

ISW – PRÁCTICAS ENTREGABLE 1

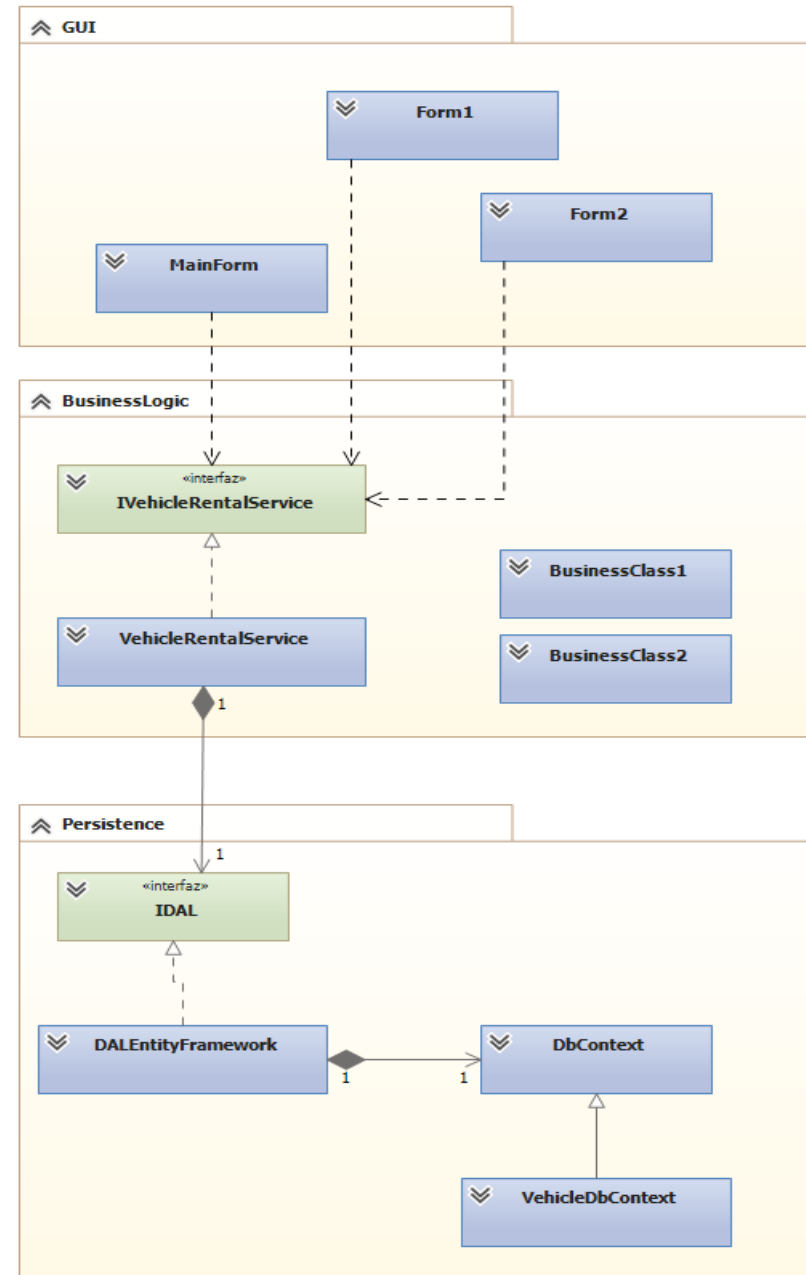
Casos de Uso– material de ayuda

ETS Ingeniería Informática
DSIC – UPV

Curso 2020-2021

Diseño de la separación de capas

- 3 capas:
 - Presentación (IGU)
 - Lógica de Negocio
 - Clase que implementa los servicios de la aplicación (casos de uso).
 - Persistencia

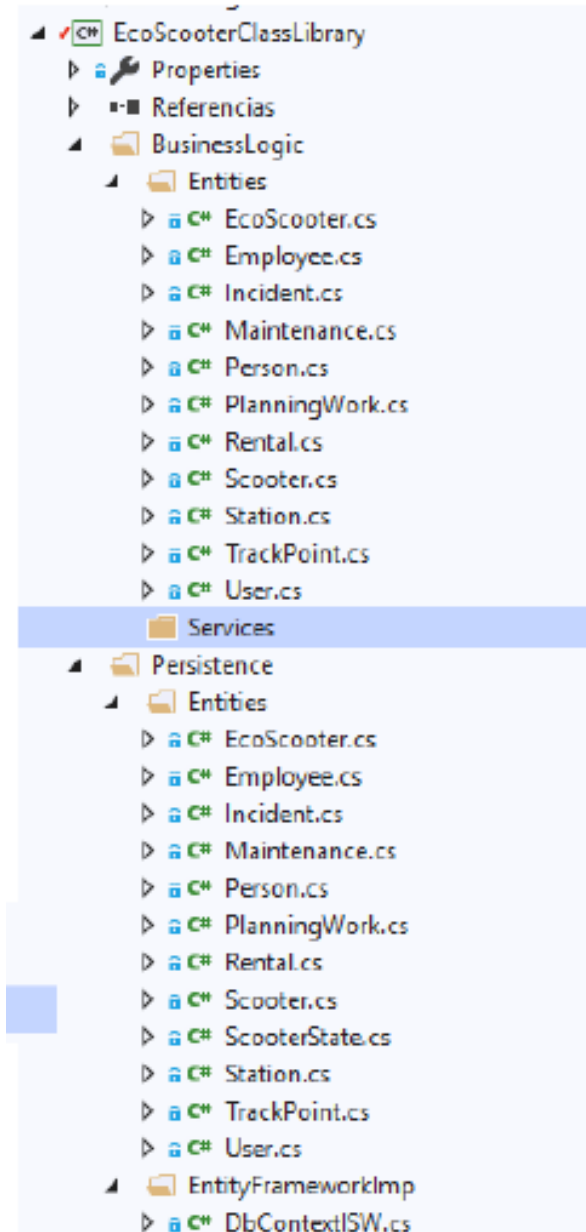


Capa Lógica

Nueva carpeta ***Services*** en la carpeta ***BusinessLogic***

Esta carpeta contendrá:

- ***IGestDepService.cs***
- ***ServiceException.cs***
- ***Populate.cs***
- ***GestDepService.cs (a implementar)***

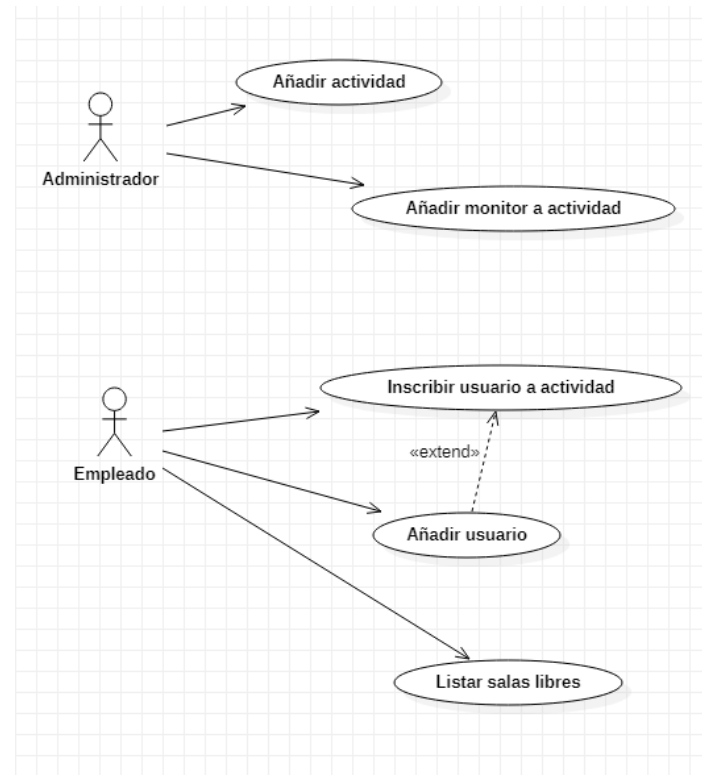


Capa Lógica

```
public class GestDepService : IGestDepService
{
    private readonly IDAL dal;
    public CityHall cityHall;
    public Gym gym;
    public GestDepService(IDAL dal)
    {
        this.dal = dal;
        try
        {
            if (dal.GetAll<CityHall>().Count() == 0) //No cityHall in the system. Data initialization.
            {
                bool CLEAR_DATABASE = true;
                int ROOMS_NUMBER = 9;
                int INSTRUCTORS_NUMBER = 5;
                Populate populateDB = new Populate(CLEAR_DATABASE, dal);
                cityHall = populateDB.InsertCityHall();
                gym = populateDB.InsertGym(cityHall); //Also in cityHall.First();
                populateDB.InsertRooms(ROOMS_NUMBER, gym); //Now available from gym.rooms;
                populateDB.InsertInstructors(INSTRUCTORS_NUMBER, cityHall); //Now available from cityHall.People;
            }
            else
            {
                //Retrieve the CityHall stored
                cityHall = dal.GetAll<CityHall>().First();

                if (cityHall.Gyms.Count > 0)
                {
                    //Retrieve the Gym stored
                    gym = cityHall.Gyms.First();
                }
                else
                {
                    //Adding Rooms and Gym
                    bool MANTAIN_DATABASE = false;
                    int ROOMS_NUMBER = 9;
                    Populate populateDB = new Populate(MANTAIN_DATABASE, dal);
                    gym = populateDB.InsertGym(cityHall);
                    populateDB.InsertRooms(ROOMS_NUMBER, gym);
                }
                int INSTRUCTORS_NUMBER = 5;
                if (dal.GetAll<Instructor>().Count() == 0) //No instructors
                {
                    bool MANTAIN_DATABASE = false;
                    Populate populateDB = new Populate(MANTAIN_DATABASE, dal);
                    populateDB.InsertInstructors(INSTRUCTORS_NUMBER, cityHall); //Now available from cityHall.People;
                }
            }
        }
        catch (Exception e)
        {
            throw new ServiceException("Error in the service init process", e);
        }
    }
}
```

Diagrama de casos de uso



Se implementa como si únicamente tuviésemos un actor.

Implementación de casos de uso

- Estamos trabajando con un lenguaje orientado a objetos.
- Las operaciones propias de las clases (por ejemplo, comprobar igualdad) **deben implementarse en las clases y no en los servicios.**
- Las colecciones no deben obtenerse directamente con llamadas al DAL (capa de persistencia), sino a través de los objetos de la capa de lógica de negocio que tienen dichas colecciones.
- Por ejemplo, la lista de *Activities* no debe pedirse al DAL, sino a *Gym*.