

# Projecte Gestió de precintes de vehicles

**Bernat Ginard Prats**  
GRAU ENGINYERIA INFORMÀTICA

**Albert Grau i Perisé**

14 de Juny del 2017

A la meva dona i als meus fills,

Gràcies a tots pel vostre suport en les hores baixes, per la vostra comprensió i per tenir la paciència necessària en aquest camí llarg que hem recorregut junts durant tota la carrera.

## **Agraïments**

A tota la comunitat de l'UOC, gràcies per totes les hores que hem passat junts.

## Resum

El present treball final de carrera consisteix en desenvolupar una aplicació que gestioni el parc de vehicles per una administració local. Gràcies a aquesta gestió, l'administració local augmentarà l'eficiència i millorarà els processos interns de diferents seccions. Es tracta d'un projecte real, el qual s'ha migrat a una tecnologia JAVAE.

El mètode seguit per dur a terme el projecte ha sigut un mix de procés en cascada amb un procés iteratiu que ens permet detectar i arreglar els inconvenients que van sorgint, sigui quina sigui la etapa a la que ens trobem. Recordem que el procés en cascada es pot desglossar en: Pressa de requisits, anàlisi i disseny, implementació i proves.

El llenguatge que s'ha fet servir a la implementació ha sigut Java i el projecte s'ha estructurat en un projecte Maven ja que ens facilita la gestió de les dependències i a la creació del producte final (executable).

L'arquitectura més adient per dur aquest projecte és una barreja d'arquitectures entre les que destaquem: arquitectura client servidor i una arquitectura heterogènia organitzat per capes en que el client només és l'encarregat de executar el programari de presentació i el servidor és l'encarregat de manejar la lògica de negoci i la persistència amb la base de dades.

Finalment, com a **patró de disseny** s'ha seguit el patró MVC i els **frameworks** utilitzats en aquest projecte i que han ajudat en el seu desenvolupament ha sigut: *SpringBoot v 1.4.6*, *Thymeleaf* com a motor de plantilles html, *BootsTrap* com gestor d'estils, *Spring JPA 2.1* i *QueryDsl* per manejar la persistència i les consultes de base de dades. Aquesta versió de *Spring JPA* maneja la versió 5 de *hibernate*.

## Índex

1. Introducció .....	5
1.1. Context i justificació del Treball. ....	5
1.2. Objectius del TFG. ....	5
1.3. Enfocament i mètode seguit. ....	5
1.4. Planificació. ....	7
1.5. Producte obtingut. ....	8
1.6. Descripció de la resta de capítols. ....	8
2. Descripció del projecte. ....	9
3. Especificació del projecte. ....	10
3.1. Especificació per subsistemes. ....	10
3.2. Seguretat. ....	11
4. Anàlisi funcional. ....	11
4.1. Actors. ....	11
4.2. Diagrama de casos d'ús. ....	12
4.3. Fitxes de casos d'ús. ....	12
5. Disseny. ....	17
Diagrama de classes JPA. ....	17
Diagrama del model Base de dades. ....	18
Diagrama de classes DTO. ....	19
Diagrama d'arquitectura. ....	20
6. Implementació. ....	24
6.1. Eines de desenvolupament. ....	24
6.2. Frameworks utilitzats. ....	25
6.3. Proves unitàries. ....	27
7. Interfícies d'usuari. ....	27
8. Instal·lació i execució .....	33
9. Glossari. ....	35
10. Conclusions. ....	36
11. Bibliografia y referències. ....	36

## 1. Introducció

### 1.1.Context i justificació del Treball.

Les administracions locals són les encarregades de gestionar els tributs municipals de vehicles, urbana, activitats econòmiques, entre d'altres. Tenir un cens de vehicles acurat permet que l'administració pugui recaptar de forma eficient els impostos i que aquests ingressos siguin repercutits en benefici de tots els ciutadans.

Tenir les ordres de precinte dels vehicle registrades és una necessitat legal que l'entitat local ha de complir, ja que, des de altres entitats, normalment judicials, arriben ordres de precinte contra vehicles que per llei s'han d'executar. No tal sols són entitats judicials les que emeten ordres de precinte, altres entitats, tant internes com externes ho poder fer.

Un cop emesa l'ordre de precinte, una secció de la pròpia entitat local s'encarrega de registrar-la i l'autoritat (policia local) rep una ordre de fer una recerca del vehicle pel carrer i quan el vehicle es localitza s'emet una ordre de retirada cap a un dipòsit. Un cop el vehicle s'hagi retirat a un dipòsit, no es permet lliurar el vehicle fins que arribi l'ordre d'aixecar el precinte.

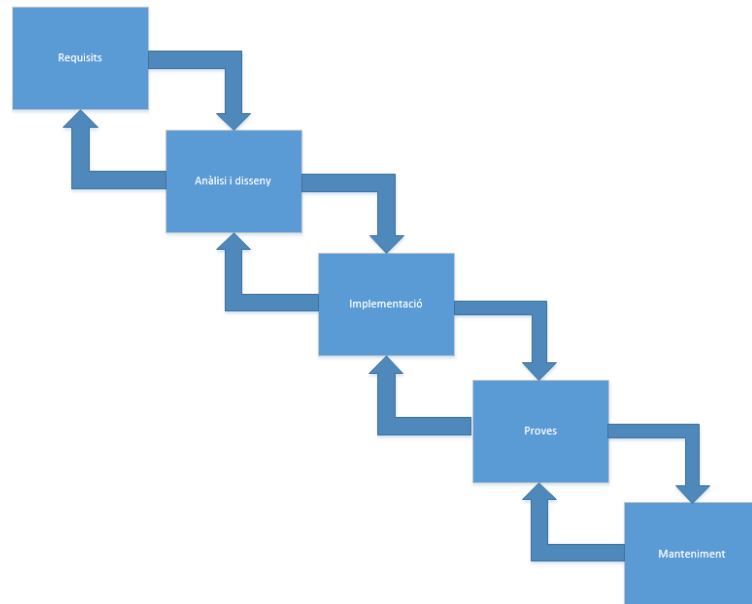
### 1.2.Objectius del TFG.

L'objectiu del treball és desenvolupar un projecte que gestioni els precintes dels vehicles. Aquest projecte permetrà augmentar l'eficiència i funcionament a diverses seccions de l'entitat local. D'una banda, la secció que ha de gestionar els precintes tindrà una eina per gestionar el seu registre, i d'altra banda, altres seccions o negociats es veuran beneficiades ja que si s'ha emès una ordre de precinte i el vehicle és localitzat i retirat a un dipòsit no es podrà retirar fins que el propietari del vehicle pagui totes les deutes que té amb l'Ajuntament.

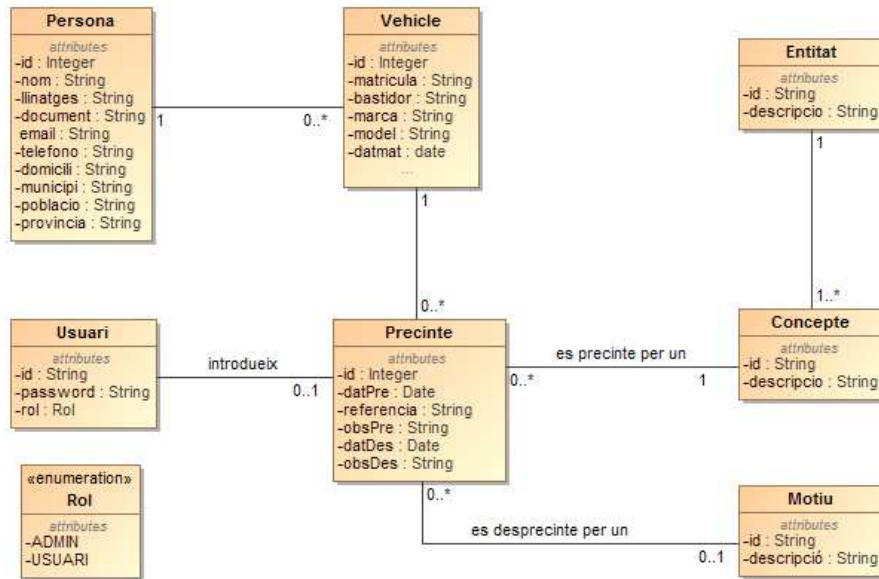
### 1.3.Enfocament i mètode seguit.

La metodologia a utilitzar en aquest projecte és el cicle de vida en cascada. Aquesta mètode agrupa les tasques a fer a cada fase de forma ben diferenciada. Així i tot, com que sabem que aquesta metodologia es poc flexible als canvis utilitzarem un procés iteratiu ja que es preveu

que ens ajudarà a resoldre defectes que s'hagin produït en les fases anteriors durant el mateix desenvolupament.



Com a punt de partida i per tenir un punt de vista de la informació segons el model de referència pel model obert i distribuït (RM-ODP), es mostra un diagrama de classes on es veuen reflectides les entitats del domini i com aquestes entitats es relacionen entre si.



### 1.4. Planificació.

Seguint la planificació del TFG podem dividir el projecte en quatre fases:

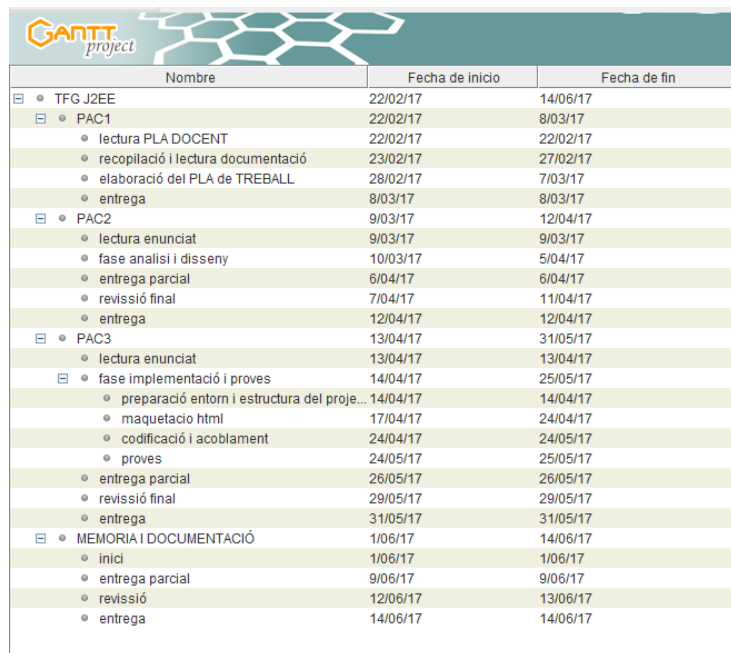
FASE 1	FASE 2	FASE 3	FASE 4
PAC1	PAC2	PAC3	Entrega final
<ul style="list-style-type: none"> <li>○ <b>Pla de treball</b> <ul style="list-style-type: none"> <li>• Objectiu del projecte</li> <li>• Planificació</li> <li>• Eines i frameworks a utilitzar</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>○ <b>Anàlisi i Diseny</b> <ul style="list-style-type: none"> <li>• Estudi de l'arquitectura i anàlisi del projecte.</li> <li>• Instal·lació i configuració de l'entorn de programació</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>○ <b>Desenvolupament</b> <ul style="list-style-type: none"> <li>• Maquetació.</li> <li>• Codificació</li> <li>• .</li> </ul> </li> <li>○ <b>Proves</b> <ul style="list-style-type: none"> <li>• Proves unitaris.</li> <li>• Proves d'integració.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>○ <b>Memòria</b> <ul style="list-style-type: none"> <li>• Documentació.</li> </ul> </li> </ul>

### Fites Temporals

Les dates clau han vingut imposades per les pròpies fases que ens venen imposades al TFG:

Data inici	Data final	Descripció
22/02/2017	08/03/2017	PAC1. Pla de treball
09/03/2017	12/04/2017	PAC2. Anàlisi i disseny
13/04/2017	31/05/2017	PAC3. Implementació i proves
01/06/2017	14/06/2017	Memòria i presentació

La tasques temporitzades a fer seria:



Nombre	Fecha de inicio	Fecha de fin
TFG J2EE	22/02/17	14/06/17
PAC1	22/02/17	8/03/17
lectura PLA DOCENT	22/02/17	22/02/17
recopilació i lectura documentació	23/02/17	27/02/17
elaboració del PLA de TREBALL	28/02/17	7/03/17
entrega	8/03/17	8/03/17
PAC2	9/03/17	12/04/17
lectura enunciat	9/03/17	9/03/17
fase analisi i disseny	10/03/17	5/04/17
entrega parcial	6/04/17	6/04/17
revisió final	7/04/17	11/04/17
entrega	12/04/17	12/04/17
PAC3	13/04/17	31/05/17
lectura enunciat	13/04/17	13/04/17
fase implementació i proves	14/04/17	25/05/17
preparació entorn i estructura del proje...	14/04/17	14/04/17
maquetacio html	17/04/17	24/04/17
codificació i acoblament	24/04/17	24/05/17
proves	24/05/17	25/05/17
entrega parcial	26/05/17	26/05/17
revisió final	29/05/17	29/05/17
entrega	31/05/17	31/05/17
MEMORIA I DOCUMENTACIÓ	1/06/17	14/06/17
inici	1/06/17	1/06/17
entrega parcial	9/06/17	9/06/17
revisió	12/06/17	13/06/17
entrega	14/06/17	14/06/17

## 1.5. Producte obtingut.

Els productes obtinguts són:

- Un fitxer executable **jar** (precintes-0.0.1-SNAPSHOT.jar) que conté:
  - Un Servidor d'aplicacions Tomcat totalment configurat i que serveix peticions web pel port 8080.
  - L'aplicació de gestió de precintes desenvolupada. Els objectes de base de dades i el joc de proves que es crea quan s'engega l'aplicació.
- Aquesta memòria, que conté l'especificació, l'anàlisi i el disseny de l'aplicació.
- Un Manual d'instal·lació del producte.

## 1.6. Descripció de la resta de capítols.

Als següents capítols presentarem una descripció del projecte i la seva especificació que és a on es realitzem la descomposició inicial del sistema en subsistemes a més a més de descriure el marc tecnològic utilitzat. A continuació presentarem l'anàlisi on es descriuen els actors, es posa a disposició un diagrama de casos d'ús per tenir una visió general del problema i es detalla cada cas d'ús mitjançant una fitxa organitzada per subsistemes.



El capítol del disseny està compost d'una sèrie de diagrames que ens facilita l'enteniment de l'aplicació des de diversos punts de vista dintre del model de referència pel model obert i distribuït RM-ODP així com un diagrama Model Entitat Relació. A continuació, al capítol d'implantació, fem referència a les eines de desenvolupament que s'han utilitzat, i els frameworks amb els que hem construït el producte. En el capítol de prototips mostrem les captures de pantalla de la interfície d'usuari i finalment exposarem les conclusions del projecte.

## 2. Descripció del projecte.

El projecte pretén facilitar la feina a la secció encarregada de gestionar els precintes dels vehicles del municipi. Disposaran d'una eina que els permetrà: consultar un vehicle per matrícula, veure quines característiques té, qui és el titular actual i els seus precintes. L'usuari ha de poder donar d'alta un vehicle i el seu titular, ha de poder introduir/modificar una ordre de precinte, introduir o modificar un desprecinte i anul·lar un desprecinte.

El projecte tindrà 2 parts diferenciades. Una part administrativa que serà gestionada només per un usuari administrador i que s'encarregarà de manejar les dades de les entitats que poden emetre l'ordre de precinte, els conceptes pel qual s'introdueix una ordre de precintes i els motius pels quals es desprecinte una ordre de precinte. L'altra part, serà gestionada per tots els usuaris que no són administradors i que podran gestionar la resta de funcionalitats que a continuació es descriuen.

Un vehicle pot tenir més d'una anotació de precinte. Del vehicle volem guardar la matrícula, la data de matriculació, el bastidor, la marca i el model, i el titular actual. Del titular volem guardar: el nom i llinatges, el document identificatiu, la data de naixement, el email, el telèfon i la seva adreça. De l'adreça volem registrar: el nom de carrer, el municipi, la població, el codi postal i la província.

De la introducció del precinte volem guardar la data d'introducció, l'entitat que emet l'ordre de precinte, el concepte pel qual es precinte, una referència externa i observacions. Els conceptes de precinte seran diferents per a cada entitat. Una anotació de precinte vigent es pot modificar per rectificar les dades introduïdes.

De la introducció del desprecinte volem guardar: la data de desprecinte, el motiu pel qual es desprecinte i observacions. Un cop s'ha desprecintat només es pot fer una modificació de les

dades de desprecinte o be anul·lar el desprecinte deixant el precinte en un estat vigent. També, s'ha de permetre fer recerques de precintes per data introducció de precinte, per data introducció de desprecinte i per entitat i/o concepte de precinte o desprecinte.

### **3. Especificació del projecte.**

#### **3.1. Especificació per sistemes.**

A partir de la descripció general del projecte podem considerar els següents Sistemes i l'agrupació de funcionalitats que ha d'oferir:

##### **3.1.1. Sistema d'identificació d'usuari.**

- Ha de permetre que un usuari s'identifiqui per entrar al sistema. Els usuaris tindran un rol assignat. Inicialment, hi haurà 2 rols que permetran a l'usuari fer tasques ben diferenciades. Un rol ADMINISTRADOR i un rol USUARI.

##### **3.1.2. Sistema de vehicles i titular.**

- Ha de gestionar l'alta o modificació del vehicle.
- Ha de gestionar l'alta o modificació de les dades del titular del vehicle.

##### **3.1.3. Sistema de gestió dels precintes.**

- S'ha de poder cercar un vehicle, mostrar les dades del vehicle, les dades del titular i els precintes que té el vehicle.
- S'ha de poder afegir o modificar una ordre de precinte. La modificació es podrà fer sempre que l'ordre no s'hagi desprecintat.
- S'ha de poder introduir un desprecinte d'una ordre de precinte indicant un motiu.
- S'ha de poder anul·lar una ordre de desprecinte. Pot ser que l'usuari s'hagi equivocat i ha de deixar l'ordre de precinte en el mateix estat anterior a l'operació de desprecintar.
- Donada una matrícula hem de saber quans precintes té vigents. Aquesta informació ajudarà a l'usuari a saber el nombre de precintes vigents que té el vehicle actualment.

- Sempre queda registrat l'usuari que ha fet la darrera operació sobre el precinte.

#### **3.1.4. Subsistema manteniments.**

- L'usuari administrador ha de poder donar d'alta, modificar o esborrar les entitats que ordenen els precintes, els conceptes de precinte i els conceptes de desprecinte.

### **3.2. Seguretat**

El maneig de la seguretat serà senzill. Qualsevol usuari amb rol USUARI tindrà accés a tots els subsistemes proposats excepte el subsistema de manteniment que serà gestionat per usuaris amb rol ADMINISTRADOR.

## **4. Anàlisi funcional.**

En aquest apartat, descriurem els actors que intervenen en el sistema i el paper que juga cada un en els diferents subsistemes. Presentarem un diagrama de casos d'ús i descriurem cada cas d'ús en una fitxa. A més a més, inclourem altres diagrames que són rellevants pel sistema que estem construint: un diagrama de classes JPA, el model conceptual de base de dades, el diagrama de classes DTO i un diagrama de components per tenir una visió acurada de l'arquitectura de 3 capes.

### **4.1. Actors.**

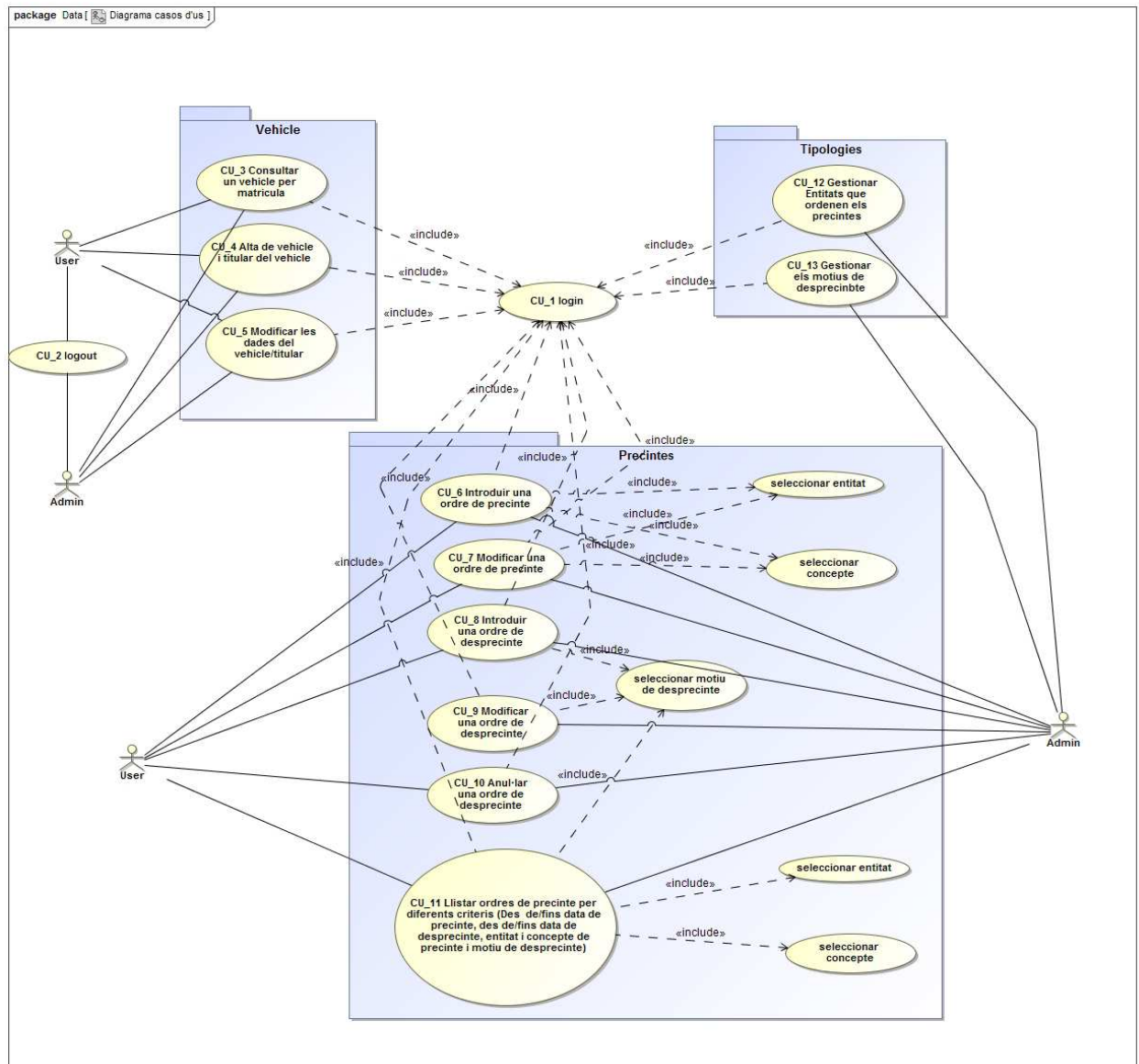
Definim un actor principal com l'usuari que vol fer servir el sistema per a satisfer un objectiu concret. Els actors principals que intervenen en el sistema són dos:

- Usuari de tipus Administrador que és l'encarregat de mantenir les tipologies.
- Usuari que s'encarrega de fer la gestió dels precintes. Consulta d'una matrícula, introduccions, modificacions, etc...

Ambdós són usuaris que s'han hagut d'autenticar per entrar al sistema i per tant el sistema només pot ser utilitzat per usuaris que disposi d'un perfil autoritzat.

Els permisos que tindran aquest tipus d'actors són. L'administrador podrà dur a terme tota la funcionalitat que té el sistema mentre que l'usuari amb un perfil d'introducció podrà fer tot el mateix que l'administrador excepte el manteniment de les tipologies. Els manteniments de les tipologies està compost per: les entitats que ordenen els precintes, conceptes de precinte i motius de desprecinte.

#### 4.2. Diagrama de casos d'ús.



#### 4.3. Fitxes de casos d'us.

Recordem que un cas d'us és una mena de contracte entre desenvolupament i l'usuari final o interessat en el sistema (stakeholder) . Un cas d'us descriu les interaccions que far l'usuari amb

el sistema. S'ha intentat agrupar tots els casos d'ús per subsistema per tenir una visió més esquematitzada.

#### *4.3.1. Subsistema de identificació d'usuaris.*

**Cas d'ús:** CU\_1. Login

**Resum de la funcionalitat:**

Permetre a un usuari introduir els seus credencials per entrar a l'aplicació.

**Actors:** Usuari, Administrador

**Casos d'us relacionats :**

**Precondició:**

**Postcondició:**

El sistema mostra la pàgina principal de l'aplicació a l'usuari.

**Escenari principal:**

1. L'usuari introdueix una url per accedir a l'aplicació
2. El sistema detecta que es vol accedir a algun recurs de l'aplicació, que l'usuari no s'ha autenticat i mostra el formulari de login.
3. L'usuari introdueix els seus credencials
4. El sistema valida els credencials i mostra a l'usuari el formulari inicial de l'aplicació

Extensions i restriccions

- 4a. El sistema no valida els credencials i mostra un error.

**Cas d'ús:** CU\_2. Logout

**Resum de la funcionalitat:**

Permetre a un usuari sortir de l'aplicació.

**Actors:** Usuari, Administrador

**Casos d'us relacionats :**

**Precondició:**

**Postcondició:**

El sistema mostra el formulari de login a l'usuari.

**Escenari principal:**

1. L'usuari prem sobre el botó de Sortir
2. El sistema allibera els recursos que identifiquen a l'usuari com autenticat i el sistema mostra la pantalla de login amb un missatge que indica a l'usuari que s'ha desconnectat del sistema

Extensions i restriccions

#### *4.3.2. Subsistema Vehicles.*

**Cas d'ús:** CU\_3. Consultar un vehicle per matricula

**Resum de la funcionalitat:**

Introduint una matricula i pitjant sobre el botó cercar, el sistema ha de recuperar i l'usuari ha de veure les dades del vehicle, les dades del seu titular i tots els precintes que s'han registrat

sobre el vehicle.

**Actors:** Usuari, Administrador

**Casos d'us relacionats**

**Precondició:**

L'usuari s'ha autenticat en el sistema.

**Postcondició:**

L'usuari veu el formulari amb tota la informació relacionada amb el vehicle.

**Escenari principal:**

1. El sistema mostra a l'usuari un formulari per introduir una matricula, un botó per llançar la consulta i un botó per un alta de vehicle.
2. L'usuari introdueix la matricula i prem sobre el botó cercar.
3. El sistema accedeix a la base de dades i mostra un formulari on es veu les dades del vehicle i el titular, els nombre de precintes actius i un botó per modificar les dades del vehicle/titular. També es mostra un detall dels precintes relacionats amb el vehicle i una barra de botons per dur a terme les operacions de: alta de precinte, editar precinte, alta desprecinte i anul·lar desprecinte.

**Extensions i restriccions.**

**Cas d'ús:** CU\_4. Alta vehicle/titular

**Resum de la funcionalitat:**

Donar d'alta un vehicle i el seu titular.

**Actors:** Usuari, Administrador

**Casos d'us relacionats**

**Precondició:**

L'usuari s'ha autenticat en el sistema.

**Postcondició:**

El sistema ha guardat els canvis a la base de dades.

**Escenari principal:**

1. L'usuari prem sobre el botó alta vehicle.
2. El sistema mostra un formulari per introduir les dades del vehicle i les dades del titular.
3. L'usuari introdueix totes les dades a complimentar i prem sobre el botó Guardar
4. El sistema valida les dades introduïdes, assigna una numeració automàticament al vehicle i al titular, registre l'alta del vehicle i del titular a la base de dades i torna a la pantalla anterior.

**Extensions i restriccions**

3a1. L'usuari prem sobre el botó Cancel·la

3a2. El sistema torna a la pantalla anterior.

4a1. El sistema no valida les dades introduïdes i mostra a l'usuari els errors que s'han produït.

**Cas d'ús:** CU\_5. Modificar dades vehicle/titular

**Resum de la funcionalitat:**

Modificar les dades d'un vehicle que prèviament s'ha consultat i/o les dades del seu titular.

**Actors:** Usuari, Administrador

**Casos d'us relacionats:** Consulta de les dades del vehicle per matricula

**Precondició:**

L'usuari s'ha autenticat en el sistema.

**Postcondició:**

El sistema ha guardat els canvis a la base de dades.

**Escenari principal:**

1. L'usuari ha consultat una matrícula i el sistema li mostra el formulari amb les dades del vehicle/titular i la relació de precintes.
2. L'usuari prem sobre el botó modificar.
3. El sistema mostra un formulari per modificar les dades del vehicle i les dades del titular.
4. L'usuari modifica les dades que el sistema li permet modificar i prem sobre el botó Guardar
5. El sistema valida les dades introduïdes, guarda les dades a base de dades i torna a la pantalla anterior.

**Extensions i restriccions**

4a1. L'usuari prem sobre el botó Cancel·la

4a2. El sistema torna a la pantalla anterior.

5a1. El sistema no valida les dades introduïdes i mostra a l'usuari els errors que s'han produït.

*4.3.3. Subsistema Precintes.*

**Cas d'ús:** CU\_6. Alta precinte, CU\_7. Modificació ordre de precinte

**Resum de la funcionalitat:**

Introduir o modificar una ordre de precinte. Les dades a registrar o modificar són: data de precinte, entitat que ordena el precinte, el concepte de precinte, una referència i observacions.

**Actors:** Usuari, Administrador

**Casos d'us relacionats** seleccionar una entitat, seleccionar un concepte

**Precondició:**

L'usuari s'ha autenticat en el sistema.

**Postcondició:**

El sistema guarda els canvis a la base de dades.

**Escenari principal:**

1. L'usuari prem sobre el botó alta precinte o editar precinte.
2. El sistema mostra un formulari per que l'usuari pugui introduir les dades especificades.
3. L'usuari introdueix totes les dades i prem sobre el botó Guardar
4. El sistema valida les dades introduïdes, assigna una numeració automàticament en cas d'alta, guarda els canvis a la base de dades i torna a la pantalla anterior.
5. El sistema mostra un avis que l'alta s'ha dut a terme de forma satisfactòria.

**Extensions i restriccions**

3a1. L'usuari prem sobre el botó Cancel·la

3a2. El sistema torna a la pantalla anterior.

4a1. El sistema no valida les dades introduïdes i mostra a l'usuari els errors que s'han produït.

**Cas d'ús:** CU\_8. Introduir una ordre de desprecinte, CU\_9 Modificació una ordre de desprecinte

**Resum de la funcionalitat:**

Sobre una ordre de precinte vigent, l'usuari ha de poder introduir o modificar una ordre de desprecinte. L'usuari informarà la data de desprecinte, seleccionar el motiu del desprecinte i afegir observacions.

**Actors:** Usuari, Administrador

**Casos d'us relacionats** seleccionar un motiu de desprecinte

**Precondició:**

L'usuari s'ha autenticat en el sistema.

**Postcondició:**

El sistema guarda els canvis a la base de dades.

**Escenari principal:**

1. L'usuari ha consultat un vehicle i el sistema mostra una relació de precintes.
2. L'usuari prem sobre el botó alta desprecinte o selecciona un precinte amb data de desprecinte i pitja editar desprecinte.
3. El sistema mostra un formulari per introduir les dades especificades.
4. L'usuari introdueix totes les dades i prem sobre el botó Guardar
5. El sistema valida les dades introduïdes, guarda els canvis a la base de dades i torna a la pantalla anterior.

**Extensions i restriccions**

3a1. L'usuari prem sobre el botó Cancel·la

3a2. El sistema torna a la pantalla anterior.

4a1. El sistema no valida les dades introduïdes i mostra a l'usuari els errors que s'han produït.

**Cas d'ús:** CU\_10. Anul·lar una ordre de desprecinte

**Resum de la funcionalitat:**

Sobre una ordre de precinte que ja s'ha desprecintat (amb data de desprecinte), l'usuari ha de poder anul·lar el desprecinte i desar l'ordre de precinte en un estat vigent.

**Actors:** Usuari, Administrador

**Casos d'us relacionats**

**Precondició:**

L'usuari s'ha autenticat en el sistema.

**Postcondició:**

El sistema guarda els canvis a la base de dades.

**Escenari principal:**

1. L'usuari ha consultat un vehicle i el sistema mostra una relació de precintes.
2. L'usuari selecciona un precinte que té data de desprecinte i prem sobre el botó anul·lar desprecinte.
3. El sistema mostra una pantalla de confirmació i l'usuari confirma l'operació.
4. El sistema guarda els canvis a la base de dades i torna a la pantalla anterior.

**Extensions i restriccions**

3a1. L'usuari no confirma l'operació.

3a2. El sistema tanca la finestra de confirmació.

**Cas d'ús:** CU\_11. Llistar ordres de precinte per diferents criteris.

**Resum de la funcionalitat:**



L'usuari ha de poder cercar i visualitzar una relació de precintes per diferents criteris de recerca.

**Actors:** Usuari, Administrador

**Casos d'us relacionats**

**Precondició:**

L'usuari s'ha autenticat en el sistema.

**Postcondició:**

L'usuari veu una relació de precintes amb els filtres especificats

**Escenari principal:**

1. L'usuari prem sobre el botó Altres recerques
2. El sistema mostra un formulari on l'usuari pot indicar: una data des de/fins de precinte, una data des/fins de desprecinte, una entitat de precinte, un concepte de precinte i un motiu de desprecinte i un botó per cercar.
3. L'usuari introdueix els criteris i prem sobre el botó cercar.
4. El sistema consulta a base de dades i mostra una relació de precintes que compleixen els criteris introduïts.

**Extensions i restriccions**

4a1. El sistema no trova cap dada amb els criteris introduïts i mostra a l'usuari un missatge que diu "no hi ha dades a mostrar"

#### *4.3.4. Subsistema Tipologies.*

**Cas d'ús:** CU\_12 i CU\_13. Gestionar entitats i conceptes de precinte, i motius de desprecinte

**Resum de la funcionalitat:**

Es tracta de funcionalitats que permeten per les diferents tipologies que hi ha a l'aplicació gestionar les funcionalitats de llistar tots els registres, donar d'alta un nou registre, modificar i fer la baixa.

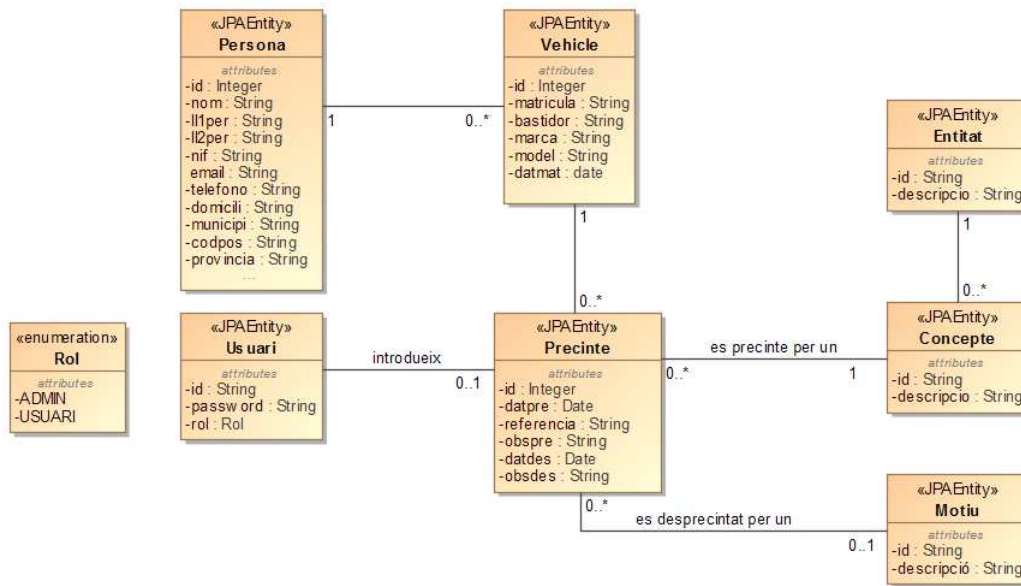
**Actors:** Administrador

## 5. Disseny.

### Diagrama de classes JPA.

El diagrama de classes JPA és una de les peces més importants dintre del projecte ja que representa el model de domini del negoci que es construirà i està molt vinculat al model entitat relació. Aquestes classes representades en el diagrama contindran anotacions que representen: el nom de la taula, la clau primària simple o composta, la seqüència que alimentarà la clau primària, les relacions que estan representades en el diagrama. Totes aquestes anotacions són utilitzades per hibernate per generar a base de dades tots els objectes involucrats dintre del diagrama.

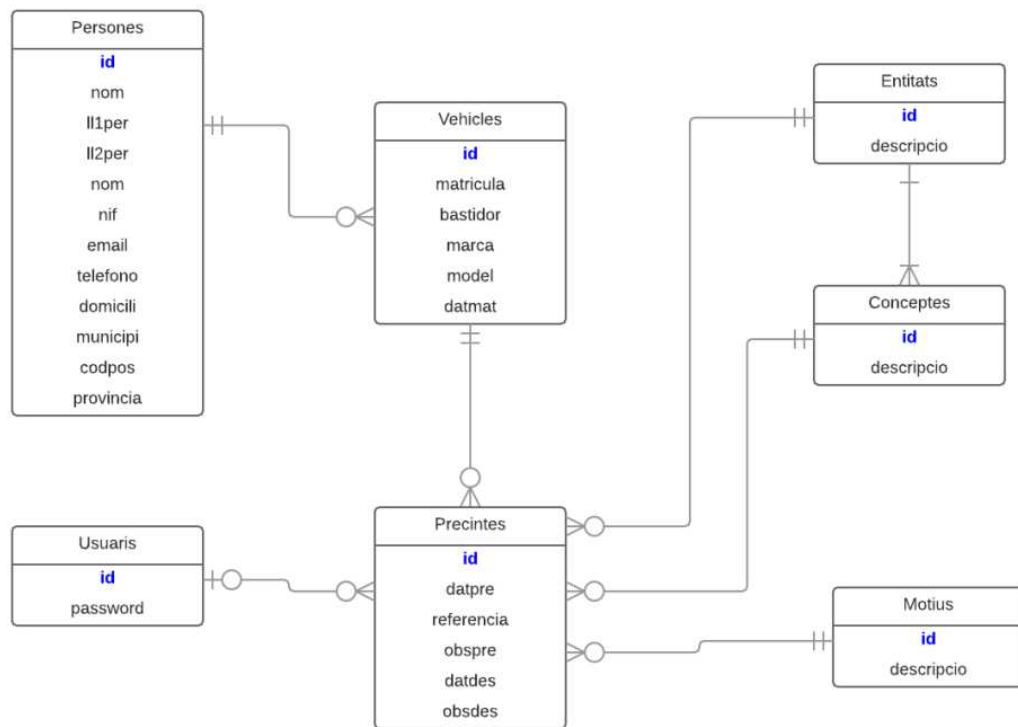
A continuació es mostra el diagrama de classes JPA.



### Diagrama del model Base de dades.

A continuació es presenta el diagrama de model de dades que correspon al model de classes JPA presentada a l’apartat anterior. En el diagrama anterior ja que es veu els tipus que tindran els atribut de les taules que generi hibernate i les cardinalitats entre les diferents entitats. Així pues un atribut de tipus String a base de dades Postgree Sql les resoldrà com un tipus character varying, un Long com un bigint i un Date com un timestamp without timezone.

El diagrama del model entitat relació és:



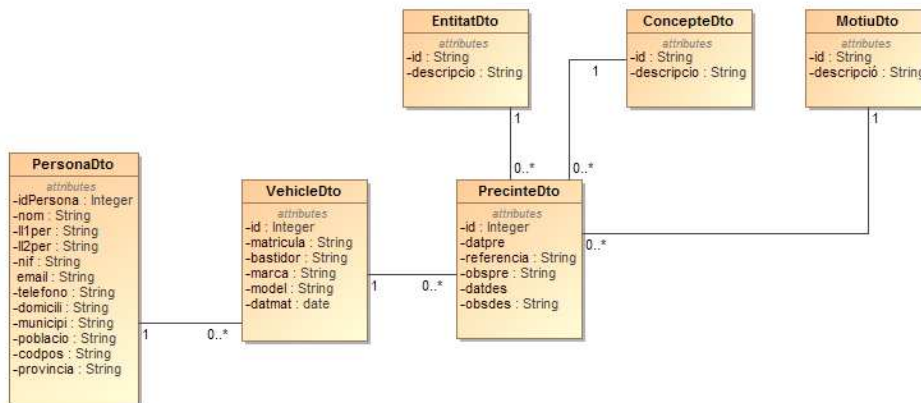
## Diagrama de classes DTO.

Un DTO és un classe Java que es pot serialitzar i la podem tractar com un objecte que es transfereix entre el client i el servidor. Aquest objecte té un constructor que permet, a partir d'una entitat JPA que li arriba per paràmetre construir una instància. El seu ús més habitual és facilitar el transport dels objectes entre processos. En el nostre cas, es preveu que en un futur aquesta aplicació quedarà dividida en un projecte front i un altre api i tenir en compte en aquest moment aquest patró de disseny ens beneficiarà en un futur.

Dintre del nostre projecte, la classe DTO és utilitzada principalment des de la capa vista fins a la capa de negoci (service). En aquesta capa és on servim les dades sol·licitades en forma de classes DTO després de fer la crida al DAO que és el component encarregat de dur a terme la persistència mitjançant la implementació de JPA Hibernate.

Al Igual que les classes JPA utilitzarem l'especificació JSR 349 (Bean Validation 1.1) per fer les validacions més habituals de les dades.

A continuació es mostra el diagrama:

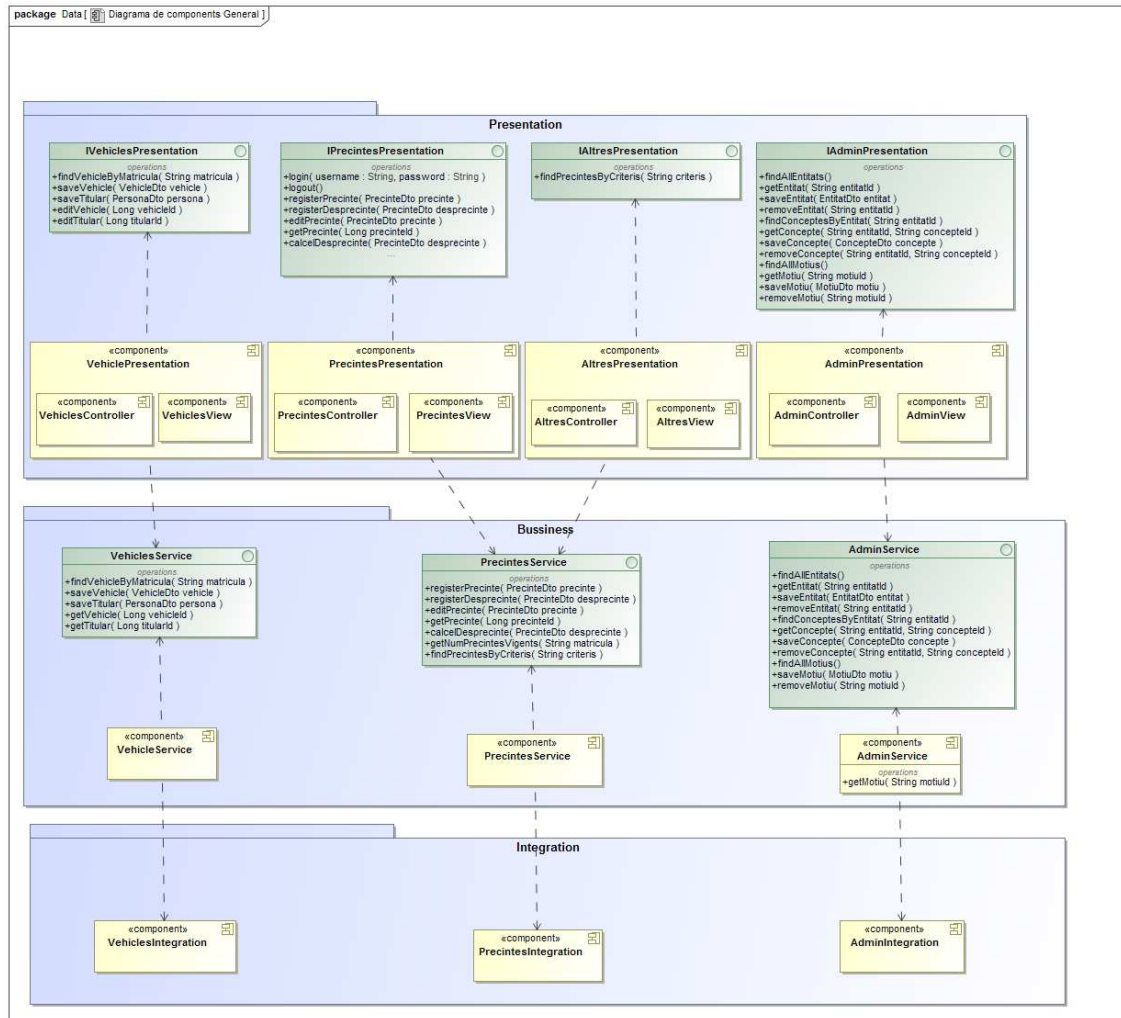


## Diagrama d'arquitectura.

L'arquitectura seleccionada per fer el projecte és una barreja d'arquitectures entre les que destaquem: arquitectura client servidor on hi ha uns components servidors que ofereixen serveis i un conjunt clients que consumeixen aquest serveis. Una arquitectura heterogènia organitzat per capes en que el client només és l'encarregat de executar el programari de presentació i el servidor és l'encarregat de manejar la lògica de negoci i la persistència amb la base de dades.

A la següent imatge presentem un diagrama de components general que està compost de:

- La capa de presentació esta formada per dos components principals. El component que gestiona les accions que inicia l'usuari o el sistema i un component que gestiona les vistes.
- La capa de servei que té un sol component que és l'encarregat de gestionar l'accés a les dades.
- La capa de integració amb un sol component que és l'encarregat d'accedir i registrar les dades a la base de dades.

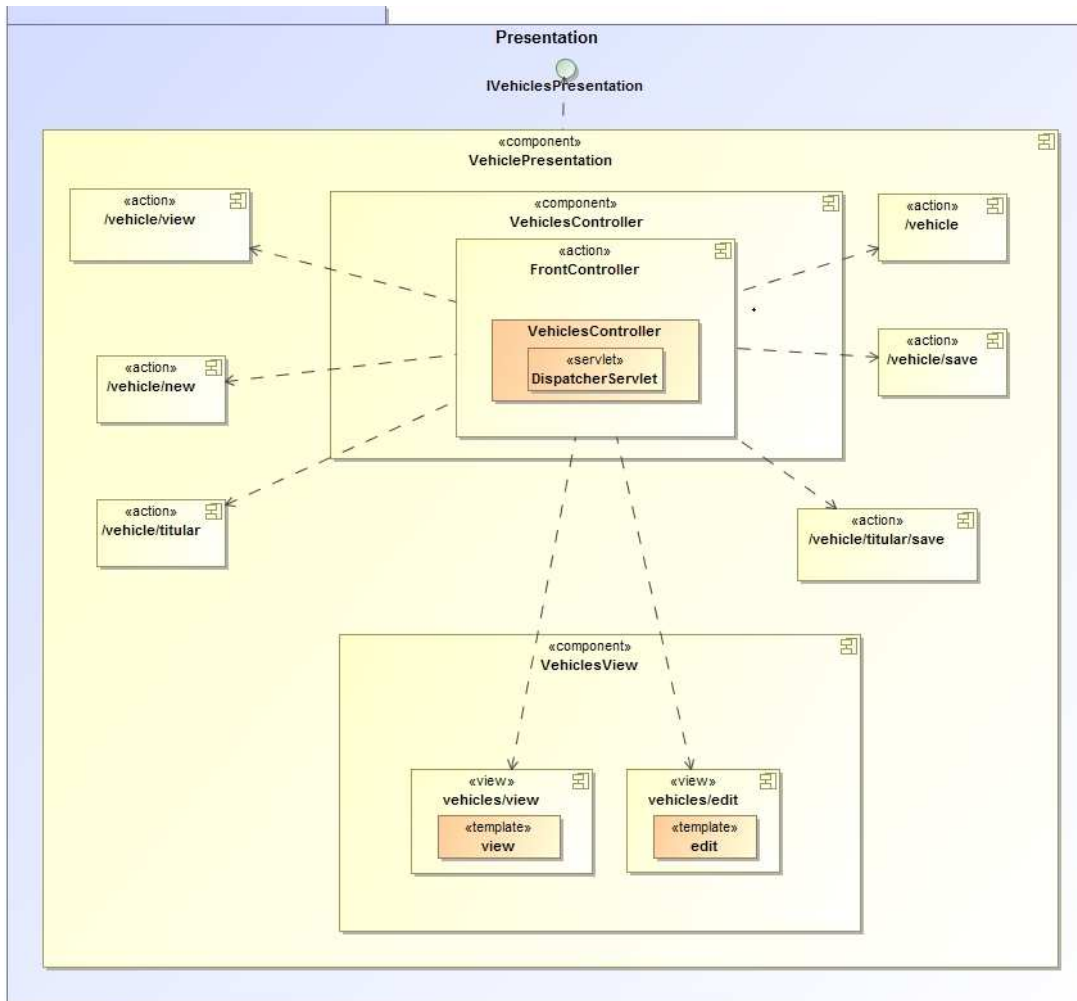


Els diagrames refinats de la capa de presentació per cada subsistema i aplicada la tecnologia a utilitzar serien:

- Subsistema de Vehicles.

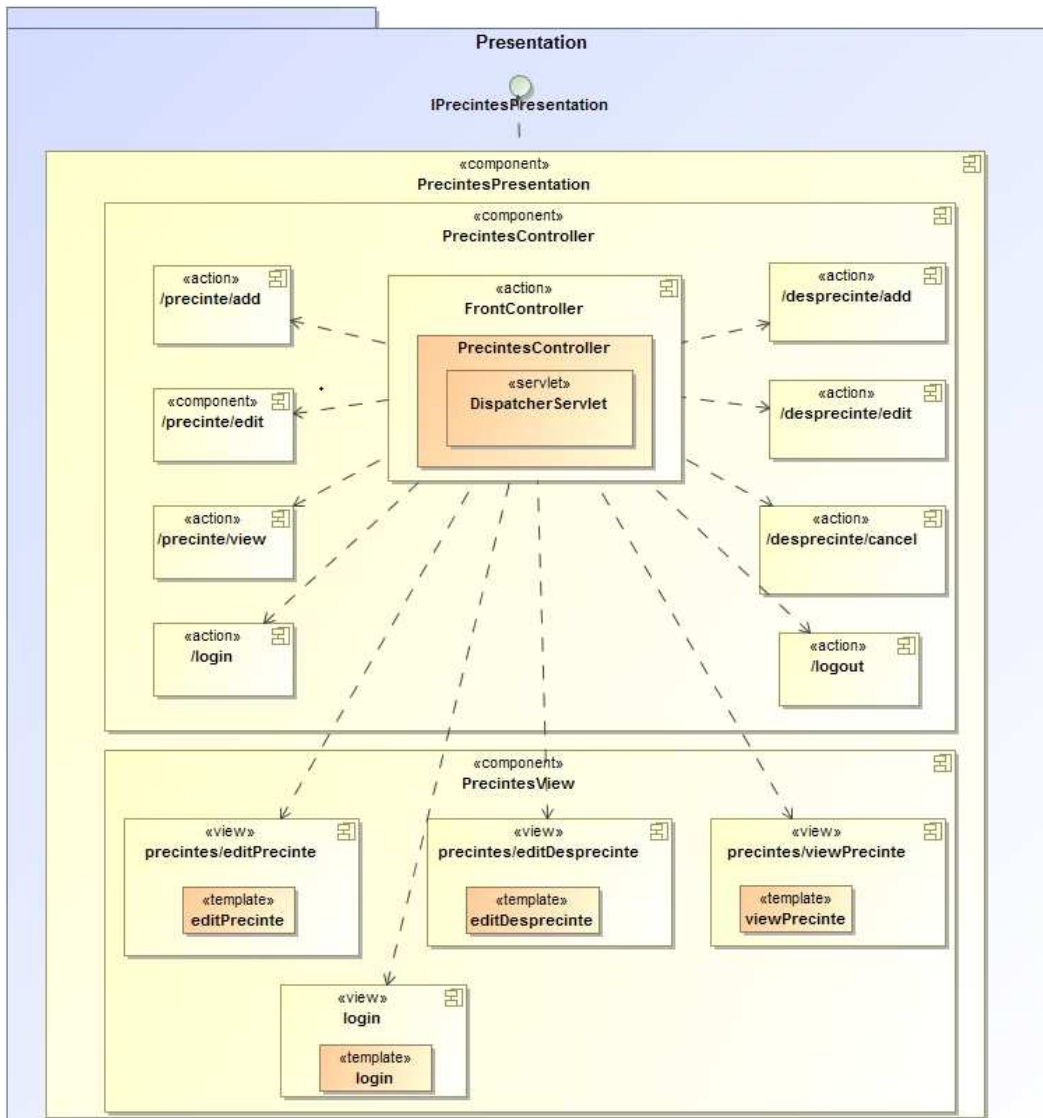
En aquest cas, apareixen nous components:

- **action.** És el component que indica l'acció que vol dur a terme l'usuari.
- **view.** És la plantilla html que s'ha de mostrar a l'usuari relacionada amb l'acció que ha iniciat.
- **VehiclesController.** És una classe amb una anotació de tipus Controller de Spring que és l'encarregada de gestionar les accions(action) que ha iniciat l'usuari

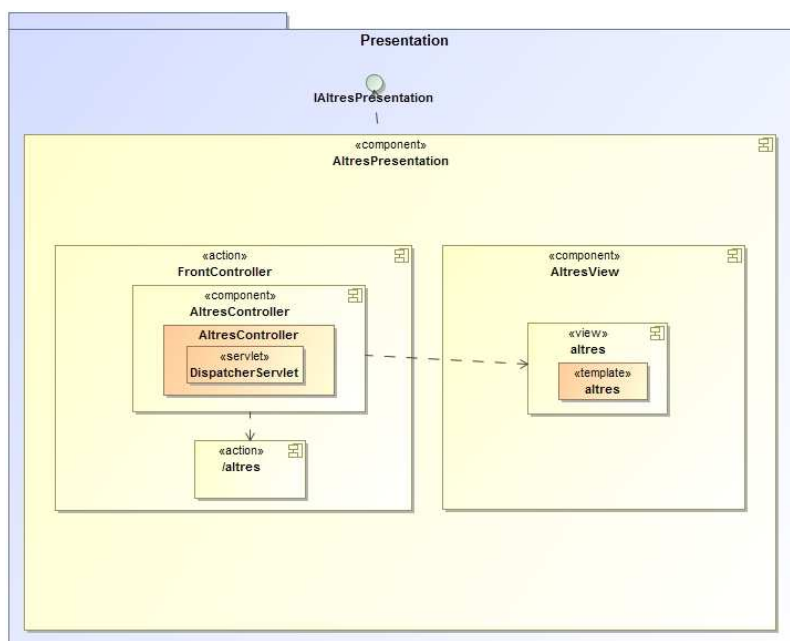


- Subsistema de Precintes.

Els components que apareixen en la imatge següent són anàlegs als comentats a l'apartat anterior. Per simplicitat del cas d'ús del login s'ha desat que el controlador PrecintesController es faci càrrec de la seva gestió.

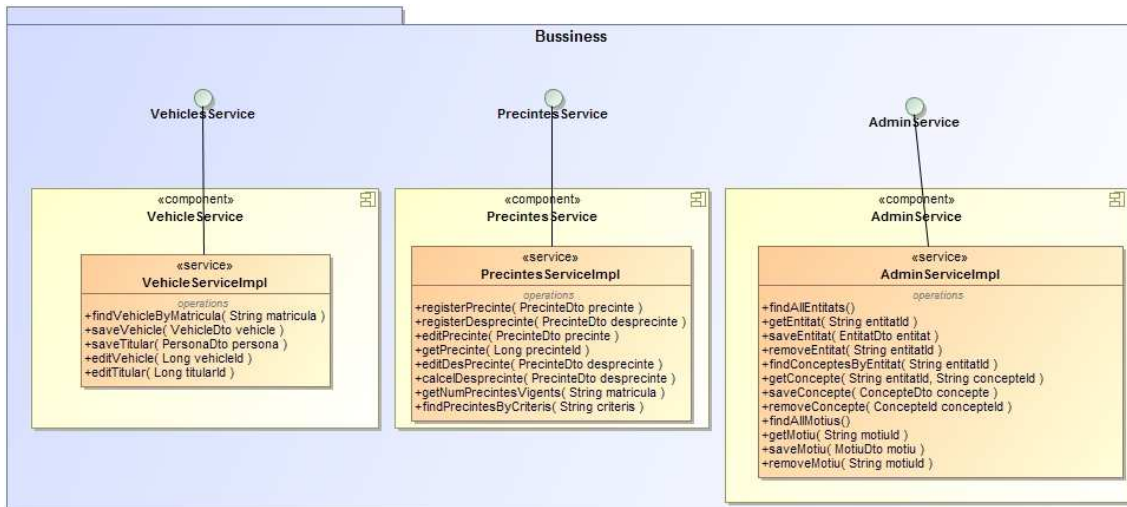


- Subsistema Altres Recerques.

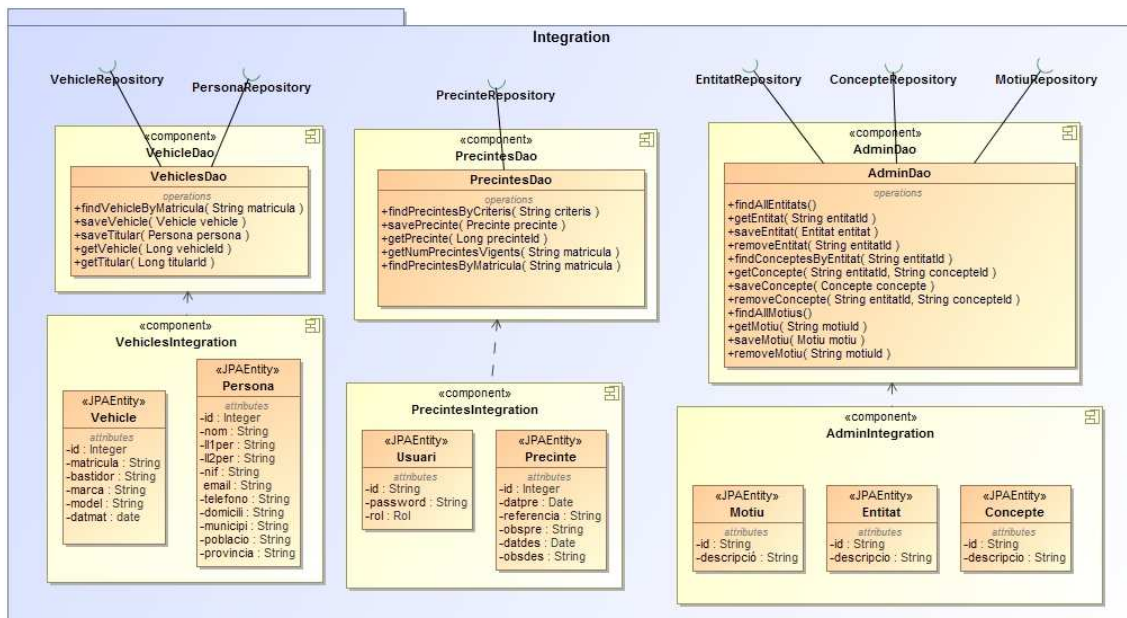




El diagrama refinat de la capa de servei i aplicada la tecnologia a utilitzar és el següent



El diagrama refinat de la capa d'integració i aplicada la tecnologia a utilitzar és el següent



## 6. Implementació.

### 6.1.Eines de desenvolupament.

Com a eina de desenvolupament s'ha utilitzat **Eclipse Neon 3** for Java EE developer i el **JDK 8** de 64 bits. S'ha afegit els pluggins: STS Spring Framework per facilitar l'integració de Spring amb l'IDE i el pluggin EGIT per facilitar el control de versions del codi mitjançant un repositori git i github.



Com a gestor del projecte s'utilitzarà **Maven** ja que ens ajuda a manejar les dependències de les llibreries que són necessàries per dur a terme el projecte.

## 6.2. Frameworks utilitzats.

El framework **java** principal que s'ha emprat és: **Spring Boot**. Es tracta d'un framework molt acceptat per la comunitat que ens permet crear diferents tipus d'aplicacions de forma senzilla i ràpida. La versió que s'ha utilitzat és la 1.4.6.

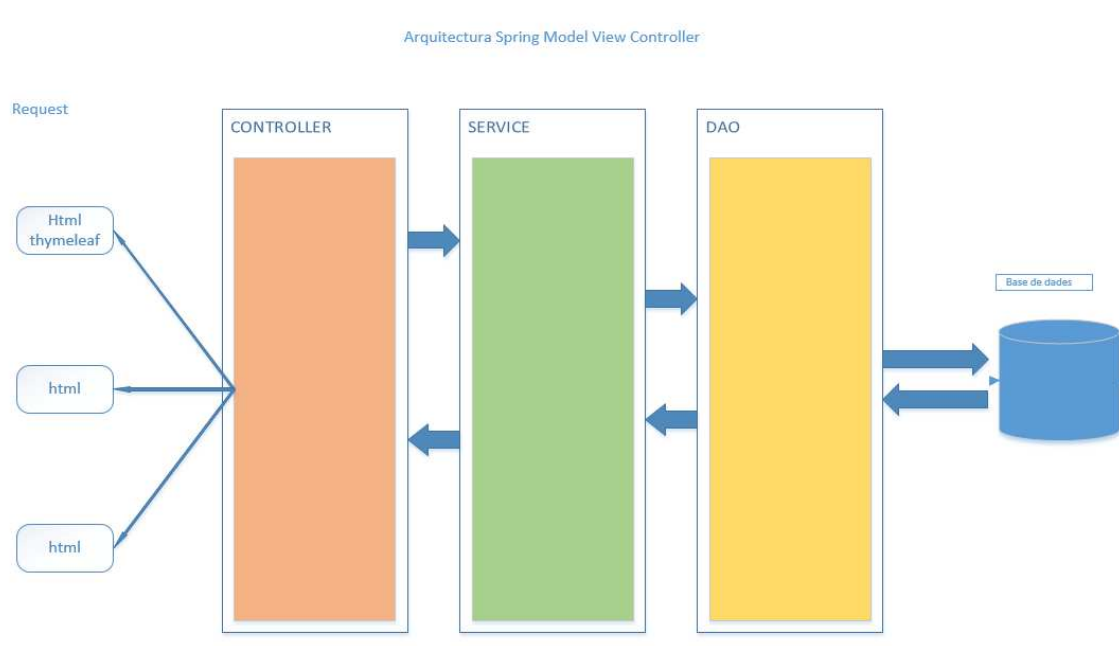
Les seves característiques principals són:

- Permet crear aplicacions stand-alone.
- Porta un servidor Tomcat encastrat.
- Configura Spring automàticament, sempre que pot.
- Facilita l'utilització d'anotacions per realitzar funcions de configuració i injecció de dependències.
- No requereix configuració xml.

Utilitzar aquest framework pot facilitar l'entrega del producte ja que és un jar que es pot executar fàcilment a una màquina que tingui una màquina virtual de Java instal·lada.

A continuació mostrem un esquema molt general de les peces que intervien en la arquitectura Spring MVC.

Els usuaris, des de la capa Vista fan una petició de un recurs al servidor i aquestes peticions són gestionades per classes que resideixen a la capa Controller. A continuació, el controlador far peticions dels recursos que necessita a la capa de servei i si necessita algun recurs de la base de dades ho demana a la capa Dao que és la que far l'accés a la base de dades. La capa Dao retorna els recursos a la capa servei i aquesta a la capa controller i finalment a la capa vista on l'usuari pot veure els resultats de la petició inicial.



A la capa de presentació, s'ha construït amb l'ajuda del framework **Thymeleaf**. És tracta d'una llibreria Java que implementa un motor de plantilles HTML5/XML/XHTML que es pot utilitzar tant en entorns web com entorn no web. Hi ha un bon acoblament entre la capa vista i Spring MVC.

D'altra banda, com a llenguatges de la part de client hem hagut de aprendre: HTML, Javascript i el framework d'estils CSS **Bootstrap** v3.3.7

Per la capa de Serveis i DAO utilitzarem la injecció de dependències que ens proporciona Spring i el component de Java EE, Bean validation JSR-303. La capa de persistència utilitzarem el mòdul **spring-data-jpa** que té com a dependències les següents tecnologies claus pel nostre projecte. Hibernate 5.0 com a implementació de JPA. **Spring Data JPA** una eina basada en repositoris JPA i que facilita l'accés i registre de les dades.

També utilitzarem i de forma puntual el framework **Querydsl**. Querydsl proporciona una marc que permet construir consultes mitjançant una fluent-API de forma que en temps de compilació en queixa si no es fa referència a objectes del domini coneguts. El gestor de base de dades que s'ha utilitzat és **PostgreSQL**.

Per acabar, com a possible millora en un futur, podríem pensar amb una arquitectura de Micro serveis. Llavors, aquest projecte Maven tindria 3 subprojectes. Un projecte front que gestionaria les peticions de l'usuari i manejaria les pantalles, un projecte API que gestionaria

les peticions de dades que far el projecte front mitjançant un servei REST i un projecte common que contindria el objectes DTO comuns als dos subprojectes anteriorment citats.

Les avantatges d'aquest tipus de projecte seria la reutilització dels serveis per diferents tipus de projecte front pugui ser un aplicació web tradicional o un projecte pel aplicacions de dispositius mòbils.

### **6.3.Proves unitàries.**

Per fer proves amb les consultes de les altres recerques s'ha utilitzat Junit. Junit és framework que permet realitzar l'execució de classes i mètodes Java que permeten comprovar el seu bon funcionament.

## **7. Interfícies d'usuari.**

La interfície d'usuari s'ha construït mitjançant plantilles html de thymeleaf. A continuació mostrem algunes d'elles.

- Formulari d'accés

Quan un usuari intenti accedir, via url a un recurs de l'aplicació de precintes el sistema mostrarà la següent pantalla.

- Formulari inicial.

Des de aquest formulari, l'usuari pot introduir la matricula per accedir a les dades del vehicle i els precintes anotats. També pot accedir al formulari d'alta del vehicle. Cal dir que tots els formularis conserven la mateixa capçalera de navegació i el mateix peu. Des de aquesta capçalera l'usuari sempre pot anar a la pàgina inicial (Home), a Altres recerques, a la zona exclusiva de treball de l'administrador i a la dreta es pot visualitzar l'usuari que s'ha identificat i el control per sortir del sistema (Logout).

- Formulari alta/modificació del vehicle

En aquest formulari, l'usuari podrà introduir les dades del vehicle, les dades del titular i el seu domicili.

Gestió de precintes Home Altres Recerques Administrador Sortir

### Formulari de vehicles

ID	<input type="text"/>	Id persona	<input type="text"/>
Matricula	ex. NNNNXXX	Nom	nom
Bastidor	ex. WWZZZ3CZ7P000000	Llinatge 1	primer llinatge
Marca	ex. SEAT	Llinatge 2	segon llinatge
Model	ex. IBIZA	Nif	document identificatiu
Data matriculació	dd/mm/yyyy	Teléfono	telefono
		email	name@domain.com
		Domicili	ex. ce Picornell núm: 26 pis 2 porta A
		Municipi	ex. PALMA
		Codi postal	cp
		Provincia	ex. ILLES BALEARS

TFG UOC Java EE, © 2017

- Formulari fitxa del vehicle

En aquest formulari l'usuari veu el resultat de la consulta del vehicle per matricula. Es veu un apartat amb les dades tècniques del vehicle, les dades del titular i el total dels precintes vigents de forma que sigui perceptible per l'usuari. L'usuari pot editar les dades del vehicle o titular o pot accedir a les operacions possibles de: alta/modificació de precinte, alta/modificació de desprecinte o anul·lació de desprecinte.

localhost:8080/vehicle/view?matric=1111BBC&\_csrf=8fd8fedd-c550-491f-b4a0-94491e262510

Gestió de precintes Home Altres Recerques Administrador admin Sortir

## Fitxa del vehicle

Identificador:	1	Data Matricula:	01/02/2000	Titular:	GINARD PRATS, BERNAT
Matricula:	1111BBC	Marca i model:	SEAT LEON	Domicili	CE JOVELLANOS num:14 pis 2 pta A
Bastidor:	W122112345RT54Y6W				PALMA 07003 ILLES BALEARS

**Relació de precintes** Total precintes actius: 2

id#	Data introducció	Entitat	Concepte	Data desprecinte	Motiu desprecinte	Acció
1	01/02/2000	JUDI	Judjat nro 1 primera instancia			<input type="button" value="Acció"/>
2	01/02/2000	RECAP	Recaptador de impostos en voluntaria			<input type="button" value="Acció"/>

TFG UOC Java FF © 2017

- Formulari introducció de dades de precinte

Des de aquest formulari l'usuari veu la fitxa del vehicle i pot introduir les dades del precinte o be cancel·lar l'operació.

## Alta de precinte

Identificador: 1 Data Matricula: 01/02/2000 Titular: GINARD PRATS, BERNAT  
 Matricula: 1111BBC Marca i model: SEAT LEON Domicili: CE JOVELLANOS num:14 pis 2 pta A  
 Bastidor: W122112345RT54Y6W PALMA  
 07003 ILLES BALEARS

**Id**

**Data de precinte**

**Entitat**

**Concepte**

**Referència**

**Observacions**

TFG UOC Java EE, © 2017

- Formulari introducció de dades de desprecinte

Des de aquest formulari l'usuari pot visualitzar la fitxa del vehicle, les dades del precinte i introduir les dades del desprecinte o be cancel·lar l'operació.

localhost:8080/desprecinte/1

## Alta de desprecinte

Identificador: 1 Data Matricula: 01/02/2000 Titular: GINARD PRATS, BERNAT  
 Matricula: 1111BBC Marca i model: SEAT LEON Domicili: CE JOVELLANOS num:14 pis 2 pta A  
 Bastidor: W122112345RT54Y6W PALMA  
 07003 ILLES BALEARS

**Id**

**Data de precinte**

**Entitat**

**Concepte**

**Referència**

**Observacions**

**Data desprecinte**

**Motius de desprecinte**

**Usuari**

**Observacions**

TFG UOC Java EE, © 2017

- Formulari consulta de dades de precinte

Des de aquest formulari l'usuari pot visualitzar: la fitxa del vehicle, les dades del precinte i les dades del desprecinte.

Gestió de precintes   Home   Altres Recerques   Administrador ▾
admin   Sortir

## Detall del precinte

Identificador:	1	Data Matricula:	01/02/2000	Titular:	GINARD PRATS, BERNAT
Matricula:	1111BBC	Marca i model:	SEAT LEON	Domicili:	CE JOVELLANOS num:14 pis 2 pta A
Bastidor:	W122112345RT54Y6W				PALMA 07003 ILLES BALEARS

**Id**

**Data de precinte**

**Entitat**

**Concepte**

**Referència**

**Observacions**

**Data desprecinte**

**Motius de desprecinte**

**Usuari**

**Observacions**

PROVES

[Tornar](#)

TFG UOC Java EE, © 2017

- Formulari altres recerques

Des de aquest formulari l'usuari pot filtrar precintes per diversos criteris per així obtenir una llista de precintes.

## Altres recerques

**Des de data precinte** 
**Des de data desprecinte**

**Fins data precinte** 
**Fins data desprecinte**

**Entitat** 
**Motiu de desprec.**

**Concepte**

Codi	Data precinte	Matricula	Títular	Entitat	Concepte	Referència	Data desprecinte	Motiu
2	01/02/2000	1111BBC	GINARD PRATS, BERNAT	RECAP	Recaptador de impostos en voluntaria	REF. 01/12/2015		
1	01/02/2000	1111BBC	GINARD PRATS, BERNAT	JUDI	Judjat nro 1 primera instancia	REF. 123456789	01/02/2017	DEPA
3	01/05/2016	2222JCS	GINARD LLABRES, LAIA	JUDI	Judjat nro 2 primera instancia	REF. 123456789		
4	01/02/2017	3333JKS	LLABRES SERRA, ISABEL	JUDI	Judjat nro 3 primera instancia	REF. 123456789		
5	06/06/2017	1111BBC	GINARD PRATS, BERNAT	ORGPO	Policia Local	REF. EXP MULTA:12345		

TFG UOC Java EE, © 2017

- Formulari llista d'entitats de precinte

Des de aquest formulari l'usuari visualitza totes les entitat de precinte, té la possibilitat d'afegir una nova entitat o editar una entitat per modificar la seva descripció a través d'un vincle en el mateix identificador. També pot accedir als seus conceptes mitjançant un botó.

### Llista d'entitats

id	Descripció	Acció
JUDI	JUDICIAL	<input type="button" value="Conceptes"/>
ORGPO	ORGANISME POLICIAL	<input type="button" value="Conceptes"/>
RECAP	RECAPTADOR	<input type="button" value="Conceptes"/>

TFG UOC Java EE, © 2017

- Formulari manteniment d'entitats de precinte

Des de aquest formulari l'usuari pot fer una alta, modificar o eliminar una entitat.



Gestió de precintes Home Altres Recerques **Administrador** Sortir

### Formulari d'entitats

ID

Descripció

TFG UOC Java EE, © 2017

El formulari de la llista de conceptes, llista de motius, formulari de concepte de precinte, formulari de motius de desprecinte són anàlegs al prototips de la llista d'entitat i formulari d'entitats.

Gestió de precintes Home Altres Recerques **Administrador** Sortir

### Llista de conceptes

id	Descripció
JUGJAT01	Judjat nro 1 primera instancia
JUGJAT02	Judjat nro 2 primera instancia
JUGJAT03	Judjat nro 3 primera instancia

TFG UOC Java EE, © 2017

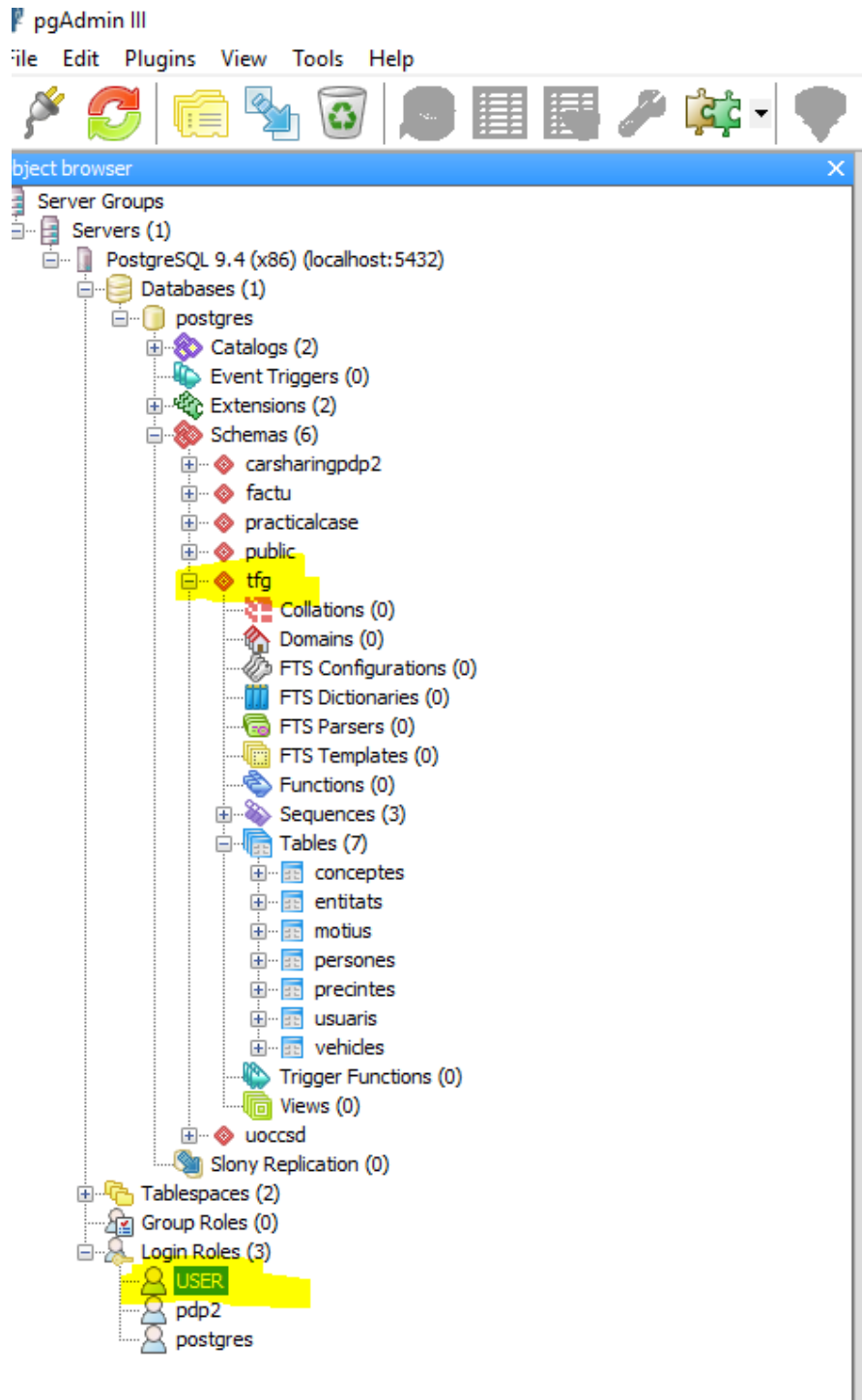
## 8. Instal·lació i execució

Prèviament a executar l'aplicació:

- Instal·lar el JDK 8 de Oracle que es pot descarregar de la pàgina oficial de Oracle.
- instal·lar la base de dades "PostgreSQL" que es pot descarregar des de la pàgina oficial <https://www.postgresql.org/>.

Cal seguir les instruccions per instal·lar-la i la versió que s'ha fet servir durant aquest projecte és la 9.4. Un cop s'ha instal·lat cal crear, mitjançant l'eina PGADMIN III, l'usuari USER amb la clau de pas PASSWORD i un esquema anomenat TFG en que el propietari sigui USER. Es proporciona un script que es pot llançar des de aquesta mateixa eina.

```
CREATE SCHEMA tfg
AUTHORIZATION "USER";
```



A continuació, per executar el lliurable, executar la comanda

```
C:\WINDOWS\system32\cmd.exe
C:\Users\BERNAT1\git\uoc\precintes\target>java -jar precintes-0.0.1-SNAPSHOT.jar
```

S'inicia un procés en el que s'engega un Tomcat

```

C:\Users\BERNAT1\git\uoc\precintes\target>java -jar precintes-0.0.1-SNAPSHOT.jar

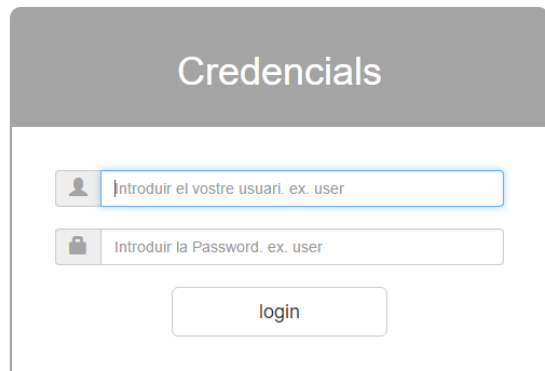
  ____  __
 / ___/  / /
/ /___ /  / /
/_____/___/_/

:: Spring Boot ::      (v1.4.6.RELEASE)

2017-06-06 16:53:13.805 INFO 14724 --- [           main] es.uoc.precintes.PrecintesApp      : Starting PrecintesApp v0.0.1-SNAPSHOT on BERNAT-PC7 with PID 14724 (C:\Users\BERNAT1\git\uoc\precintes\target\precintes-0.0.1-SNAPSHOT.jar started by BERNAT1 in C:\Users\BERNAT1\git\uoc\precintes\target)
2017-06-06 16:53:13.815 INFO 14724 --- [           main] es.uoc.precintes.PrecintesApp      : No active profile set, falling back to default profiles: default
2017-06-06 16:53:14.503 INFO 14724 --- [           main] ationConfigEmbeddedWebApplicationContext : Refreshing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@7591083d: startup date [Tue Jun 06 16:53:14 CEST 2017]; root of context hierarchy
2017-06-06 16:53:19.225 INFO 14724 --- [           main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat initialized with port(s): 8080 (http)
2017-06-06 16:53:19.249 INFO 14724 --- [           main] o.apache.catalina.core.StandardService : Starting service Tomcat
2017-06-06 16:53:19.251 INFO 14724 --- [           main] org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.13
2017-06-06 16:53:19.535 INFO 14724 --- [ost-startStop-1] o.a.c.c.c.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2017-06-06 16:53:19.537 INFO 14724 --- [ost-startStop-1] o.s.web.context.ContextLoader      : Root WebApplicationContext: initialization completed in 5038 ms
2017-06-06 16:53:19.810 INFO 14724 --- [ost-startStop-1] o.s.b.w.s.c.ServletFilterRegistrationBean : Mapping filter: 'web'

```

la url de entrada es <http://localhost:8080/> que per defecte ens porta a la pàgina de login



Els usuaris que poder entrar són: user pwd=user i admin pwd=admin

## 9. Glossari.

### JDK.

Java development Kit. És un programari que proveeix d'eines de desenvolupament per desenvolupar i executar aplicacions construïdes en el llenguatge Java.

### spring-data-jpa.

Mòdul de spring que maneja mitjançant hibernate la persistència amb la base de dades.

### **Querydsl.**

Querydsl proporciona una marc que permet construir consultes mitjançant una fluent-API de forma que en temps de compilació en queixa si no es fa referència a objectes del domini coneguts

### **Bootstrap.**

És un dels frameworks HTML, CSS i JavaScript més populars per desenvolupar aplicacions responsive, es a dir, aplicacions que s'ajusten a qualsevol tipus de dispositiu.

## **10. Conclusions.**

La meua percepció del treball ha sigut bona ja que en general he après moltes coses noves i n'he consolidat d'altres. Així i tot, la corba d'aprenentatge d'aquesta tecnologia és molt gran i encara em queda molt de temps fins que pugui tenir la soltesa necessària per poder ser productiu.

També m'ha ajudat la metodologia que he fet servir i que he après a diferents matèries que he impartit durant la carrera. Fer un bon anàlisi i disseny, documentar els casos d'ús, dibuixar els diagrames per tenir una visió general del projecte i saber quins components s'han d'implementar són tasques que en la fase de construcció s'agraeixen.

Finalment, i respecte a la impartició d'aquest TFG, la planificació que es fa a l'inici del semestre i la distribució de la càrrega de la feina en entregues parcials és una practica molt encertada ja que ens marca un ritme que es pot dur sense patir. A la fase de implementació, les iteracions d'un cas d'ús fins dominar l'arquitectura es una practica que segurament faré servir quan hagi d'aprendre qualsevol altre framework.

## **11. Bibliografia y referències.**

- **Craig Wells.** (2015). *Spring Cuarta Edición*. Madrid: Grupo Anaya.
- **Thymeleaf.** - <https://es.wikipedia.org/wiki/Thymeleaf>
- **Eclipse Foundation** - <http://www.eclipse.org/neon/>
- **Spring Boot** - <http://projects.spring.io/spring-boot/>

- PostgreSQL- <http://www.postgresql.org.es/>
- Querydsl - <http://www.querydsl.com/>
- Bootstrap - <http://getbootstrap.com/>