l'agrupació a la part superior de la pàgina.
- Un llistat de les notícies, cadascuna acompanyada d'una imatge, el titular i el subtítol.
- A la part dreta, una llista amb les últimes notícies per tal de que l'usuari pugui accedir ràpida i fàcilment a les últimes notícies.

Si l'usuari fa clic en alguna notícia de la llista, serà redirigit a la pàgina de la notícia ([apartat 3.3.12](#)).

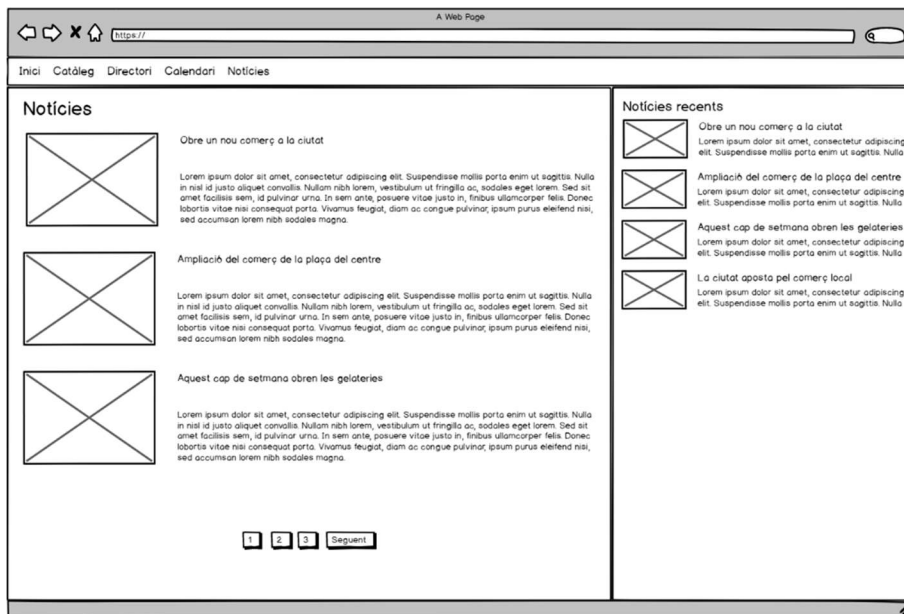


Figura 31: Template feed de notícies

3.3.7 Perfil del comerç

A la següent figura s’hi pot veure el prototip del perfil del comerç.

Des de aquesta pàgina l’usuari podrà veure les dades més rellevants del comerç i les últimes actualitzacions. També podrà trobar-hi una barra de navegació. Però en aquest cas li permetrà navegar per les pàgines del comerç.

Les pàgines accessibles des de la barra de navegació del comerç seran:

- [Catàleg del comerç.](#)
- [Calendari d’esdeveniments del comerç.](#)
- [Feed de notícies del comerç.](#)

La barra de navegació del comerç, com també la barra de navegació de l’agrupació, permetrà accedir al *dashboard* i iniciar i tancar sessió.

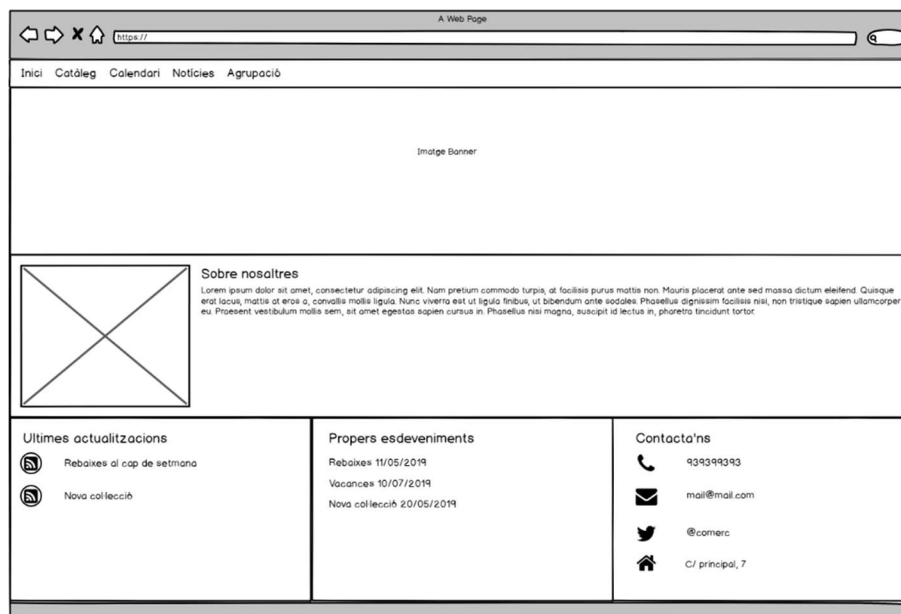


Figura 32: Template pàgina principal del comerç

3.3.8 Catàleg del comerç

Aquesta pàgina és molt similar a la pàgina **catàleg de l'agrupació** ([3.3.3 Catàleg de l'agrupació](#)), exceptuant:

- La barra de navegació, permetrà la navegació entre les pàgines del comerç.
- Els productes mostrats seran només aquells productes del comerç.

3.3.9 Pàgina d'esdeveniments del comerç

Aquesta pàgina i la **pàgina d'esdeveniments de l'agrupació** ([3.3.4 Pàgina d'esdeveniments de l'agrupació](#)) són iguals, però varien en:

- La barra de navegació que permetrà navegar per les pàgines del comerç.
- Els esdeveniments que s'hi mostraran (que seran només els del comerç).

3.3.10 Dashboard del comerç

El *dashboard* del comerç és com el *dashboard* de l'agrupació ([3.3.5 Dashboard de l'agrupació](#)). Tot i que es diferencien en un aspecte. El *dashboard* de comerç en comptes de tenir l'opció de gestionar comerços té l'opció de gestionar productes.

Si l'opció de gestionar productes es selecciona, el contingut del *dashboard*, serà un llistat amb tots els productes del comerç. Cada producte tindrà un botó al costat que permetrà eliminar el producte.

3.3.11 Feed de notícies del comerç

La *feed* de notícies del comerç, és semblant a la *feed* de notícies de l'agrupació (3.3.11 [Feed de notícies de l'agrupació](#)), però en aquest cas la barra de navegació permetrà navegar per les pàgines del comerç. En quant a les notícies, només s'hi mostraran les del propi comerç.

3.3.12 Pàgina de la notícia

La següent figura mostra el prototip de la pàgina de la notícia, que constarà de dues parts, el menú de navegació i el contingut de la pàgina.

El menú de navegació variarà segons d'on provingui l'usuari. Si l'usuari ve des de alguna de les pàgines del comerç, serà el menú de navegació per les pàgines del comerç, si ve d'una de les pàgines de l'agrupació, el menú permetrà la navegació per les pàgines de l'agrupació.

El contingut de la pàgina estarà format per dues parts, la notícia i una barra lateral a la part dreta amb les notícies recents per facilitar a l'usuari la navegació.

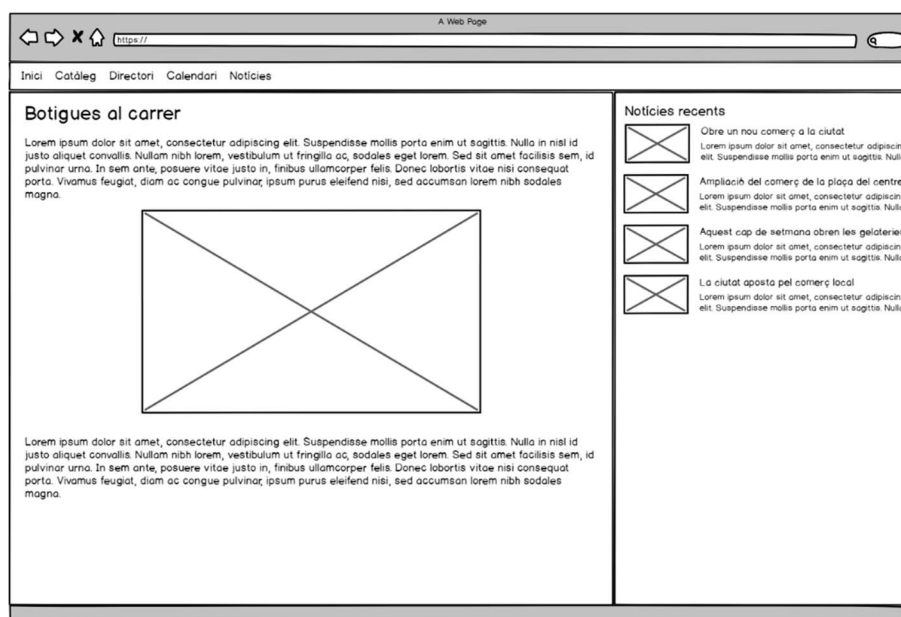


Figura 33: Pàgina d'una notícia

3.3.13 Pàgina d'un producte

A la següent figura, es pot observar la pàgina d'un producte.

A la pàgina es pot veure una barra de navegació, que variarà segons d'on provingui l'usuari.

L'usuari també hi podrà trobar tota la informació del producte i una barra lateral a la part dreta amb productes similars.

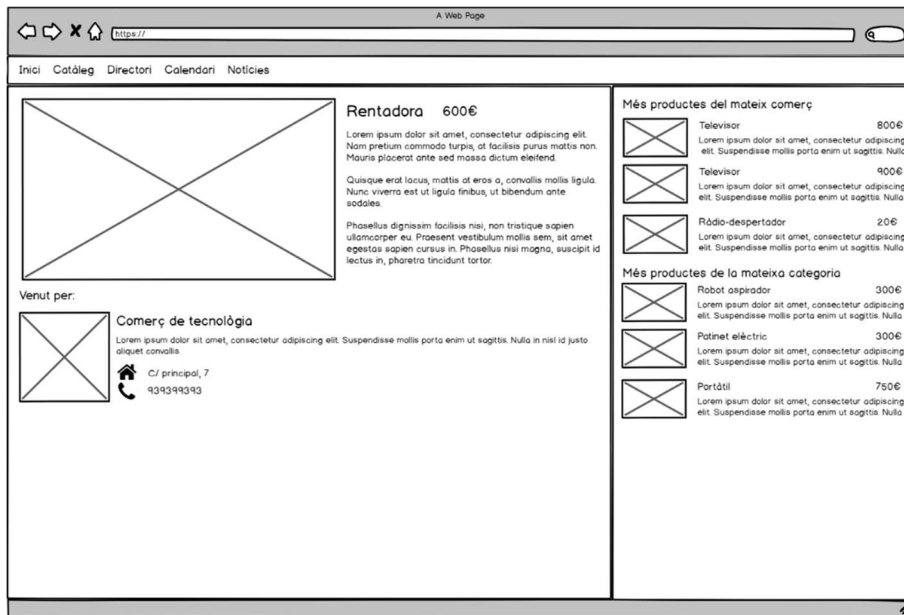


Figura 34: Pàgina d'un producte

3.3.14 Pàgina del esdeveniment

A continuació, s'hi pot veure el prototip de la pàgina del esdeveniment, en la qual la barra de navegació variarà segons d'on provingui l'usuari, com també succeeix a la pàgina d'una notícia i a la pàgina d'un producte.

L'usuari també hi podrà trobar la informació del esdeveniment i una barra lateral amb els esdeveniments en curs i els propers.

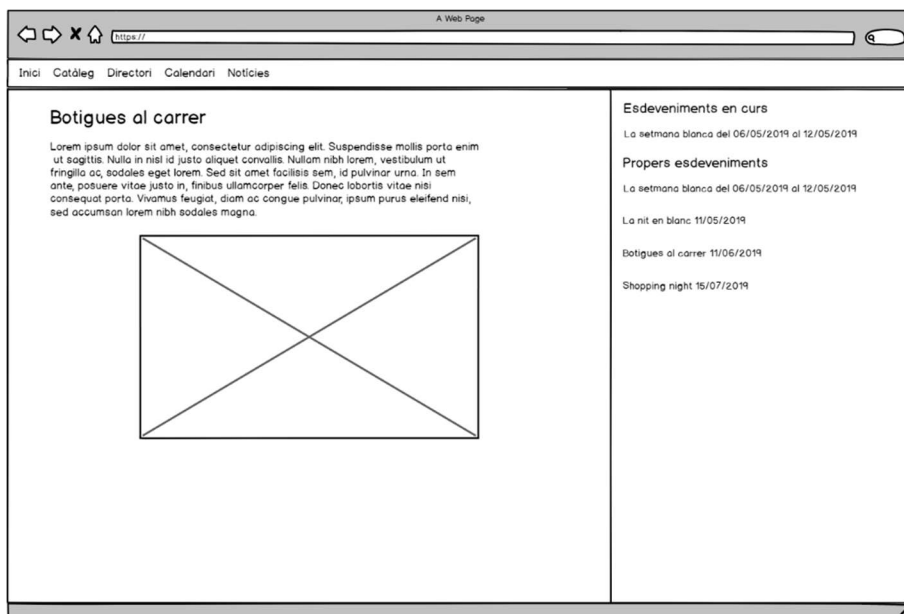


Figura 35: Pàgina d'un Esdeveniment

3.3.15 Templates de formularis

S'ha realitzat també dins del prototip un dels formularis que hi ha a la web. La raó per la qual només s'ha realitzat un dels formularis és que entre tots ells tan sols variaran en els continguts (les dades que es demanen i el tipus d'aquestes). Per tant tots els formularis seguiran el mateix patró de disseny.

El prototip mostra una finestra de navegador amb el títol "A Web Page" i una barra d'adreces que comença amb "https://". A sota de la barra d'adreces hi ha un menú de navegació amb les opcions "Inici", "Catàleg", "Directori", "Calendari" i "Notícies". El formulari principal està dividit en seccions: "Titular" amb un camp de text; "Cos de la notícia" amb un camp de text més gran; "Imatges" amb un botó "Afegeir" i dos quadrats amb una creu vermella a la cantonada superior dreta; i dos botons "Cancel·lar" i "Desar" a la part inferior.

Figura 36: Formulari nova notícia

4 Implementació

4.1 Organització dels fitxers

L'esquema de l'aplicació web, segueix l'estructura establerta pel propi *framework*. En aquest esquema s'hi poden trobar els següents directoris:

- *TFG*.
- *elocalcommerce*.
- *media*.
- *templates*.

4.1.1 Directori *TFG*

Aquest directori té dos arxius que cal destacar:

- ***settings.py***: aquest fitxer conté la configuració que s'aplicarà a totes les aplicacions desenvolupades dins d'aquest projecte.
- ***urls.py***: conté una relació entre tots els patrons de URL admeten les aplicacions dins del projecte i quina *view* s'executarà al rebre una petició.

4.1.2 Directori *elocalcommerce*

Dins del directori *elocalcommerce* hi ha dos subdirectoris i tres fitxers de codi *Python* importants per l'execució de l'aplicació web.

Els directoris que s'hi poden trobar són:

- ***migrations***: conté els fitxers generats pel *framework* un cop s'han realitzat les migracions de la base de dades amb les comandes *makemigrations* i *migrate*.
- ***static***: conté les icones que utilitzen els templates per generar pàgines. Per tal de poder accedir a les icones des dels templates, s'ha configurat l'adreça *STATIC_ROOT* del fitxer *settings.py* per a que apunti al directori actual. Un cop realitzada la configuració, des dels templates es podrà accedir mitjançant la paraula clau *static*.

I els fitxers de codi *Python* que hi ha són:

- ***models.py***: aquest fitxer té la definició de les classes que representaran el model de dades de l'aplicació. Totes les classes i relacions que s'hi defineixin es veuran reflectides a la base de dades un cop s'executin les comandes *makemigrations* i posteriorment *migrate*.

- **forms.py**: és un fitxer on hi ha definits els codis dels formularis que hi ha. El *framework Django* permet definir un formulari a partir dels models i personalitzar-lo.
- **views.py**: és el fitxer on hi ha la lògica de l'aplicació. Dins hi ha els mètodes que s'encarregaran de generar respostes a les peticions rebudes.

4.1.3 Directori *templates*

Dins del directori hi ha els fitxers *html* on s'hi defineix l'estructura de les diferents pàgines de l'aplicació web.

Per a guanyar en comprensió i reaprofitament del codi, cada fitxer té definida només una petita part de la pàgina. Així doncs, un fragment de codi *html* que es vulgui utilitzar en varies pàgines només haurà d'estar definit a un fitxer.

4.1.4 Directori *media*

El directori *media*, és on aniran tots els continguts multimèdia que es pugin al sistema. Cal remarcar que per a que sigui així s'ha modificat el valor de *MEDIA_URL* del fitxer *settings.py* per a que apunti a aquest directori.

4.2 Model

La base de dades que s'utilitza en aquest projecte és la que el *framework Django* utilitza per defecte, es tracta de *SQLite*, un sistema gestor de bases de dades (SGBD).

SQLite utilitza el model relacional per a desar les dades, aquest sistema es basa en l'ús de relacions entre entitats.

Com s'ha comentat prèviament, el propi *framework* genera automàticament la base de dades a partir de les classes definides al fitxer *models.py*, i per tant també genera les relacions al nivell de la base de dades.

Al projecte s'hi poden trobar dues relacions entre classes del model, la relació *Many-to-one* i la relació *One-to-one*

4.2.1 Classes del model

Per a que el *framework* generi les taules i relacions entre elles a la base de dades, cal definir les classes al fitxer *models.py*. A partir del fitxer *Django* crearà les taules amb els camps necessaris. El tipus de dades de cada camp dependrà dels atributs que se li posin a la classe que es defineix al model.

Django permet l'ús d'una sèrie de classes per a establir de quin tipus serà l'atribut. Les classes que s'han utilitzat al projecte són:

- ***CharField.***
- ***TextField.***
- ***EmailField.***
- ***DecimalField.***
- ***DateTimeField.***
- ***ImageField.***
- ***ForeignKey.***
- ***OneToOneField.***

4.2.2 *ForeignKey.* Relació *Many-to-one*

Django permet crear relacions molts a un (*Many-to-one*) entre dues classes del model utilitzant la classe *ForeignKey*.

Per defecte, *Django* estableix el comportament que tenen aquestes relacions al esborrar-se com a cascada. És a dir, si s'esborra l'objecte unitari, també s'esborren els objectes amb els que està relacionat.

Però hi ha la possibilitat de modificar el comportament, per exemple establint que si s'esborra l'objecte unitari, la *ForeignKey* es canviï a un valor nul a totes les seves relacions.

Tot i que hi ha més possibles actuacions per quan s'esborra l'element unitari. Al projecte s'utilitzen només els dos comportaments esmentats.

Per exemple, una categoria pot estar relacionada amb varis productes o comerços. Si una categoria es esborrada, no es pretén esborrar cada un dels productes o comerços que s'hi relacionen, es per això que la *ForeignKey* de la categoria es posa en un valor nul.

En canvi, un comerç pot estar relacionat amb múltiples productes. És per això que si s'esborra un comerç, és vol que també s'esborrin els seus productes. En aquest cas el comportament emprat és el comportament per defecte.

4.2.3 *OneToOneField.* Relació *One-to-one*

Aquest tipus de relació permet generar relacions d'un a un (*One-to-one*) entre dues classes del model.

Aquest tipus de relació, com la relacions *Many-to-one*, també té per defecte el comportament d'esborrar-se com a cascada. Això comportarà que si s'esborra un dels objectes, la seva relació s'esborra també.

Aquesta relació també té més possibles actuacions, però al projecte només s'utilitza el comportament esmentat.

Dins del projecte, s'utilitza per relacionar la classe *User* (que proporciona el *framework Django*) amb una classe *Perfil* creada al projecte.

Fent ús d'aquesta relació permet ampliar les dades que es tenen per a cada usuari, en aquest cas permet establir quin element pot gestionar l'usuari. Per tant, en el cas de que s'esborrés o bé el perfil o bé l'usuari, es vol que s'esborri l'objecte que té relacionat també.

4.2.4 Consultes al model

Tot i que es poden fer consultes amb *SQL*, el *framework* proporciona una *API* que permet utilitzar la base de dades. Les crides a la *API* generen una crida *SQL* que interacciona amb la base de dades i retorna un o una llista d'objectes. Les crides a la *API* es fan de la següent manera:

```
Comerc.objects.get(nom=nomComerc)
```

Figura 37: exemple d'una crida a la API

On **Comerc** és el nom de la classe que s'ha definit al model. **get** és el mètode que es crida.

Els mètodes de la *API* que s'han utilitzat al projecte són:

- **get**. Requereix d'una o més condicions. Retorna un objecte que compleixi rebudes. En el cas de que hi hagi més d'una coincidència, aquesta crida llença una excepció.
- **filter**. Requereix d'un *QuerySet* (resultat d'una *query*) i d'una o més condicions. Retorna totes les coincidències que compleixen les condicions rebudes.
- **all**. No requereix de cap paràmetre. Retorna tots els objectes del model.
- **order_by**. Requereix d'un *QuerySet* i paràmetres. Retorna un *QuerySet* ordenat segons els paràmetres rebuts.
- **exclude**. Requereix d'un *QuerySet* i condicions. Exclou del *QuerySet* els objectes que compleixen les condicions i en retorna el *QuerySet* resultant.
- **count**. Requereix d'un *QuerySet*. Retorna el número d'objectes que hi ha al *QuerySet*.
- **create_user**. Requereix d'un nom d'usuari i d'un *password*. Crea un nou objecte de la classe de *Django User*.

4.3 Formularis de *Django*.

Al lloc web es permet als usuaris introduir dades mitjançant formularis *HTML*. Aquests formularis serviran per crear o modificar classes del model. Per crear aquests formularis s'han utilitzat les eines i llibreries que incorpora el *framework Django* per facilitar la creació i posteriorment el processat d'aquests formularis.

A l'arxiu *elocalcommerce/forms.py* s'hi ha definit una classe per cada formulari del lloc web, aquestes classes tenen especificat:

- **model:** nom de la classe del model.
- **fields:** aquest camp indica quins atributs de la classe del model es mostraran al formulari.
- **labels:** és un diccionari amb el nom del atribut de la classe del model i la descripció que apareixerà al renderitzar el formulari.
- **exclude:** un llistat de tots els atributs que es volen excloure al formulari.

Des de la lògica de l'aplicació es poden referenciar i crear noves instàncies d'aquests formularis, per després enviar-los als *templates* per a incloure'ls com a *HTML*.

Quan un usuari utilitza un formulari i introdueix una dada incorrecte o es deixa algun camp requerit per omplir, la pàgina no permet enviar el formulari i marca els errors. Tanmateix les comprovacions a la part del client no aporten seguretat al sistema i per això no només es realitzen les comprovacions de les dades al client. A la part del servidor, els formularis *Django* incorporen un mètode que permet comprovar si un formulari conté o no dades correctes.

4.4 Gestió de les imatges pujades

A l'aplicació web hi ha varies classes del model que tenen alguna imatge com a atribut. Per a definir aquests atributs, s'ha utilitzat la classe *ImageField*.

Totes aquestes imatges es desen en subdirectoris del directori *media* depenent a on pertanyin.

Per aconseguir això s'ha establert el directori *media* com al directori on s'hi desen els elements multimèdia per defecte, modificant el valor *MEDIA_ROOT* del fitxer de configuració *settings.py*.

Un cop s'ha configurat el directori per a que s'hi desin els continguts multimèdia, s'ha d'indicar a cada classe del model on guardar les imatges. Si no el contrari tota la multimèdia que es pugui anirà al directori on hi apunti el valor de *MEDIA_ROOT*.

Per tal d'indicar a quin directori s'han de desar les imatges s'ha de passar un valor al paràmetre *upload_to* de la classe *ImageField*. Mitjançant aquest paràmetre es pot indicar el directori on s'hi guardaran les imatges. Tot i que el camí del directori serà el camí resultant d'ajuntar la direcció *MEDIA_ROOT* i el valor del paràmetre *upload_to*.

4.5 Interfície

Per la implementació de la interfície s'ha utilitzat:

- *HTML*.
- El *framework Bootstrap* de *CSS*.
- Els filtres i etiquetes de templates que incorpora el *framework Django*.
- La llibreria *Crispy-Forms*.

4.5.1 *HTML* i *Bootstrap*

Amb *HTML* i *Bootstrap* s'han creat, seguint el prototip detallat al apartat de disseny, les plantilles de cada una de les pàgines que hi ha al lloc web. Totes les plantilles són *responsive* gràcies a les classes que facilita *Bootstrap*.

S'ha pres la decisió d'utilitzar *Bootstrap* ja que avui en dia hi ha molta gent que accedeix a llocs web mitjançant telèfons o tauletes. *Bootstrap* permetia crear una aplicació web amb una bona estètica des d'aquest tipus de dispositius.

A les següents figures, s'hi pot veure dues captures de la mateixa pàgina amb mesures diferents de la pantalla. Com es pot comprovar, el contingut de la pàgina s'ajusta segons el tamany de la pantalla.

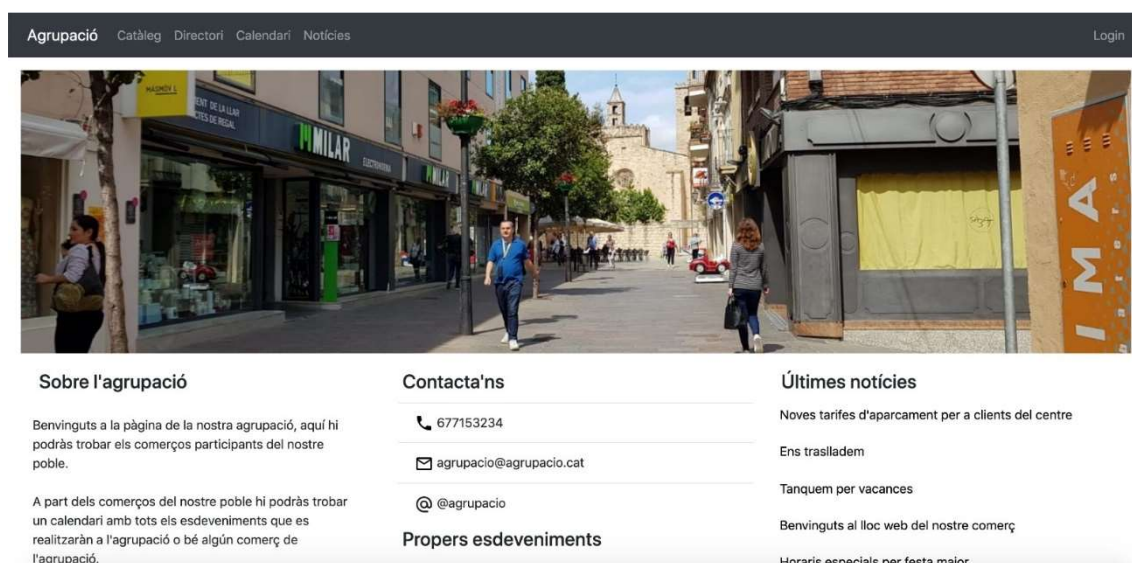


Figura 38: perfil de l'agrupació



Sobre l'agrupació

Benvinguts a la pàgina de la nostra agrupació, aquí hi podràs trobar els comerços participants del nostre poble.

A part dels comerços del nostre poble hi podràs trobar un calendari amb tots els esdeveniments que es realitzaran a l'agrupació o bé algún comerç de l'agrupació.

També hi podràs trobar una secció amb les notícies sobre l'agrupació i sobre els comerços per estar així plenament informat.

Esperem que us agradi el nostre lloc web.

Contacta'ns

Figura 39: perfil de l'agrupació amb pantalla petita

4.5.2 Filtres i etiquetes de templates

El framework Django incorpora un seguit de filtres i etiquetes que permeten omplir plantilles amb dades generades a la lògica de l'aplicació web d'una manera senzilla.

La utilització d'aquests filtres és molt senzilla, es generen un seguit de dades a la lògica de l'aplicació que es passaran als *templates*. Un cop les dades són als *templates*, es pot fer, per exemple:

- Blocs d'iteracions per generar blocs de codi similars que s'han de construir a partir d'un conjunt de dades.
- Comparacions dels valors que s'han rebut de la lògica. Per saber si s'ha de mostrar alguna cosa.
- Renderitzacions de formularis.
- Comprovar si s'han rebut dades en una certa variable.

Utilitzant els filtres i etiquetes, s'ha d'escriure menys codi HTML i permet una llegibilitat d'aquest codi molt més elevada.

Mitjançant els filtres i les etiquetes dels templates que incorpora Django, s'omplen les pàgines amb els continguts generats a la vista. Per exemple, a la següent figura es visualitzen tant les categories i els productes que es reben de la vista amb els filtres i etiquetes.

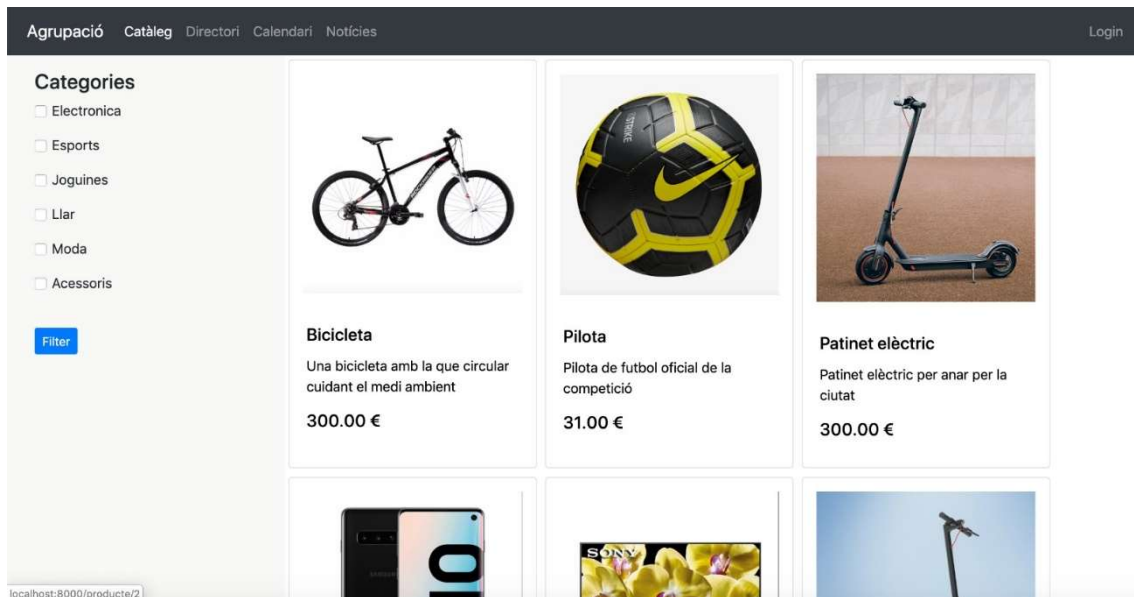


Figura 40: Catàleg

Al següent exemple, s'hi pot veure un llistat dels comerços que hi ha a l'agrupació.

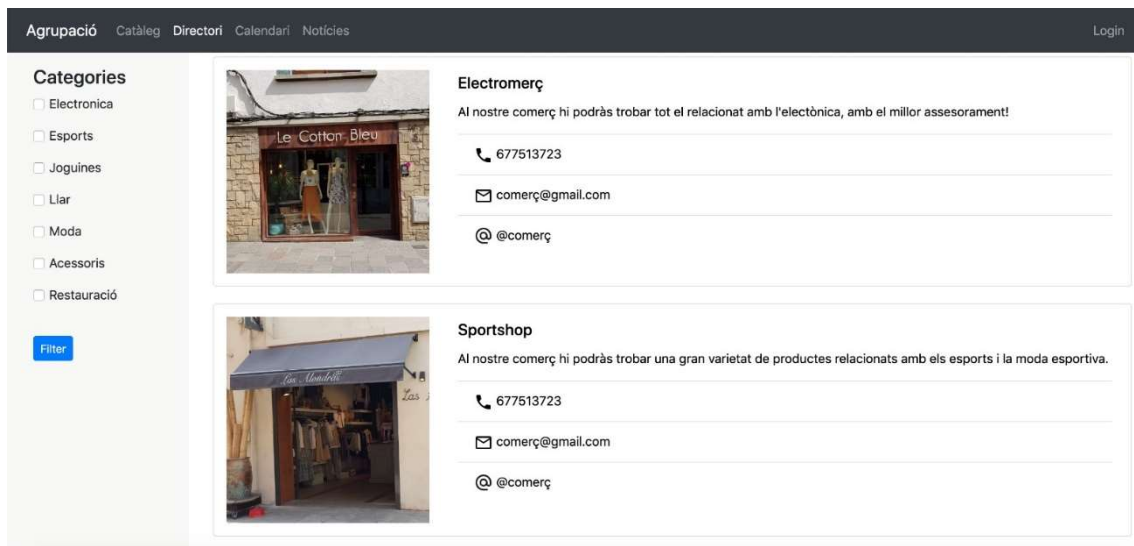


Figura 41: Directori

4.5.3 Llibreria Crispy-Forms

Com s'ha exposat anteriorment al apartat dels formularis *Django*, els formularis del lloc web que serveixen per crear o modificar dades d'un objecte del model, es generen a partir de les classes definides al arxiu *forms.py*.

Aquests formularis es creen des de la lògica de l'aplicació i es passen per context als *templates*, des d'on són renderitzats com a *HTML*. Els fragments *HTML* generats als templates no feien ús de les classes ni estils de Bootstrap, per tant no segueixen l'estètica del lloc web.

La llibreria *Crispy-Forms*, afegeix una etiqueta a les etiquetes de *templates* de *Django*. Aquesta etiqueta varia el comportament de la renderització dels formularis a codi *HTML* i el que fa es renderitzar el mateix formulari però utilitzant les classes de *Bootstrap*. Així s'aconsegueix poder seguir utilitzant les llibreries i eines pels formularis que inclou *Django* conservant l'estètica del lloc.

A continuació s'hi pot veure una comparació entre els formularis generats per *Django* i els formularis de *Django* aplicant la llibreria *Crispy-Forms*.

La següent figura, és una captura d'un formulari generat per *Django*. Com es pot comprovar no comparteix l'estètica del lloc web.

The screenshot shows a web interface with a navigation bar at the top containing 'Exit Dashboard', 'Dades de l'agrupació', 'Directori', 'Notícies de l'agrupació', and 'Esdeveniments de l'agrupació'. On the right side of the navigation bar, there is a user profile 'gestorAgrupacio' with a pencil icon and a 'Log out' button. The main content area is titled 'Dades de l'agrupació' and contains the following fields and text:

- Nom:** A text input field containing 'Agrupació del poble'.
- Descripció:** A large text area containing the text: 'Benvinguts a la pàgina de la nostra agrupació, aquí hi podràs trobar els comerços participants del nostre poble. A part dels comerços del nostre poble hi podràs trobar un calendari amb tots els esdeveniments que es realitzaran a l'agrupació o bé algun comerç de l'agrupació. També hi podràs trobar una secció amb les'.
- Telefon:** A text input field containing '677153234'.
- Email:** A text input field containing 'agrupacio@agrupacio.cat'.
- Xarxa social:** A text input field containing '@agrupacio'.
- ImatgeBanner:** A label indicating the current image is 'imatgesBanner/imatgeAgrupacioBanner.jpeg'.
- Change:** A file upload section with a 'Seleccionar archivo' button and the text 'Ningún archivo seleccionado'.
- Save:** A blue button at the bottom of the form.

Figura 42: formularis Django per defecte

La següent figura, hi podem veure el mateix formulari però aplicant la llibreria *Crispy-Forms*.

The screenshot shows the same web interface as the previous figure, but the form is styled using *Crispy-Forms*. The navigation bar and user profile are identical. The main content area is titled 'Dades de l'agrupació' and contains the following fields and text:

- Nom*:** A text input field containing 'Agrupació del poble'.
- Descripció*:** A large text area containing the text: 'Benvinguts a la pàgina de la nostra agrupació, aquí hi podràs trobar els comerços participants del nostre poble. A part dels comerços del nostre poble hi podràs trobar un calendari amb tots els esdeveniments que es realitzaran a l'agrupació o bé algun comerç de l'agrupació. També hi podràs trobar una secció amb les notícies sobre l'agrupació i sobre els comerços per estar així plenament informat. Esperem que us agradi el nostre lloc web.'
- Telefon*:** A text input field containing '677153234'.
- Email*:** A text input field containing 'agrupacio@agrupacio.cat'.

Figura 42: formularis amb Crispy-Forms

4.6 Views, la lògica de l'aplicació

La lògica de l'aplicació esta definida al fitxer *views.py*. El fitxer està format per una sèrie de funcions, que segons el conveni de *Django*, cada una d'aquestes funcions s'anomena vista (*view*). Tot i que es podrien definir vistes a altres fitxers o directoris, totes les vistes de l'aplicació web s'han col·locat al fitxer *views.py* com recomana el conveni del *framework*.

Les vistes són funcions que reben com a mínim un paràmetre i retornen una resposta.

Totes les vistes han de rebre com a mínim un objecte *HttpRequest* com a paràmetre, però també s'hi poden afegir més segons es necessitin per a l'execució de la funció.

L'objecte *HttpRequest* és un objecte *read-only* que conté la informació de la petició HTTP que s'ha realitzat al servidor.

Una vista no ha de retornar sempre una renderització d'una pàgina *HTML*, sinó que pot retornar qualsevol altra cosa. Al projecte però, les vistes retornen codis del protocol *HTTP* i renderitzacions *HTML*.

Quan es fan renderitzacions de fitxers *HTML*, es passen dades de les vistes als *templates*. Per enviar dades als templates des de la vista es fa amb un diccionari on la *key* és el nom de la variable, i el *value* es el valor. Així doncs quan es vulgui fer referència a una variable des dels *templates* es farà amb les *keys* del diccionari.

4.7 URLs

Fins ara s'ha comentat com s'ha definit la lògica i com es generen els arxius *HTML* a partir de les dades que s'han generat a la lògica. Però per saber quina vista s'ha d'executar al rebre una petició *HTTP* a una determinada direcció *URL* s'ha d'explicar el que succeeix quan es rep una petició.

Al rebre una petició amb una *URL*, el servidor comprova si hi ha algun patró de *URL* que coincideixi amb la *URL* rebuda. Si existeix alguna patró que coincideixi amb la *URL*, s'executa la vista assignada per a aquella direcció, si no es troba cap patró que coincideixi amb la *URL*, es genera una pàgina d'error.

Django busca a el llistat de patrons de *URLs* a la ruta *ROOT_URLCONF* que hi ha al fitxer de configuració *settings.py*. Per conveni del framework, el llistat de patrons de *URLs* de l'aplicació es defineix al fitxer *urls.py*.

El fitxer *urls.py*, conté una relació de llista de patrons de *URLs*, la vista a executar, un diccionari on col·locar els paràmetres per la vista i un nom.

El nom que es defineix per cada patró, serveix per crear una direcció *URL* que compleixi el patró en qualsevol part de l'aplicació, tant en les vistes com als *templates*.

4.8 Permisos

Com s'ha comentat a la introducció i al disseny, el lloc web permet crear varis perfils. Aquests perfils no poden realitzar accions sobre tots els objectes que hi ha al model, només podran gestionar tots aquells objectes del model que els pertanyin.

Per exemple, els perfils que gestionin l'agrupació només podran

- Modificar les dades d'aquesta.
- Afegir un comerç nou.
- Crear/modificar notícies d'aquesta.
- Crear/modificar esdeveniments d'aquesta.



Figura 43: barra de navegació del Dashboard de l'agrupació

En canvi, els perfils que gestionin comerços podran:

- Modificar les dades del comerç que gestionen.
- Crear/modificar els productes del comerç que gestionen.
- Crear/modificar les notícies del comerç que gestionen.
- Crear/modificar els esdeveniments del comerç que gestionen.



Figura 44: barra de navegació del Dashboard del comerç

Per tant no tots els perfils podran realitzar peticions sobre determinades direccions URL.

Per exemple, els usuaris que no tinguin una sessió iniciada, no podran accedir totes les funcions esmentades. Per a evitar que succeeixi això, el *framework Django* té un sistema de verificació que permet evitar que certes vistes siguin executades per usuaris que no tenen la sessió iniciada.

Per definir que una vista no pugui ser executada s'utilitza el decorador de vista "*@login_required*". Si un usuari sense una sessió iniciada intenta entrar a una URL que executa una vista amb aquest decorador, *Django* el redirigirà a la URL que hi ha definida al paràmetre *LOGIN_REDIRECT_URL* del fitxer *settings.py*.

Però no només s'ha d'evitar que un usuari sense una sessió vàlida iniciada pugui executar certes vistes, sinó que també s'ha d'evitar que un perfil modifiqui dades que no pot. Per aconseguir això, al projecte s'utilitzen dos maneres diferents.

La primera manera, utilitza el decorador de la vista "*@user_passes_test*". El que fa el decorador és executar un mètode abans d'executar la vista, només si el mètode retorna un booleà *True*, s'executarà la vista. Aquest decorador té dos paràmetres, primer se li passa el nom del mètode que s'executarà i una URL on s'haurà de redirigir al usuari si no passa el test.

Amb el decorador, no es poden passar paràmetres al mètode que es crida abans d'executar la vista, aquest mètode només rebrà automàticament un paràmetre indicant el usuari que té la sessió iniciada.

Per determinar si un usuari pot editar o no un producte, una notícia o un esdeveniment, és necessària més informació. Per tant, per aquests casos no és pot utilitzar aquest decorador.

La segona forma que s'utilitza a l'aplicació web per a evitar que una vista sigui executada per un usuari que no pot, és la creació d'unes funcions auxiliars per determinar si s'executarà o no la vista. Aquestes funcions, rebran les dades necessàries per determinar si es pot executar o no la vista i retornaran un booleà indicant si pot executar o no la vista.

Al principi de l'execució de les vistes que requereixen d'aquests permisos, s'obtindran les dades necessàries i es cridarà a la funció auxiliar. Si l'usuari pot executar la vista, continuarà amb l'execució, en canvi, si l'usuari no pot executar-la serà redirigit.

5. Proves i resultats

5.1 Avaluacions del *sprints*

Un cop es donava per finalitzat un *sprint* es comprovava si tot el que s'havia realitzat era correcte, si era així es donaven per finalitzades les tasques d'aquest.

Durant els 4 primers *sprints*, s'ha realitzat la planificació, disseny i estimació de les tasques que s'anaven a realitzar en els futurs *sprints*.

El quart *sprint* es va estimar de manera incorrecte, donat que realitzar varis prototips prou bons requeria de més temps del estimat. Així que al finalitzar el període, no es va acabar de realitzar totes les tasques que s'havien planificat. En concret, no es va poder instal·lar i posar en marxa l'entorn de desenvolupament.

El cinquè *sprint*, també estava mal estimat. Com era el primer cop que s'utilitzaven els formularis de *Django*, a aquesta tasca se li va estimar més temps del necessari. Però com aquest *sprint* es va començar realitzant la tasca "instal·lar i posar en marxa l'entorn de desenvolupament" del *sprint* anterior, al final del *sprint* es va acabar correctament.

El sisè *sprint*, es va estimar correctament. Ja que per a la creació d'usuaris ja hi havia contemplat el fet de fer una recerca i realitzar proves de varis mètodes d'ampliació de les dades de la classe *User* de *Django*.

El setè *sprint*, es va complir en el temps previst però no es va estimar bé. El fet de que no s'estimés bé és que les proves van trigar menys temps del esperat, però la gestió dels permisos va resultar més complicada i llarga del esperat. La gestió dels permisos, en principi s'anava a realitzar amb els decoradors "@user_passes_test" de *Django*. Però com s'ha descrit a la memòria, no tots els permisos que requereix l'aplicació es poden realitzar amb aquest decorador, es per això que es va haver de realitzar una cerca per a trobar una forma alternativa de complir la gestió de permisos.

Al vuitè i penúltim *sprint*, hi havia estimat unes hores per a posar al dia tasques que no s'haguessin pogut finalitzar o realitzar alguna tasca que es considerés correcte per l'aplicació. Com s'anava al dia amb les tasques es va decidir utilitzar aquest temps per a realitzar dues millores:

- Afegir la llibreria *Crispy-Forms*, per tal de que els formularis implementessin les classes de Bootstrap i mantenir una estètica al lloc web.
- Afegir alertes després de trametre formularis. Fins ara, si un formulari es rebia i era correcte, l'usuari era redirigit i no sabia si s'havia realitzat la operació o no. Per tant, es va incloure que a la pàgina on anés redirigit l'usuari, es mostrés una alerta si s'havia realitzat la operació. Els casos en que el formulari no sigui correcte, el sistema mostra l'error, per tant no feia falta afegir res en aquests casos.

A l'últim *sprint* s'havia d'acabar de realitzar la memòria amb els punts que restaven.

6. Conclusions

6.1 Conclusions

Un cop acabat el projecte, considero que els objectius que es van plantejar en un principi s'han pogut assolir, gràcies als coneixements que he anat adquirint i posant en pràctica.

He pogut comprovar la importància de fer un anàlisi, un plantejament, un disseny i una planificació adequada del software. Ja que si no es realitza correctament pot dificultar-ne la implementació. Cal destacar que aquest problema creix com ho fa el tamany del projecte. Donat que com més creixia el projecte, més grans es feien aquests problemes.

Considero que he après molt a dissenyar i planificar la implementació d'un software.

7. Futures ampliacions

7.1 Futures ampliacions

Aquest projecte pretén assentar les bases d'una plataforma per a que els comerços locals puguin tenir un lloc web sense molt esforç ni dedicació.

Després de la implementació i les proves al lloc web, s'ha pogut comprovar que hi ha aspectes que es podrien millorar respecte a com han estat dissenyades.

A part de les millores, el lloc web també és ampliable.

Les futures ampliacions que es contemplen són:

- **Millorar del *look and feel*:**
 - Millora dels estils i permetre que els usuaris puguin canviar l'estil tant del lloc web del comerç com el de l'agrupació.
 - Millora de la interacció del usuari, afegint un buscador.
- **Millora en el registre d'un nou comerç:** Enviar un correu electrònic amb una direcció web per a que l'usuari pugui crear una conta un nom d'usuari i una contrasenya.
- **Gestió d'estocs:** Permetre una gestió d'estocs als comerços, així es podria deixar de mostrar productes que estiguessin fora d'estoc.
- **Gestió de reserves:** Possibilitat de crear reserves d'un producte per part d'un client.
- **Sitemap:** afegir el *sitemap* del lloc web per a que es puguin indexar els continguts de la pàgina web als motors de cerca.

8. Glossari

Django: És un framework per al desenvolupament d'aplicacions web. Permet la creació de llocs web utilitzant el patró *MVT (Model-View-Template)*.

Framework: Un marc de treball que simplifica el desenvolupament de software. Per a fer-ho, un framework aporta una sèrie de convenis, patrons i llibreries.

HTML: *HyperText Markup Language*, és un llenguatge de programació amb etiquetes que permet donar una descripció i característiques a tot allò que conté. És el llenguatge més utilitzat per programar pàgines web.

Templates: Arxiu on hi ha la definició de l'aspecte i com es mostraran els continguts d'una pàgina web.

9. Bibliografia

Django Software Foundation. (2019). *Django documentation*. [En línia]. [Consulta: 2 de Març de 2019]. Disponible a: <https://docs.djangoproject.com/en/2.2/>

Python Software Foundation. (2016). *Python documentation*. [En línia]. [Consulta 2 de Març de 2019]. Disponible a: <https://docs.python.org>

Bootstrap Team. (2019). *Bootstrap documentation*. [En línia]. [Consulta 5 de Març de 2019]. Disponible a: <https://getbootstrap.com/>

Miguel Araujo and Daniel Greenfield. (2013). *Crispy-Forms docs*. [En línia]. [Consulta 28 de Març de 2019]. Disponible a: <https://django-crispy-forms.readthedocs.io/en/latest/>

W3C Editor's Draft. (2010). *HTML5 Reference*. [En línia]. [Consulta 5 de Març de 2019]. Disponible a: <https://dev.w3.org/html5/html-author/>