

MyParallelOrg  
DEDUCCIÓN DE JERARQUÍAS  
ALTERNATIVAS EN UNA COMPAÑÍA EN  
BASE A INTERACCIONES DE  
USUARIOS

Anexo III. Especificación de diseño



TRABAJO DE FIN DE GRADO  
INGENIERÍA INFORMÁTICA

Yoel Pérez Carrasco

Dirigido por Fernando De La Prieta Pintado y Walter Vinci

## Tabla de contenidos

1. Introducción.....	5
2. Modelo de Diseño.....	6
2.1. Patrón MVVM.....	6
2.1.1. Patrón DAO.....	7
2.2. Subsistemas de Diseño.....	8
2.3. Clases de Diseño.....	9
2.3.1. Server.....	9
2.3.1.1. Model.....	9
2.3.1.2. Back-End DAO.....	10
2.3.1.3. ControllerApi.....	11
2.3.2. Frontend.....	11
2.3.2.1. Components.....	12
2.3.2.2. Views.....	13
2.3.2.3. Utils.....	14
2.3.2.4. Authcontext.....	15
2.4. Realización Casos de Uso.....	15
2.4.1. Paquete Gestión de Autenticación.....	15
2.4.2. Paquete Gestión Usuarios.....	16
2.4.3. Paquete Gestión Visualización 3D.....	17
2.4.4. Paquete Gestión Organizaciones.....	17
2.4.5. Paquete Gestión API Github.....	18
3. Diseño de la base de Datos.....	19
4. Modelo de Despliegue.....	20
5. Bibliografía.....	21

## Lista de figuras

Figura 1.Modelo - Vista - Vista/Modelo.....	5
Figura 2: Patrón DAO.....	6
Figura 3.Subsistemas de Diseño.....	7
Figura 4: Model.....	9
Figura 5.Back-End DAO.....	9
Figura 6 .ControllerApi.....	10
Figura 7 .Components.....	11
Figura 8 .Views.....	12
Figura 9 .Utils.....	13
Figura 10 .Authcontext.....	14
Figura 11 .Diagrama Actividad Login Usuario.....	14
Figura 12 .Diagrama Actividad Cerrar Sesión.....	15
Figura 13 .Diagrama Secuencia Ver Usuarios.....	15
Figura 14 .Diagrama Secuencia Cagar grafo 3D.....	16
Figura 15 .Diagrama Secuencia Listar Organizaciones.....	16
Figura 16 .Diagrama Secuencia Obtener Datos API.....	17
Figura 17.Diagrama Diseño Base De Datos.....	18
Figura 18.Modelo de Despliegue.....	19



# 1. Introducción

Este anexo expone el diseño del sistema propuesto, proporcionando un enfoque detallado de cómo se materializan físicamente los casos de uso dentro del sistema. Este modelo de objetos constituye la base para describir los elementos clave que permiten la implementación eficiente del sistema. La estructura del anexo se organiza de la siguiente manera:

- **Modelo de diseño:** Este apartado está compuesto por varios elementos que detallan la arquitectura del sistema y sus componentes.
- **Diseño de la base de datos:** Aquí se muestra un diagrama que ilustra las entidades de datos que el sistema almacenará, así como las relaciones entre ellas, garantizando un acceso y gestión de la información eficiente y coherente.
- **Modelo de despliegue:** Se proporciona un modelo de objetos que describe cómo se distribuyen físicamente los componentes del sistema en los distintos nodos, detallando la funcionalidad de cada uno y cómo se interrelacionan. [\[1\]](#)

## 2. Modelo de Diseño

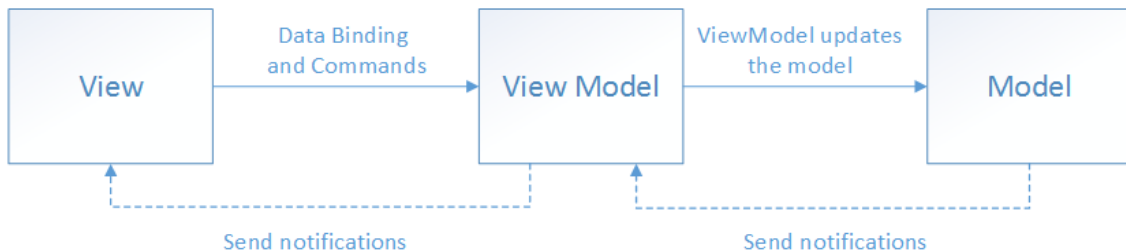
### 2.1. Patrón MVVM

La plataforma sigue un modelo de desarrollo basado en el patrón MVVM (Model-View-ViewModel) que proporciona una estructura clara para gestionar la interacción entre la interfaz de usuario y la lógica del sistema. Este patrón es especialmente útil para aplicaciones que requieren una separación efectiva entre la interfaz y la lógica, permitiendo un flujo de datos eficiente.

El patrón MVVM estructura el sistema en tres componentes principales:

- **Modelo:** Contiene y gestiona los datos del sistema, actuando como la representación lógica de los mismos.
- **Vista:** Es la interfaz de usuario que presenta los datos y elementos interactivos al usuario final.
- **Modelo de Vista (ViewModel):** Funciona como un intermediario entre el modelo y la vista, permitiendo la actualización bidireccional de los datos.

Al utilizar el patrón MVVM, la aplicación garantiza que la lógica de presentación y los datos se mantengan separados, facilitando el mantenimiento, las pruebas y la escalabilidad de la aplicación. [2]



*Figura 1. Modelo - Vista - Vista/Modelo*

### 2.1.1. Patrón DAO

El patrón **DAO (Data Access Object)** es un componente de software que proporciona una interfaz estándar entre la aplicación y los sistemas de almacenamiento de datos, como bases de datos o archivos. A través de este patrón, la lógica de negocio puede interactuar con el almacenamiento de datos utilizando métodos CRUD (Crear, Leer, Actualizar y Eliminar) sin preocuparse por los detalles específicos del acceso a esos datos.

El patrón DAO se organiza en:

**BusinessObject:** El objeto que solicita el acceso a los datos.

**DataAccessObject:** Abstrae el acceso a la fuente de datos.

**DataSource:** Implementa la fuente de datos.

**TransferObject:** Objeto que se intercambia entre el BusinessObject y el DAO.[3]

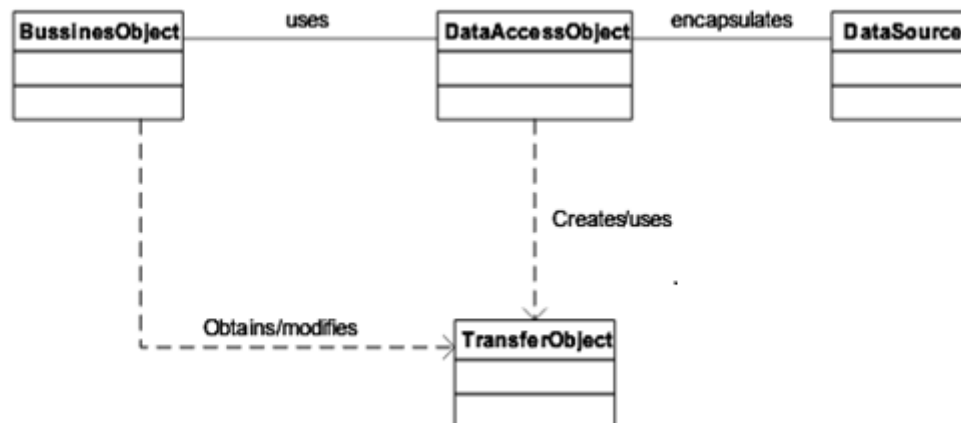
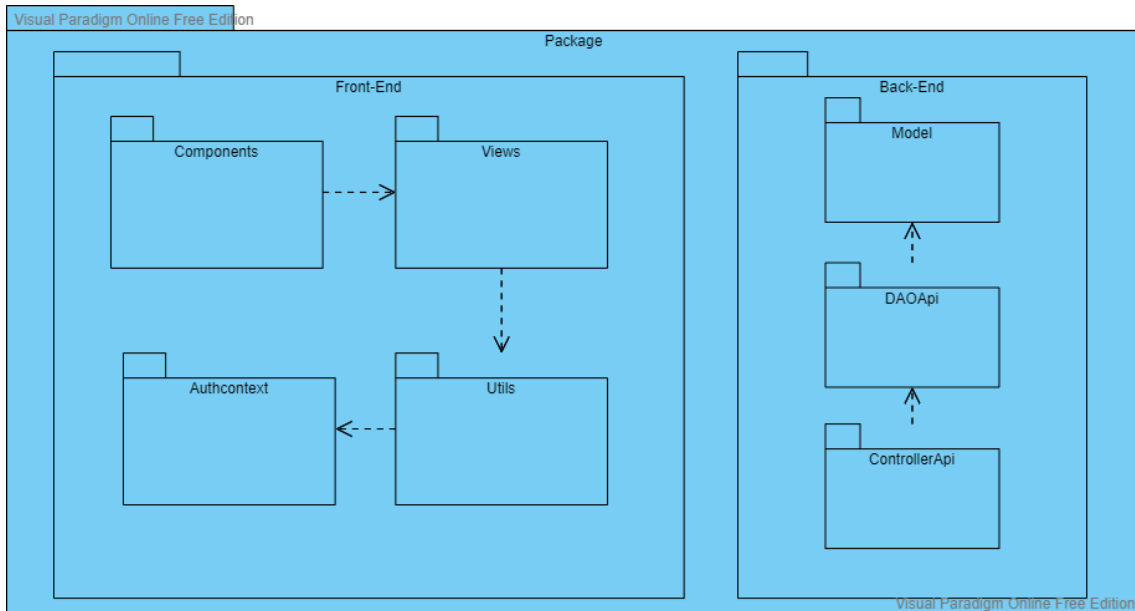


Figura 2: Patrón DAO

## 2.2. Subsistemas de Diseño

El sistema se ha dividido en diferentes subsistemas para facilitar su manejo y modularización:



*Figura 3. Subsistemas de Diseño*



## 2.3. Clases de Diseño

**Back-End;** Este subsistema contiene la lógica y la gestión de los datos del sistema. Los datos persistentes se almacenan en el paquete Model, que representa la estructura y las relaciones de los datos. La especificación del patrón DAO se encuentra en el paquete DAOApi, que gestiona el acceso a los datos y su persistencia. La lógica de control y la interfaz de comunicación con otras partes del sistema se implementan en el paquete ControllerApi.

**Front-End:** Representa la interfaz con la que interactúa el usuario. En este subsistema se utiliza un enfoque modular con diferentes paquetes:

- **Components:** Contiene componentes reutilizables que se pueden utilizar en varias vistas.
- **Views:** Gestiona las diferentes vistas de la aplicación y su presentación al usuario.
- **AuthContext:** Administra el contexto de autenticación del sistema, asegurando el control de acceso y la identificación del usuario.
- **Utils:** Almacena utilidades y funciones auxiliares que son utilizadas por otros paquetes dentro del front-end.

### 2.3.1. Server

#### 2.3.1.1. Model

El sistema gestiona el almacenamiento de datos de forma persistente en la base de datos, considerando las relaciones entre ellos. Esto incluye las clases que están asociadas a User, GitHub User, Commit User Github y Contribute User Github, así como las relaciones y cardinalidades correspondientes.

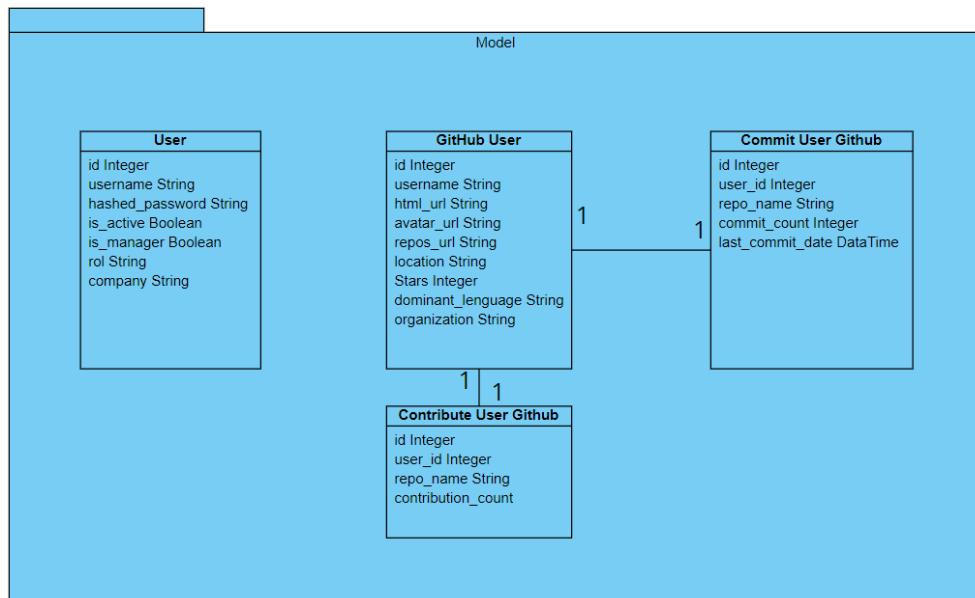


Figura 4: Model

#### 2.3.1.2. Back-End DAO

En este paquete se encuentran las clases relacionadas con el acceso a los datos. Implementa los métodos para obtener, crear, modificar y eliminar los datos del modelo.

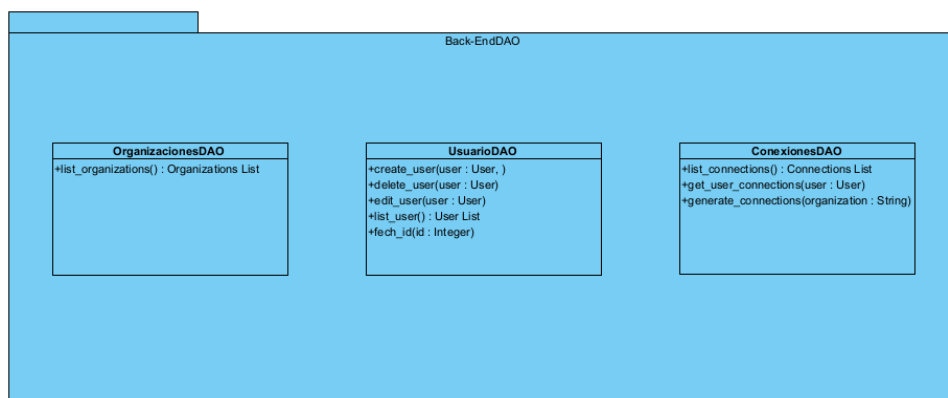


Figura 5.Back-End DAO

### 2.3.1.3. ControllerApi

Este paquete describe el funcionamiento de las peticiones y funcionalidades de la APILocal:

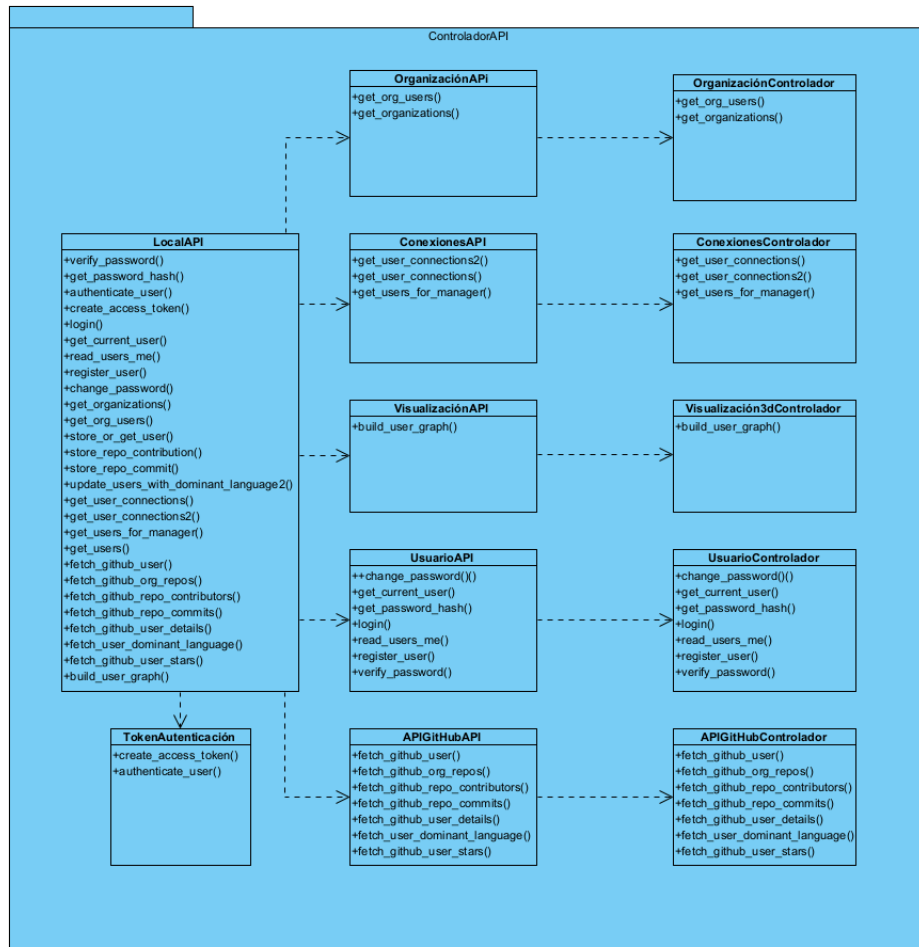
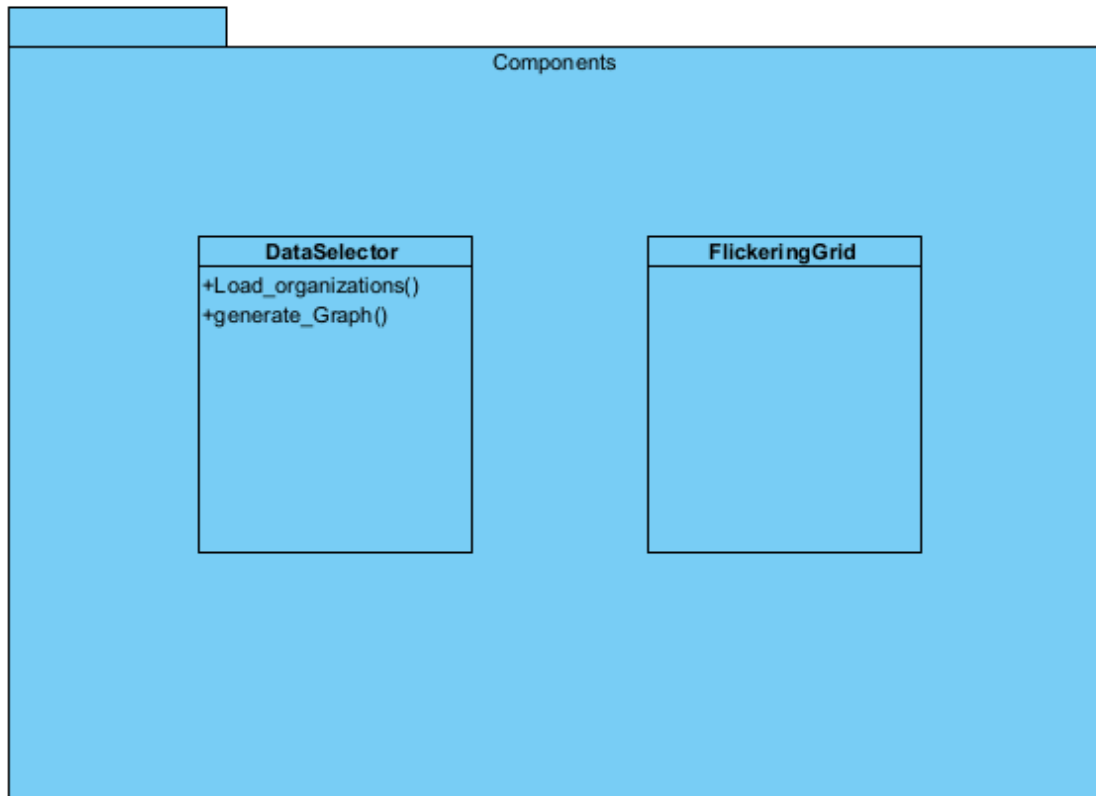


Figura 6 .ControllerApi

## 2.3.2. Frontend

Este apartado se describirán los diferentes tipos de de vistas y funcionalidades de esta:

### 2.3.2.1. Components



*Figura 7 .Components*

### 2.3.2.2. Views

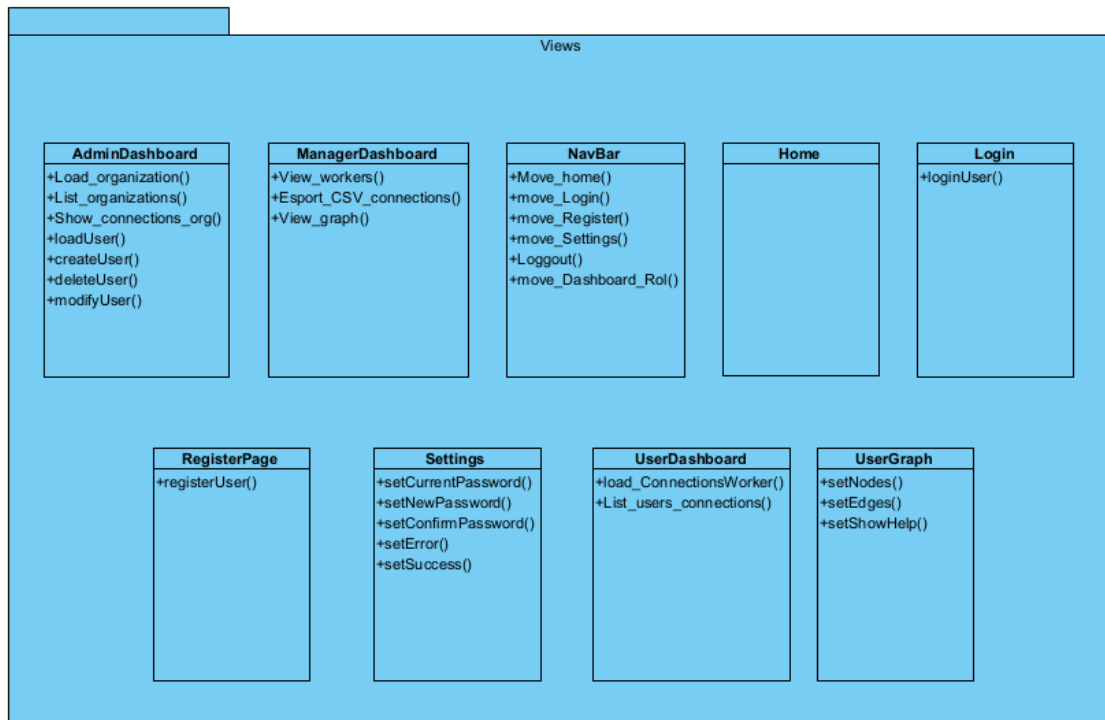
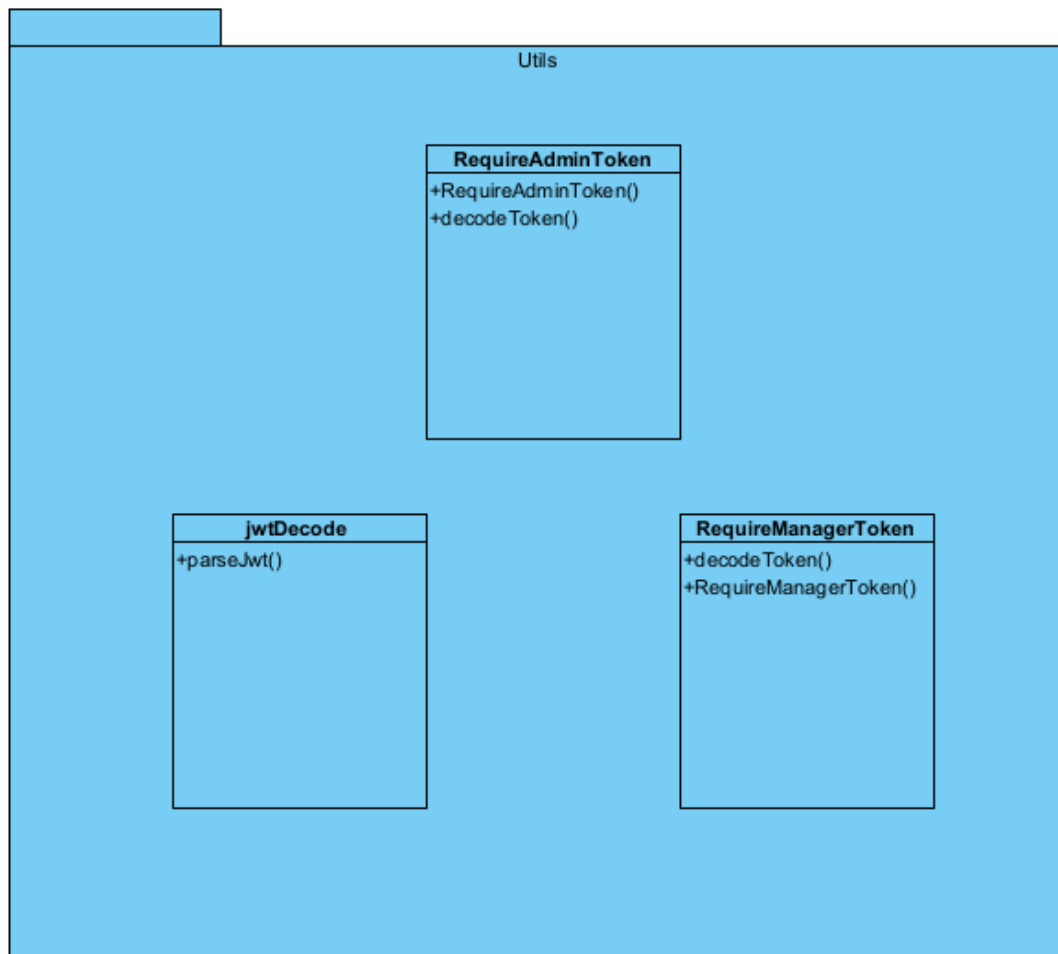


Figura 8 .Views

### 2.3.2.3. Utils



*Figura 9 .Utils*

#### 2.3.2.4. Authcontext

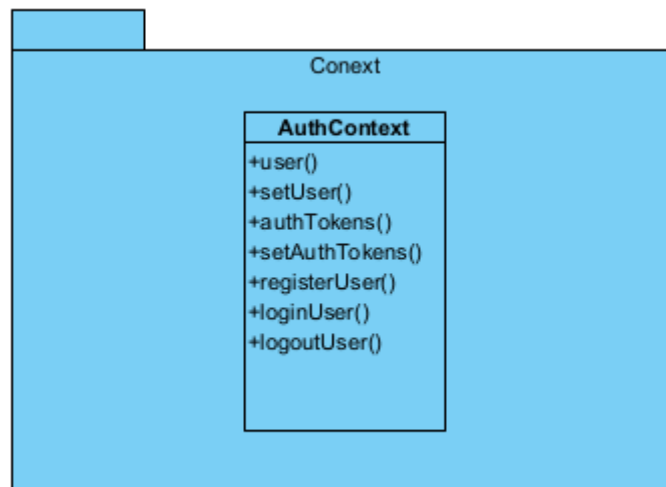


Figura 10 .Authcontext

### 2.4. Realización Casos de Uso

En este apartado apartado describiremos un diagrama de secuencia y de actividad de cada paquete de objetivos del sistema como ejemplificación del diseño general del sistema:

#### 2.4.1. Paquete Gestión de Autenticación

Login Usuario:

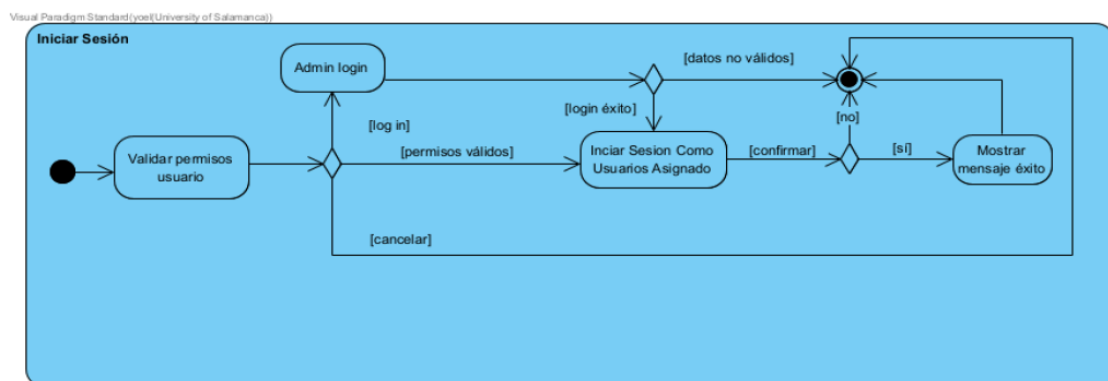


Figura 11 .Diagrama Actividad Login Usuario

Cerrar Sesión:

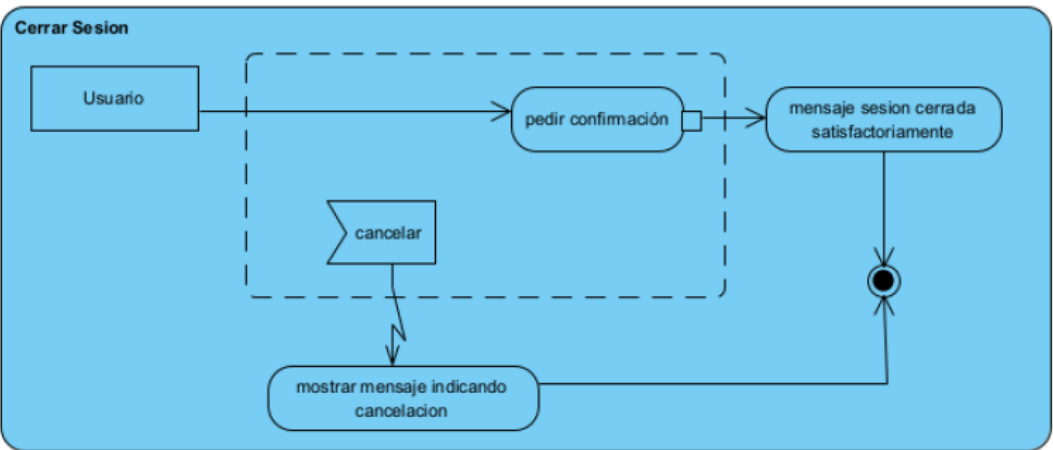


Figura 12 .Diagrama Actividad Cerrar Sesión

2.4.2. Paquete Gestión Usuarios

Ver Usuarios:

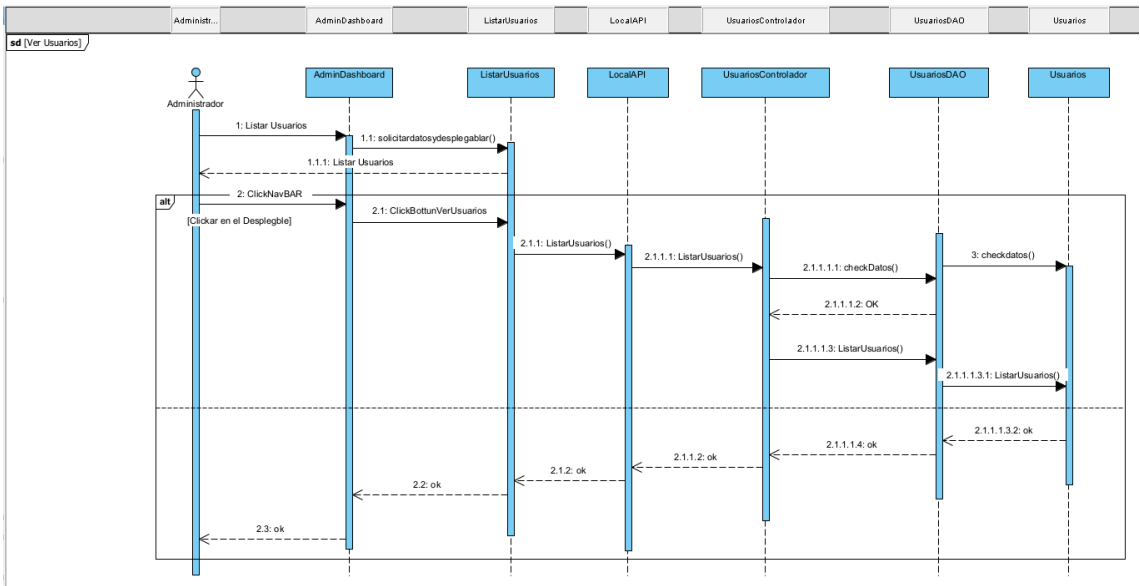


Figura 13 .Diagrama Secuencia Ver Usuarios



### 2.4.3. Paquete Gestión Visualización 3D

Cagar grafo 3D:

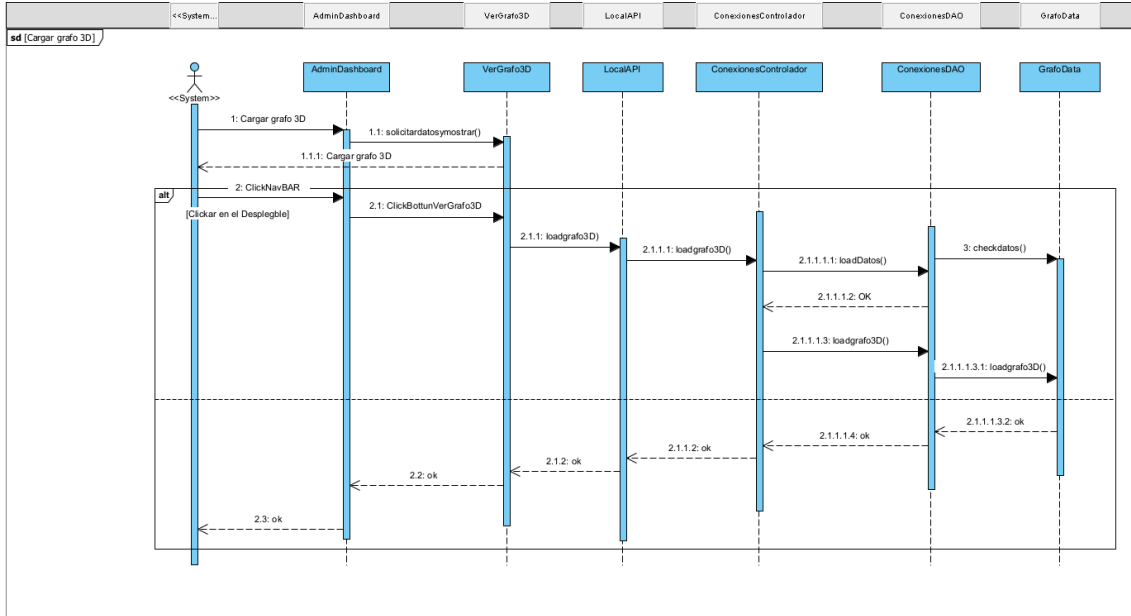


Figura 14 .Diagrama Secuencia Cagar grafo 3D

### 2.4.4. Paquete Gestión Organizaciones

Listar Organizaciones:

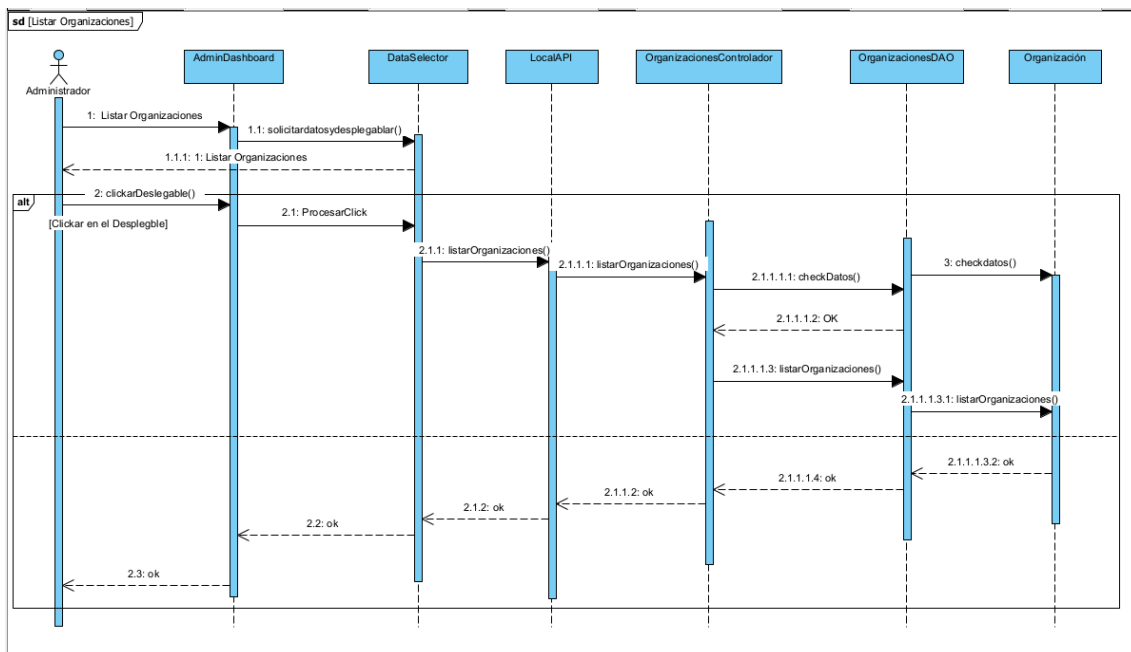


Figura 15 .Diagrama Secuencia Listar Organizaciones

## 2.4.5. Paquete Gestión API Github

Obtener Datos API:

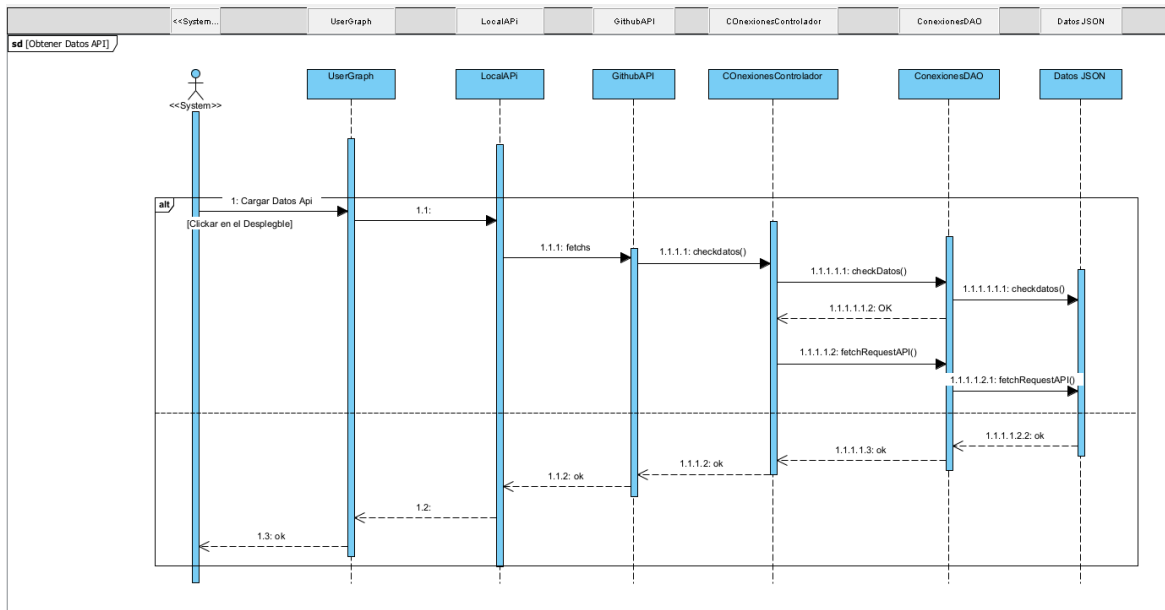
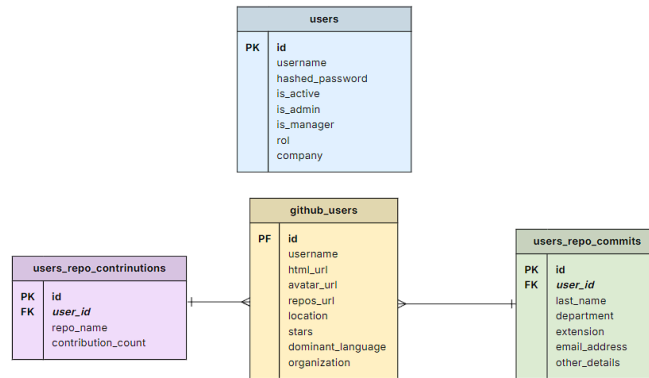


Figura 16 .Diagrama Secuencia Obtener Datos API

### 3. Diseño de la base de Datos

**Diagrama Diseño Base De Datos**



*Figura 17. Diagrama Diseño Base De Datos*

#### 4. Modelo de Despliegue

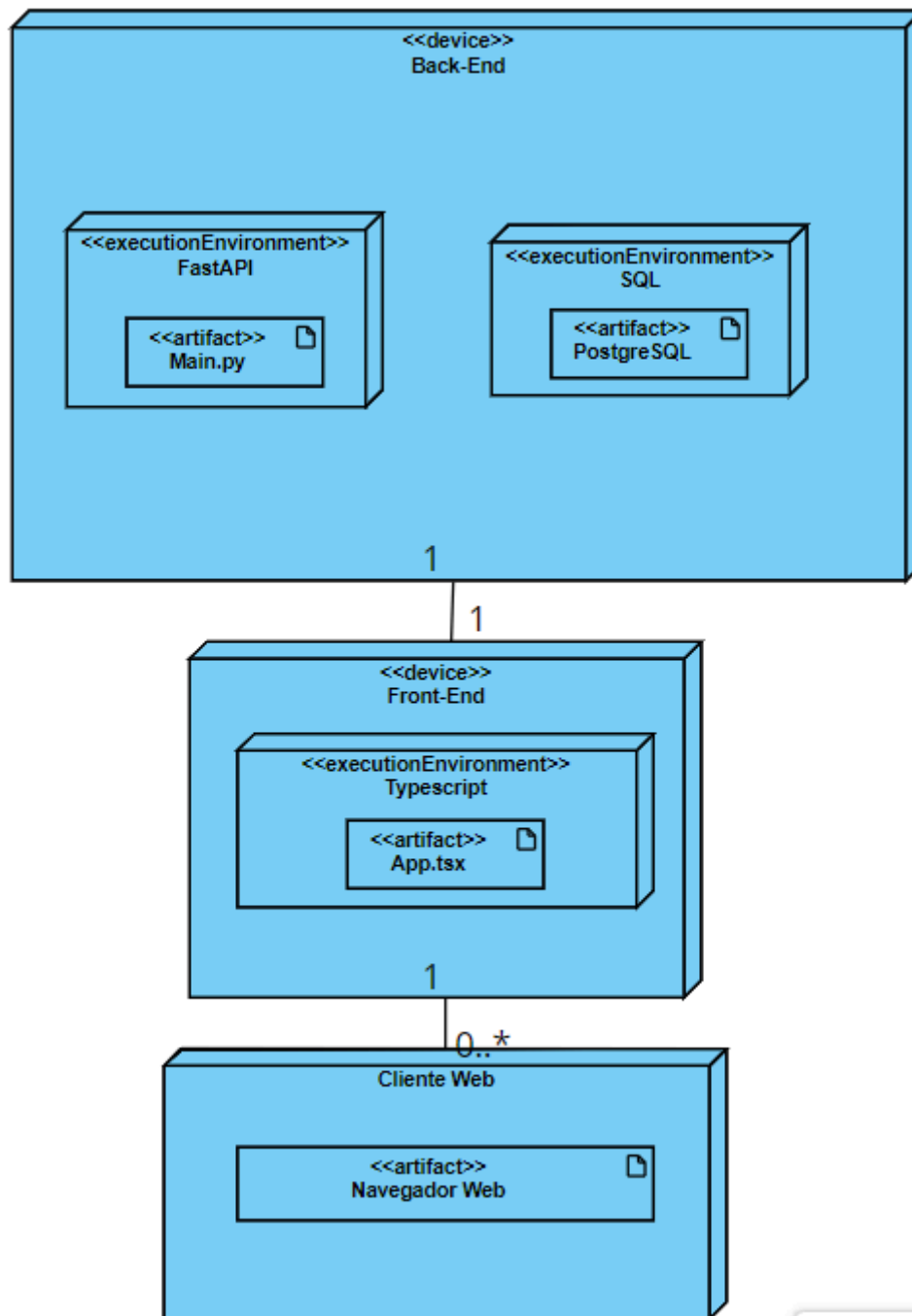


Figura 18. Modelo de Despliegue

## 5. Bibliografía

1. GP-GII: Diapositivas. (s. f.). Recuperado 3 de septiembre de 2024, de <https://studium23.usal.es/mod/resource/view.php?id=126798>
2. GP-GII: Diapositivas. (s. f.). Recuperado 3 de septiembre de 2024, de <https://studium23.usal.es/mod/resource/view.php?id=126808>