

Pràctiques de Laboratori

Sessions 5, 6 i 7

Lliurament 1

**Enginyeria del
Programari**

ETS Enginyeria
Informàtica
DSIC – UPV

Curs 2019-2020

Abast del lliurament

Al lliurament 1, l'alumnat presentarà el treball d'implementació que ha desenvolupat durant les darreres sessions de practiques relatives al cas d'estudi **EcoScooter**:

- Sessió 2 i 3: Escriptura de codi amb constructors
- Sessió 4: Capa de persistència amb Entity Framework.

Concretament, l'alumnat haurà de demostrar que ha implementat:

- La capa de negoci per a totes les classes del model de disseny presentat al butlletí de la pràctica 3.

La figura 1 mostra un fragment del model de cassos d'ús del cas d'estudi. Alguns d'ells (explicats posteriorment) hauran de ser implementats per al lliurament 1. **La figura no inclou tots els cassos d'ús del cas d'estudi.**

- La implementació completa de les capes de negoci y persistència, seguint les guies dels butlletins respectius, per als següents cassos d'ús (tal i com es va indicar als butlletins de les practiques 3 i 4 i als seminaris de classe de teoria):
 - **Registrar-se** (actor: Anònim)
 - **Inici sessió empleat** (actor: Empleat)
 - **Registre d'estació** (actor: Empleat d'administració)
 - **Registre de patinet** (actor: Empleat de manteniment)
 - **Llogar patinet** (actor: Usuari)
 - **Tornar patinet** (actor: Usuari)
 - **Registrar incident** (actor: Usuari)
 - **Inici sessió** (actor: Usuari)
 - **Obtenir recorreguts** (actor: Usuari)

NOTA: aquestos cassos d'ús hauran de ser implementats a les sessions 5, 6 i 7.

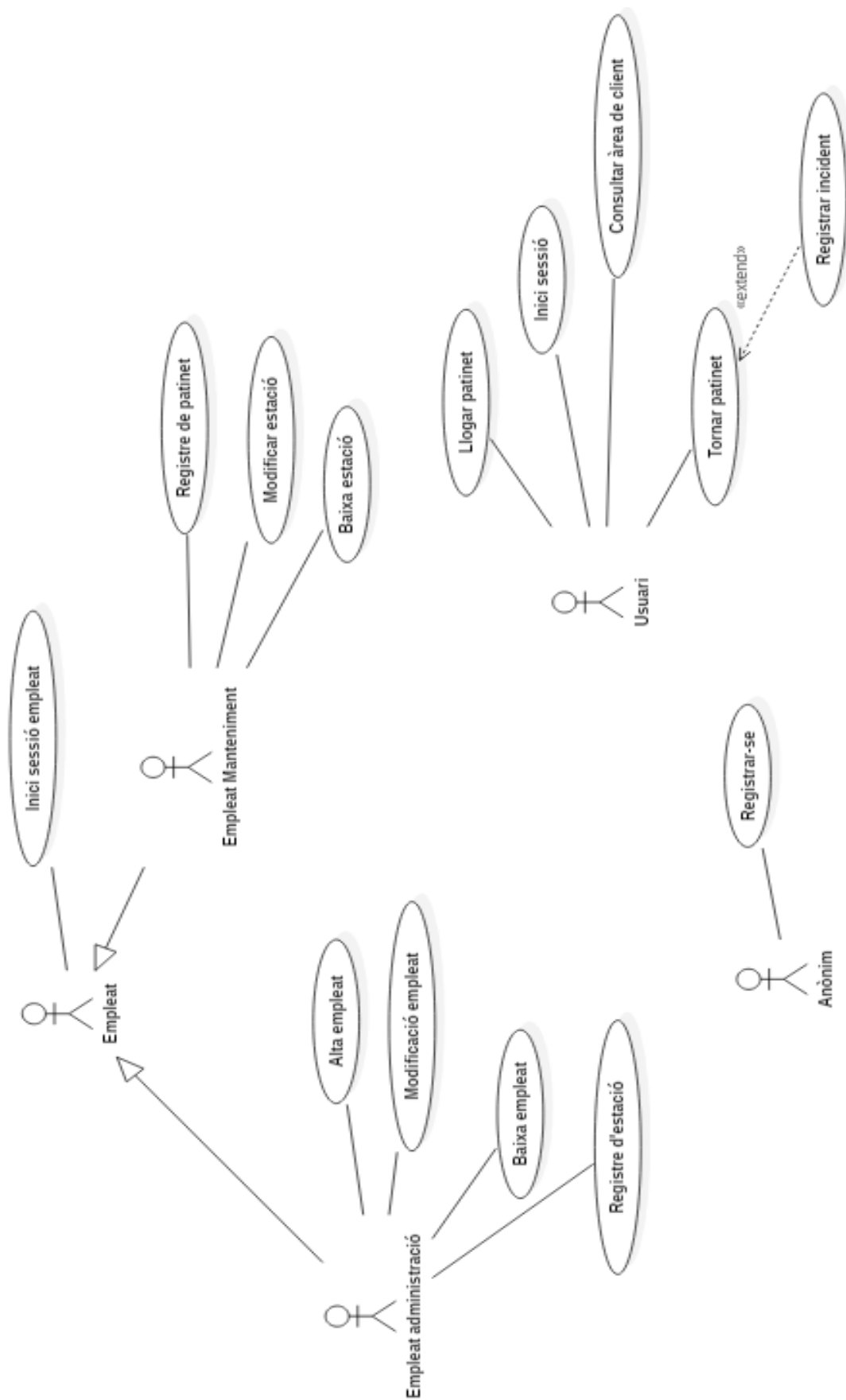


Figura 1. Model de Casos d'Us (fragment)

Descripció dels cassos d'ús a implementar

A continuació es detalla la descripció mitjançant una plantilla textual dels cassos d'ús. Per a aquesta entrega només es demanen les capes de negoci i persistència. **No es demana la capa d'interfície gràfica**, que es farà per a la següent entrega.

| | |
|-------------------|---|
| ID | 1 |
| Cas d'ús | Registrar-se |
| Actors | Anònim |
| Propòsit | Registrar-se com a usuari d'EcoScooter |
| Passos | <ol style="list-style-type: none">1. L'usuari introduïx la informació pròpia: nom, DNI, data de naixement, telèfon, correu electrònic i targeta de crèdit. (número de targeta, mes i any de caducitat i codi de control CVV), així com un nom d'usuari i una contrasenya.2. El sistema comprova que les dades són correctes (usuari major de 16 anys, targeta vàlida -és a dir, seqüència numèrica de 8 posicions i CVV- i nom d'usuari únic al sistema) i crea el nou usuari. |
| Precond | -- |
| Postcond | L'usuari és emmagatzemat al sistema. |
| Extensió síncrona | Al pas 2, si l'usuari proporciona informació incorrecta, el sistema mostrarà un error i el control passarà al pas 1. |

| | |
|-------------------|---|
| ID | 2 |
| Cas d'ús | Inici sessió usuari |
| Actors | Usuari |
| Propòsit | Identificar-se a l'aplicació |
| Passos | <ol style="list-style-type: none">1. L'usuari introduïx la informació d'inici de sessió (nom d'usuari i contrasenya).2. El sistema comprova les dades i permet l'accés a l'usuari. |
| Precond | -- |
| Postcond | L'usuari queda identificat a l'aplicació. |
| Extensió síncrona | Al pas 2, si l'usuari proporciona informació incorrecta o no existix l'usuari, el sistema mostrarà un error. |

El cas d'ús d'**Inici de sessió d'empleat** és similar a l'anterior, per la qual cosa no s'inclou la seua descripció textual.

| | |
|-------------------|--|
| ID | 3 |
| Cas d'ús | Registre d'estació |
| Actors | Empleat d'administració |
| Propòsit | Afegir una nova estació al sistema |
| Passos | <ol style="list-style-type: none">1. L'empleat introduirà la informació d'una nova estació: adreça (carrer, número i ciutat), coordenades GPS (latitud i longitud) i identificador d'estació.2. El sistema comprova la informació introduïda i crea la nova estació al sistema. |
| Precond | L'empleat ha iniciat sessió prèviament. |
| Postcond | La nova estació és emmagatzemada. |
| Extensió síncrona | Al pas 2, si l'empleat proporciona informació incorrecta o incompleta, o l'identificador de l'estació ja existix, el sistema mostrarà un error i el control torna al pas 1. |

| | |
|-------------------|--|
| ID | 4 |
| Cas d'ús | Registre de patinet |
| Actors | Empleat de manteniment |
| Propòsit | Afegir un nou patinet al sistema |
| Passos | <ol style="list-style-type: none">1. L'empleat introduirà la informació del nou patinet (data de registre i estat). Observe que l'identificador és autogenerat.2. El sistema validarà la informació introduïda. Si l'estat és disponible, el sistema demanarà l'estació a la que serà assignat.3. L'empleat introduirà l'estació.4. El sistema crearà el nou patinet. |
| Precond | L'empleat ha iniciat sessió prèviament. |
| Postcond | Un nou patinet queda enregistrat al sistema. |
| Extensió síncrona | Al pas 2, si la informació no és vàlida, es mostrarà un missatge d'error i el control tornarà al pas 1. Al pas 4, si l'estació no existix es mostrarà un missatge d'error i el control tornarà al pas 3. |

| | |
|-------------------|--|
| ID | 5 |
| Cas d'ús | Llogar patinet |
| Actors | Usuari |
| Propòsit | Llogar un patinet a una de les estacions |
| Passos | <ol style="list-style-type: none">1. L'usuari seleccionarà l'estació on llogarà el patinet.2. El sistema assignarà un dels patinets de l'estació.3. L'usuari retira el patinet.4. El sistema crea el nou lloguer. |
| Precond | L'usuari haurà iniciat sessió prèviament. |
| Postcond | El nou lloguer queda enregistrat al sistema. |
| Extensió síncrona | Al pas 2, si no queden patinets disponibles es mostrarà un missatge d'error. |

Nota: Els trackPoints i la velocitat del patinet no es tindran en consideració.

| | |
|-------------------|--|
| ID | 6 |
| Cas d'ús | Tornar patinet |
| Actors | Usuari |
| Propòsit | Tornar un patinet que ha estat llogat prèviament |
| Passos | <ol style="list-style-type: none"> 1. L'usuari selecciona "tornar patinet" i introduïx l'identificador de l'estació. 2. El sistema recupera l'últim lloguer de l'usuari i el número de sèrie del patinet. 3. El sistema pregunta si s'ha produït algun incident durant el lloguer. 4. L'usuari pot prémer "Sí" en el cas de que s'haja produït algun incident. En aquest cas, s'activa el cas d'ús de "Registrar incident". 5. El sistema actualitza la informació associada a la devolució (actualitza la data i l'hora del retorn i l'estació on s'ha produït. També canvia l'estat del patinet a disponible) i calcula el cost del lloguer. |
| Precond | L'usuari haurà iniciat sessió prèviament. |
| Postcond | El sistema emmagatzema la nova informació. |
| Extensió síncrona | <p>Al pas 1, si no existix l'identificador de l'estació, mostra un missatge d'error.</p> <p>Al pas 2, si l'últim lloguer ja té una devolució, es mostra un missatge d'error.</p> |

| | |
|-------------------|--|
| ID | 7 |
| Cas d'ús | Registrar incident |
| Actors | Usuari |
| Propòsit | Indicar l'existència d'un incident durant el lloguer d'un patinet. |
| Passos | <ol style="list-style-type: none"> 1. L'usuari introduïx una descripció del tipus d'incident i l'hora i dia en què s'ha produït. 2. El sistema actualitza la informació associada a un lloguer amb incident. |
| Precond | - |
| Postcond | El sistema emmagatzema la nova informació |
| Extensió síncrona | - |

| | |
|----------|--|
| ID | 8 |
| Cas d'ús | Obtenir recorreguts |
| Actors | Usuari |
| Propòsit | Obtenir un llistat dels recorreguts en un interval de dates. |

| | |
|-------------------|---|
| Passos | 1. L'usuari introduïx la data d'inici i de fi del període a consultar. 2. El sistema recuperarà els recorreguts de l'usuari en eixe període de temps. Cada recorregut inclou les següents dades separades per comes: data i hora d'inici i fi, preu, identificador de l'estació d'origen i de destí. |
| Precond | L'usuari haurà iniciat sessió prèviament. |
| Postcond | - |
| Extensió síncrona | Al pas 1, si l'interval de dates és invàlid, es mostrarà un missatge d'error. Al pas 2, si no existixen recorreguts es mostrarà un missatge. |

Detalls de la capa lògica

A les sessions anteriors de pràctiques se ha construït la part de la funcionalitat relacionada amb el disseny de les classes i el seu emmagatzemament persistent.

Per a aquest entregable s'ha d'implementar el controlador (o proveïdor de serveis) de la lògica de negoci. Aquest controlador haurà d'oferir a la capa superior (IU) accés a tota la funcionalitat que aquesta necessita.

El conjunt de tots els serveis oferits a la IU ha d'estar definit en una interfície anomenada, per exemple, `IEcoScooterService`. Els mètodes d'aquesta interfície s'implementaran en una classe anomenada `EcoScooterService`. La interfície i la seua implementació han d'estar a la carpeta `BusinessLogic/Services` i formar part de l'espai de noms `EcoScooter.Services`. Tots els errors que puguin aparèixer durant l'execució dels mètodes implementats s'hauran de reportar mitjançant l'ús d'excepcions.

Per a reportar els errors generats en la capa lògica, a la carpeta `BusinessLogic/Services` haurà de crear-se una subclasse d'`Exception` anomenada `ServiceException`. Quan es creen objectes d'aquesta classe per informar d'algun error, la propietat `Message` d'aquesta classe contindrà el missatge de text amb l'errada generada. Per exemple, en el cas d'estudi de referència que hi ha a `PoliformaT` anomenat *VehicleRental_ISW*, quan es vol afegir una persona cal tindre en compte si ja existeix una amb el mateix DNI que la que es vol crear. A continuació es pot veure un fragment de codi de mètode corresponent, on abans d'afegir una persona es comprova que no

existisca cap amb el mateix DNI. Si hi haguera alguna, s'envia una excepció.

```
public void addPerson(Person person)
{
    if (dal.GetById<Person>(person.Dni) == null)
    {
        dal.Insert<Person>(person);
        dal.Commit();
    }
    else throw new ServiceException("Person already exists.");
}
```

Amb l'objectiu de facilitar la identificació dels mètodes a implementar en *EcoScooterService*, de PoliformaT es podrà descarregar un projecte amb un programa de test de la lògica. Aquest programa **es podrà utilitzar com a base, que s'haurà d'adaptar a les necessitats concretes de cada disseny particular**. Concretament, aquest programa assumix l'existència d'un proveïdor de serveis (o controlador) que implementa determinats mètodes i els provarà. L'alumnat ha d'afegir a la carpeta Testing de la seua solució una nova carpeta de solucions amb el nom *BusinessLogicTest*, on hauran d'agregar el projecte de consola que descarregaran de PoliformaT anomenat *BusinessLogicTestApp*. Aquest programa té les proves mínimes que ha de superar l'*EcoScooterService* implementat.

A la Figura 2 es pot veure l'aspecte que té l'explorador de solucions per a l'exemple *VehicleRental_ISW* amb els continguts creats a la sessió. Destaca la interfície *IVehicleRentalService*, la seua implementació en una classe anomenada *VehicleRentalService*, la classe d'excepció *ServiceException* i el programa de test *BusinessLogicTestApp*.

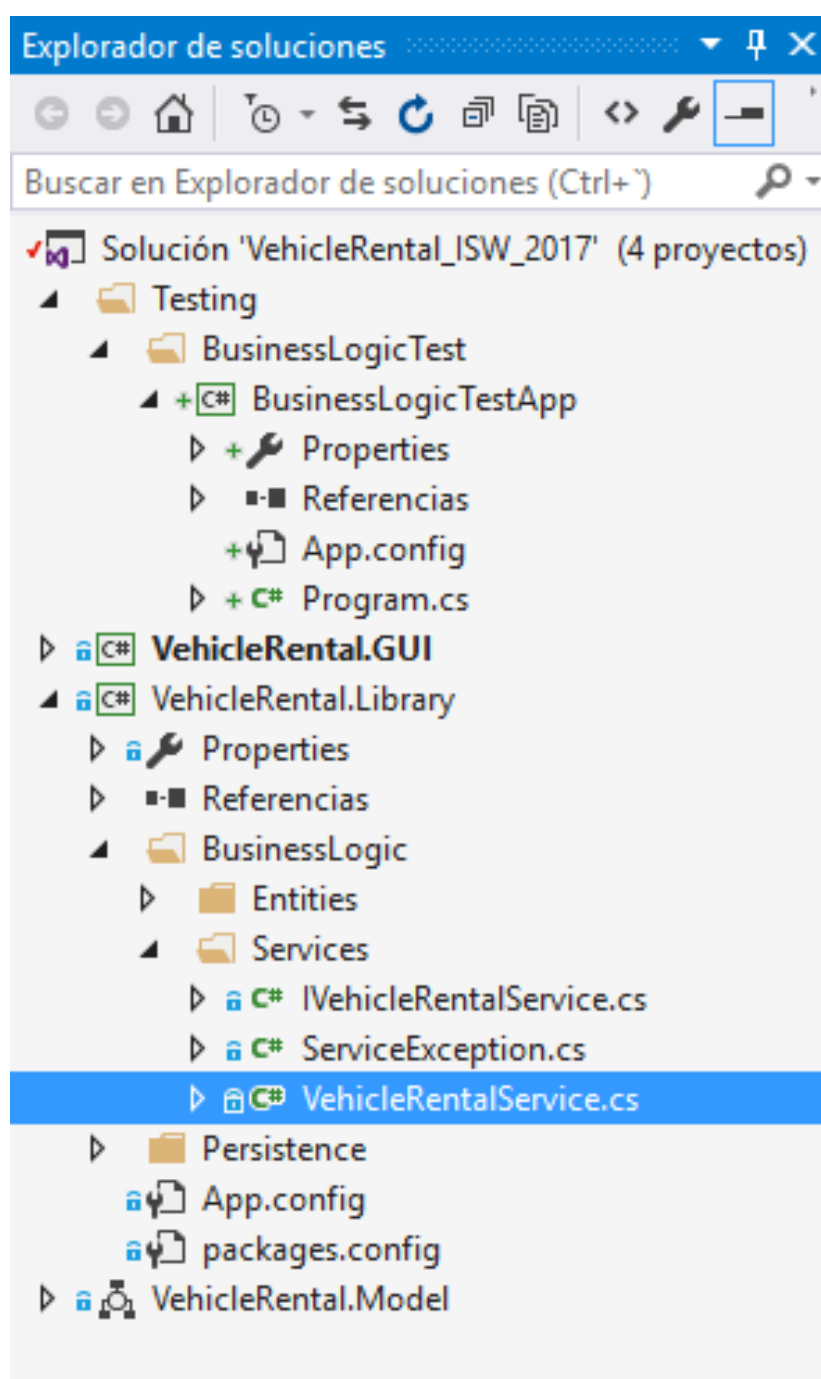


Figura 2. Explorador de solucions amb *Services*

Data d'entrega

La data de entrega serà la corresponent a la 8a setmana de pràctiques als laboratoris. Per al grup 3A – València serà el dimecres 27 de novembre de 2019. **TOTS** els membres del grup hauran d'acudir a la sessió, ja que seran requerits per contestar preguntes sobre el projecte.

Forma de Lliurament

Existirà una tasca a PoliformaT on cada grup haurà de pujar el fitxer comprimit del projecte. A més, es realitzarà una operació d'inserció al repositori GIT amb el comentari "Lliurament 1". El projecte s'ha de pujar a l'activitat de PoliformaT ABANS del començament de la sessió d'avaluació.

Avaluació

A la sessió de laboratori, el professorat avaluarà a cadascun dels grups, comprovant que s'ha implementat correctament la funcionalitat sol·licitada i **fent preguntes a TOTS els membres del grup**.