INMO

Aplicación Web

Análisis y Diseño de Sistemas
Proyecto Integrador

Característica Deseables

Aplicación Web

Múltiplos Dispositivos

Fácilmente Escalable

Componentes Independientes

Web Service

MVC

↓

API

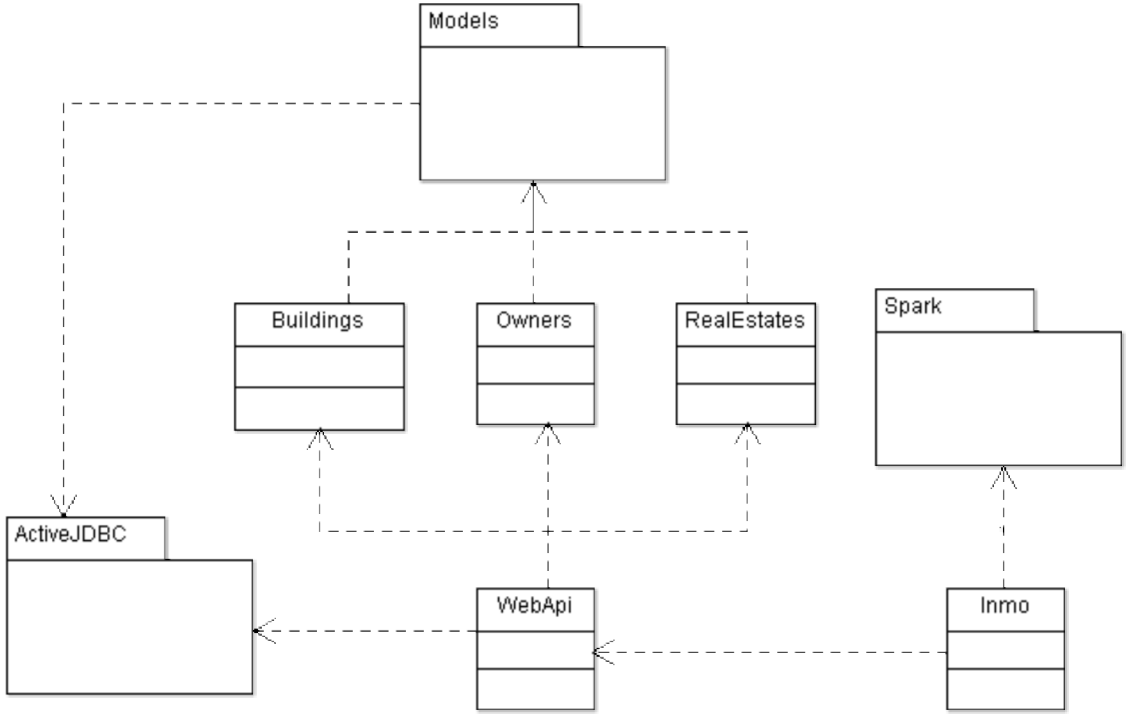
↓

REST

INMO

Aplicación Web

Diagrama de Clases



ActiveJDBC - Models

El acceso a los datos se realiza mediante ActiveJDBC.

Las clases del modelo se encarga de corroborar que los datos que se intentan guardar en la base de datos cumplan con las restricciones necesarias.

Adicionalmente, el diseño de la base de datos incluye todas las mismas restricciones que serán controladas por el SGBD.

Manejo de Datos

Owners - **Buildings** - **RealEstates**

Métodos Estáticos

Métodos Comunes

- Add
- Modify
- Delete

Owners - **RealEstates**: implementan métodos para gestionar la relación entre estas. Para realizar esto, RealEstates invoca las funciones de Owners.

Consulta de Datos

Todas las funciones que consulta información retornan un ***String*** con formato ***JSON***.

Internamente estas funciones trabajan con ***LazyLists*** (provistas por **ActiveJDBC**). Estas listas poseen un método que recursivamente las transforma en ***JSON***.

Consulta de Datos

```
public static String getOwnersByCity(String city){
    //Busco los Owners
    String where = "";
    if (city != ""){where = "city='"+city+"'";}

    LazyList<Owner> ownerList = Owner.where(where).include(RealEstate.class);

    return ownerList.toJson(false,"id","first_name","email","street",
                            "last_name","neighborhood","city");
}
```

Consulta de Datos

Buildings

- getBuildings
- getBuildingTypes

⟨ String[] city
⟨ String pMin
⟨ String pMax

Owners

- getOwnersByCity ⟨ String city

RealEstates

- getRealEstatesByCity ⟨ String city

INMO
Aplicación Web

WEBAPI

Intermediario entre el Servidor(view), el modelo de datos y sus clases de gestión.

Ofrece métodos públicos para los clientes que deseen acceder a los datos.

getOwners
getBuildings
addOwner
bindOwnerRealEstate
getRealEstates
getBuildingTypes
addRealEstte

connect: Método privado encargado de conectar a la base de datos y comprobar que la conexión esta abierta cada vez que se necesita consultar datos.

WEBAPI

```
private void connect(){
    if (!Base.hasConnection()){
        Base.open("com.mysql.jdbc.Driver",
            "jdbc:mysql://localhost/inmoapp_development", "root", "");
        System.out.println("---> Se conecto a la base de datos."+Base.connection()+"\n");
    }
}

public String getOwners(String city)
{
    connect();
    return Owners.getOwnersByCity(city);
}
```

INMO - Server

El Servidor define cada una de las rutas a las que responderá tanto para los **GET**s como para los **POST**s y responderá los pedidos utilizando los métodos provistos por **WEBAPI** (controler).

Nota: por razones de implementación de la aplicación de Testing *(TestWebApp)* los POST están implementados como GET y por esta misma razón, se debió definir un método ***makeJSONP*** que de ser necesario, transforma un JSON tradicional en uno encapsulado en una función de callback para posibilitar las peticiones *crossdomain*.

¿preguntas?

INMO
Aplicación Web