

MyParallelOrg

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



TRABAJO DE FIN DE GRADO INGENIERÍA
INFORMÁTICA

Yoel Pérez Carrasco

Dirigido por Fernando De La Prieta Pintado y Walter Vinci

Salamanca, septiembre 2024

Certificado de los Tutores

D. Fernando De La Prieta Pintado y D. Walter Vinci, profesor del Departamento de Informática y Automática de la Universidad de Salamanca y Master Software Specialist en HP-SCDS respectivamente.

CERTIFICAN:

Que el trabajo titulado “MyParallelOrg deducción de jerarquías alternativas en una compañía en base a interacciones de usuarios” ha sido realizado por D. Yoel Pérez Carrasco, con el número de documento ****1992Y y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo Fin de Grado de la Titulación Grado de Ingeniería Informática en esta Universidad.

Y para que así conste a todos los efectos oportunos.

Firmas de los tutores:

--	--

En Salamanca, a 04 de septiembre de 2024.

Resumen

En todas las empresas existe una jerarquía y una división en equipos/áreas/organizaciones fijadas por los objetivos de la compañía y los equipos de dirección. Sin embargo, en entornos colaborativos y proyectos compuestos por múltiples componentes, los trabajadores interactúan a través de diversas herramientas, creando una suerte de “organización paralela” en la que puede que trabajen más con gente de diferentes equipos al suyo o que reciban peticiones por parte de personas que no están directamente en su vertical jerárquica.

Este proyecto tiene como objetivo la creación de una aplicación que permita gestionar y visualizar las conexiones entre los empleados de una empresa, utilizando información recolectada a través de la API de GitHub. A partir de los commits y contributors, se generan perfiles de los trabajadores de una empresa ficticia, lo que permitió establecer conexiones entre ellos basadas en distintos criterios. Estas conexiones no solo reflejan las relaciones dentro de los mismos equipos de trabajo, sino que también revelan interacciones entre distintos grupos.

Para facilitar la interpretación de estas relaciones, se ha desarrollado una visualización en 3D de un grafo, que permite observar la estructura de conexiones en la interfaz gráfica. El objetivo principal de la aplicación es ordenar y visualizar de manera paralela las interacciones entre los empleados, ayudando a identificar las relaciones más allá de los límites de los grupos de trabajo convencionales. Gracias a estas conexiones filtradas, se pudieron generar grupos de trabajo interrelacionados de forma más eficiente.

Palabras clave: [React](#), [Control de versiones](#), [Organización jerarquía de empresas](#), [APIs](#), [web applications](#), [ML](#), [AI](#), [github](#).

Summary

In all companies there is a hierarchy and division into teams/areas/organisations established by company goals and management teams. However, in collaborative environments and multicomponent projects, workers interact through various tools, creating a kind of ‘parallel organisation’ in which they can work more with people from teams other than their own or receive requests from people who are not directly in their hierarchical vertical.

This project aims to create an application to manage and visualise the connections between employees in a company, using information collected through the GitHub API. From the commits and contributors, profiles of the employees of a fictitious company are generated, allowing to establish connections between them based on different criteria. These connections not only reflect the relationships within the same work teams, but also reveal the interactions between different groups.

To facilitate the interpretation of these relationships, a 3D visualisation of a graph has been developed that allows the structure of the connections to be observed in the graphical interface. The main objective of the application is to sort and visualise in parallel the interactions between employees, helping to identify relationships beyond the boundaries of conventional work groups. Thanks to these filtered connections, interlinked working groups could be generated more efficiently.

Keywords: React, version control, organisation hierarchy, APIs, web applications, ML, AI, github.

Lista de contenidos

1. Introducción.....	9
2. Objetivos.....	10
2.1. Objetivos del Sistema.....	11
2.2. Objetivos Personales.....	11
3. Conceptos Teóricos.....	12
3.1. API (Application Programming Interface).....	12
3.2. Clustering.....	13
3.3. Commit (GitHub).....	13
3.4. Grafo.....	13
4. Técnicas y Herramientas.....	14
4.1. Front-End.....	14
4.1.1. TypeScript.....	14
4.1.2. HTML.....	14
4.1.3. CSS.....	14
4.1.4. Axios.....	14
4.1.5. BootStrap.....	14
4.1.6. Tailwind CSS.....	15
4.2. Back-End.....	15
4.2.1. FastAPI.....	15
4.2.2. NPM.....	15
4.2.3. PostgreSQL.....	15
4.2.4. Docker.....	15
4.2.5. Jose.....	15
4.2.6. Pydantic.....	16
4.2.7. Networkx.....	16
4.2.8. Sqlalchemy.....	16
4.3. IDE.....	16
4.3.1. Visual Studio Code.....	16
4.4. Herramientas de Gestión de Proyectos.....	16
4.4.1. Microsoft Project.....	16
4.4.2. Visual Paradigm.....	16
4.4.3. EZestimate.....	17
4.5. Herramientas de Generación de Documentación.....	17
4.5.1. Sphinx.....	17
5. Aspectos Relevantes del Desarrollo.....	17
5.1. Marco de Trabajo.....	17
5.2. Estimación de la Duración del Proyecto.....	18
5.3. Planificación Temporal.....	20

5.4. Especificación de Requisitos.....	20
5.4.1. Actores del Sistema.....	21
5.4.2. Objetivos del Sistema.....	22
5.4.3. Requisitos de Información.....	23
5.4.4. Requisitos No Funcionales.....	24
5.4.5. Requisitos Funcionales.....	25
5.5. Diseño del Sistema.....	29
5.5.1. Patrón Modelo-Vista-Modelo de Vista (MVVM).....	29
5.5.2. Patrón DAO.....	30
5.5.3. Subsistemas de Diseño.....	31
5.5.4. Clases de Diseño.....	31
5.5.5. Realización de Casos de Uso (Diseño).....	37
5.5.6. Diseño de la Base de Datos.....	38
5.5.7. Modelo de Despliegue.....	40
5.6. Implementación.....	40
5.7. Pruebas.....	41
5.8. Funcionalidad del Sistema.....	42
5.8.1. Instalación y Configuración.....	43
5.8.2. Inicio.....	43
5.8.3. Autenticación y Registro.....	44
5.8.3.1. Registro de Usuarios.....	44
5.8.3.2. Login de Usuarios.....	45
5.8.4. Vista del Worker.....	46
5.8.5. Vista del Manager.....	47
5.8.6. Barra de Navegación (Navbar).....	49
5.8.6.1. Logotipo y Enlaces Principales.....	50
5.8.6.2. Opciones de Usuario (Ajustes y Cerrar Sesión).....	51
5.8.7. Visualización Gráfica 3D.....	52
5.8.7.1. Nodos y Conexiones.....	53
5.8.7.2. Controles de Navegación.....	54
5.8.8. Vista de Administrador.....	57
5.8.8.1. Gestión de Organizaciones.....	58
5.8.8.2. Carga de Nuevas Organizaciones.....	58
5.8.9. Gestión de Usuarios.....	58
5.8.9.1. Listar Usuarios:.....	59
5.8.10. Crear Usuario:.....	60
6. Limitaciones del Sistema.....	61
7. Conclusiones y Líneas de Trabajo Futuras.....	62
7.1. Conclusiones.....	62
7.2. Líneas de Trabajo Futuras.....	63
7.2.1. Back-End.....	63

7.2.1.1. Mejoras en las conexiones entre usuarios:.....	63
7.2.1.2. Migración a una infraestructura cloud:.....	63
7.2.2. Front-End.....	63
7.2.2.1. Desarrollo de nuevas vistas:.....	63
7.2.2.2. Incorporación de notificaciones y chat:.....	63
7.2.2.3. Mejora en la gestión de roles de usuario:.....	63
8. Bibliografía.....	64

Lista de tablas

Tabla 1. ACT-03.....	22
Tabla 2. OBJ-03.....	23
Tabla 3. IRQ-05.....	25
Tabla 4. RNF-01.....	26
Tabla 5. ACT-03.....	26
Tabla 6. Tabla de UC.....	27
Tabla 7. UC-26.....	28
Figura 8.Modelo - Vista - Vista/Modelo.....	30
Figura 9: Patrón DAO.....	31

Lista de figuras

Figura 1. Diagrama MT.....	20
Figura 2. EZ.....	21
Figura 3.Techical complexity.....	21
Figura 4.Efort.....	21
Figura 5. Projects.....	22
Figura 6. Paquete 3D.....	30
Figura 7. diagrama de paquetes.....	31
Figura 10.Subsistemas de Diseño.....	33
Figura 11.subsistema controlador.....	34
Figura 12.subsistema Back-EndDAO.....	35
Figura 13.subsistema model.....	36
Figura 14.subsistema vista components.....	36
Figura 15.subsistema vista utils.....	37
Figura 16.subsistema vista Views.....	38
Figura 17.subsistema vista Context.....	38
Figura 18.Diagrama de Secuencia.....	39
Figura 19. Diagrama de estados I.....	40
Figura 20. Diagrama de estados II.....	40
Figura 21. Diseño de la Base de Datos.....	41
Figura 22. Diseño de la Base de Datos.....	42
Figura 22. Prueba 3D.....	44
Figura 23. Vista Inicio Bienvenida.....	46
Figura 24. Vista Inicio Utilidades y Referencias.....	46
Figura 25. Vista Registro.....	47
Figura 26. Vista Registro Organización.....	47
Figura 27. Vista Login.....	48
Figura 28. Vista Dashboard de Worker.....	49
Figura 29. Vista Dashboard de Worker sin conexiones.....	49
Figura 30. Vista Dashboard de Manager.....	50
Figura 31. Vista Dashboard de Manager Utilidades.....	51
Figura 32. Vista Dashboard de Manager Trabajadores.....	51
Figura 33. Vista NavBar.....	52
Figura 34. Vista NavBar.....	52
Figura 35. Vista NavBar No Login.....	53
Figura 36. Vista NavBar Login.....	53
Figura 37. Vista NavBar Desplegable.....	53
Figura 38. Vista Ajustes.....	54
Figura 39. Vista Cerrar sesión.....	54

Figura 40. Vista Visualización 3D.....	55
Figura 41. Vista Nodos.....	56
Figura 42. Vista Hub Herramientas Visualización.....	57
Figura 43. Vista Dashboard Administrador.....	59
Figura 44. Vista Dashboard Administrador Seleccionar Organización.....	60
Figura 45. Vista Dashboard Administrador Cargar Organización.....	60
Figura 46. Vista Dashboard Administrador Listar Usuarios.....	61
Figura 47. Vista Listar Usuarios.....	61
Figura 48. Vista Listar Usuarios Acciones.....	62
Figura 49. Vista Dashboard Administrador Crear Usuario.....	62
Figura 50. Vista Crear Usuario.....	63

1. Introducción

Hoy en día toda empresa que posea un mínimo de infraestructura interna desarrollada se enfrenta a multitud de problemas cotidianos, como puede ser, mal entendidos por falta de documentación de un lenguaje que no conoce todo el equipo o no coincidir todos los miembros del grupo de trabajo puesto que los ciclos de noche son opuesto.

Por esto cada vez más empresas deciden invertir dinero en la creación de estos llamados “grupos de trabajo ideales”, o al menos intentar que dentro de estos, los trabajadores se encuentren con el mejor ambiente laboral posible.

Que esto pueda llevarse a cabo depende en gran medida a la gestión de recursos humanos internos de la empresa, los cuales no siempre aciertan, por eso una forma de agrupar estas características comunes entre trabajadores, por ejemplo, el afán por el mismo lenguaje de programación, que comparten la misma franja horario o el nivel de reputación que tiene cada miembro dentro de la empresa; esto puede ayudar al arduo trabajo que es crear estas jerarquías ideales.

Este trabajo fin de grado propuesto por HP-SCDS intenta cubrir esta necesidad, a través de interacciones en GitHub, simulando que una organización dentro de GitHub es una empresa real y que mediante las interacciones (Commits/Contributions) de los usuarios de la plataforma con los repositorios dar por hecho que forman parte de dicha empresa como trabajadores.

Una vez ya se han filtrado todos los trabajadores de la empresa y están localizables, se procede a su recopilación en la base de datos con información extra, que será la que forme la heurística de clusterización. Esta estará compuesta por el lenguaje de programación más usado por el trabajador, el continente en el que reside, y reputación.

Tras esta contextualización del trabajo, ya se podrá entender mejor todos los contenidos de esta memoria.

2. Objetivos

En esta sección se establecen los objetivos que debe cumplir el sistema desarrollado en el Trabajo de Fin de Grado. Se distinguen entre los objetivos del sistema y los objetivos personales.

2.1. Objetivos del Sistema

El objetivo general de este proyecto es la manipulación de datos recolectado de la api de github

Gestión de autenticación: El sistema debe poder proveer un sistema seguro a través de tokens de autenticación web,

Gestión de visualización 3D: El sistema debe poder realizar la visualización de todos los nodos y aristas representando los trabajadores y sus conexiones dentro de la empresa.

Gestión de organizaciones: El sistema debe poder permitir la carga de nuevas organizaciones dentro de la plataforma y a su vez administrarlas.

Gestión de usuarios: El sistema debe poder ofrecer diferentes tipos de rol como así crear, modificar, eliminar y ver los diferentes usuarios de la plataforma.

Gestión de conexiones: El sistema debe poder tratar dichos criterios entre usuarios de la empresa para generar estas conexiones y después clusterizar las, pasarlas a nodos y aristas, para así poder tener un visión más concreta.

Gestión de API GitHub: El sistema debe poder gestionar todas la comunicaciones con la API de github, gracias a un token generado por el propio GitHub, se podrá acceder a todo el dataset para poder descubrir las posibles conexiones que existan entre los empleados de la organización.

2.2. Objetivos Personales

En este apartado se exponen los objetivos personales que se intentan cumplir tras realizar el proyecto y los motivos a realizarlo.

Durante la realización de este proyecto, me planteé una serie de objetivos personales basados en mis intereses profesionales y en el deseo de ampliar mis conocimientos en diversas áreas. En primer lugar, quería aplicar los conceptos y habilidades adquiridos a lo largo de la carrera, especialmente en el desarrollo de front-end y back-end, donde me siento particularmente competente. Este proyecto me ofreció la oportunidad de poner en práctica esos conocimientos y consolidarlos en un entorno real.

Además, tenía un fuerte interés en explorar nuevas tecnologías y herramientas que no había utilizado previamente, como la API de GitHub, Reagraph y técnicas avanzadas de clustering. Este proyecto me permitió investigar cómo los datos de grandes repositorios de código abierto podían recolectarse, generalizarse y luego modelarse para ser visualizados de manera clara y eficiente.

Finalmente, un objetivo personal importante fue desafiarne a nivel técnico y profesional. Este proyecto requería integrar varias áreas del desarrollo, desde el back-end con FastAPI y Docker hasta la visualización en front-end con React, lo que me permitió abordar un reto integral que involucra tanto la investigación como el desarrollo práctico. Además, desde que vi este proyecto en la lista de trabajos propuestos por HP-SCDS, me sentí atraído por la relevancia del tema, especialmente por la popularidad de GitHub como foro de colaboración en proyectos Open Source, lo que aumentó mi motivación para llevarlo a cabo.

3. Conceptos Teóricos

En este apartado se detallarán teóricamente conceptos que se dan por supuestos en el desarrollo del sistema y la documentación para la buena comprensión de la misma.

3.1. API (Application Programming Interface)

Una API (del inglés, application programming interface, en español, interfaz de programación de aplicaciones) es una pieza de código que permite a dos aplicaciones comunicarse entre sí para compartir información y funcionalidades. Se usan generalmente en bibliotecas de programación.²

Por ejemplo, si se tiene una app de recetas en un teléfono móvil y al operar esta app se hace una búsqueda, se puede utilizar una API para que esta aplicación se comunique con el sitio web de recetas, solicite las recetas que cumplen con los criterios de búsqueda, retorne los resultados y posteriormente tras que el usuario seleccione la receta deseada, solicite y realice la descarga de la misma.

De este modo, una API es un puente para conectar diferentes aplicaciones y hacer que trabajen de manera más eficiente y efectiva.

Existen diferentes tipos de API, como REST y SOAP, dependiendo del tipo de comunicación y transferencia de datos. [\[1\]](#)

3.2. Clustering

El clustering es una tarea que tiene como finalidad principal lograr el agrupamiento de conjuntos de objetos no etiquetados, para lograr construir subconjuntos de datos conocidos como Clusters. Cada cluster dentro de un grafo está formado por una colección de objetos o datos que a términos de análisis resultan similares entre sí, pero que poseen elementos diferenciales con respecto a otros objetos pertenecientes al conjunto de datos y que pueden conformar un clúster independiente.

Este tipo de proceso es aplicado en modelos de machine learning de tipo no supervisado. Gracias a su implementación el sistema puede analizar los datos, realizar la tarea y encontrar los posibles errores dentro de su funcionamiento. El Clustering, en este caso sirve para segmentar datos en grupos de dimensiones similares en base a características para facilitar este proceso. [\[2\]](#)

3.3. Commit (GitHub)

El comando git commit captura una instantánea de los cambios preparados en ese momento del proyecto. Las instantáneas confirmadas pueden considerarse como versiones "seguras" de

un proyecto: Git no las cambiará nunca a no ser que se lo pidas expresamente. Antes de ejecutar git commit, se utiliza el comando git add para pasar o "preparar" los cambios en el proyecto que se almacenarán en una confirmación. Estos dos comandos, git commit y git add, son dos de los que se utilizan más frecuentemente.[\[3\]](#)

3.4. Grafo

Un grafo en el ámbito de las ciencias de la computación es un tipo abstracto de datos (TAD), que consiste en un conjunto de nodos (también llamados vértices) y un conjunto de arcos (aristas) que establecen relaciones entre los nodos. El concepto de grafo TAD desciende directamente del concepto matemático de grafo.[\[4\]](#)

Formalmente se define así $G=(V,A)$, siendo V un conjunto cuyos elementos son los vértices del grafo y A un conjunto cuyos elementos son las aristas, las cuales son pares (ordenados si el grafo es dirigido) de elementos en V .

- **Vértices (o nodos):** Son los puntos o entidades que representan los objetos.
- **Aristas (o arcos):** Son las conexiones entre los vértices, que representan la relación o interacción entre ellos.

4. Técnicas y Herramientas

4.1. Front-End

4.1.1. TypeScript

TypeScript es un superconjunto tipado de JavaScript que agrega tipos estáticos y facilita el desarrollo de aplicaciones a gran escala. Se compila a JavaScript estándar, asegurando compatibilidad con cualquier navegador [5].

4.1.2. HTML

HTML (HyperText Markup Language) es el lenguaje estándar utilizado para estructurar el contenido web. Define los elementos básicos como encabezados, párrafos, imágenes y enlaces.[6]

4.1.3. CSS

CSS (Cascading Style Sheets) es el lenguaje utilizado para definir la presentación visual de un documento HTML, controlando el diseño, colores, fuentes y otros aspectos visuales.[7]

4.1.4. Axios

Es una biblioteca basada en Promesas para hacer solicitudes HTTP desde el navegador. Se utiliza comúnmente para interactuar con APIs, simplificando las operaciones de CRUD.[8]

4.1.5. BootStrap

Bootstrap es un framework de desarrollo web de código abierto que proporciona una colección de herramientas de diseño preconstruidas, como componentes UI y plantillas basadas en HTML, CSS y TypeScript.[9]

4.1.6. Tailwind CSS

Tailwind CSS es un framework CSS utilitario que permite aplicar estilos directamente en los elementos HTML mediante clases predefinidas, evitando la creación de hojas de estilo separadas.[22]

4.2. Back-End

4.2.1. FastAPI

FastAPI es un framework web para construir APIs con Python, conocido por su alta velocidad y capacidad para manejar grandes volúmenes de solicitudes gracias a su uso de "asyncio"[\[11\]](#)

4.2.2. NPM

NPM (Node Package Manager) es el gestor de paquetes para Node.js que permite gestionar dependencias y módulos JavaScript necesarios para el desarrollo de aplicaciones.[\[12\]](#)

4.2.3. PostgreSQL

PostgreSQL es un sistema de base de datos de código abierto que se usa para almacenar y gestionar información. Es conocido por ser muy confiable y flexible. Admite datos complejos como JSON, es muy seguro y cumple con las normas de bases de datos, garantizando que las transacciones se realicen de manera correcta y segura.[\[13\]](#)

4.2.4. Docker

Docker es una plataforma que permite empaquetar aplicaciones y sus dependencias en contenedores, garantizando que funcionen de la misma manera independientemente del entorno en el que se ejecuten.[\[14\]](#)

4.2.5. Jose

Jose es una biblioteca en Python que implementa JSON Web Tokens (JWT) y otros estándares de criptografía relacionados con JSON para la autenticación y autorización en aplicaciones.[\[15\]](#)

4.2.6. Pydantic

Pydantic es una biblioteca de validación de datos para Python que usa la definición de modelos basados en anotaciones de tipos para garantizar la integridad y validez de los datos.[\[16\]](#)

4.2.7. Networkx

NetworkX es una biblioteca en Python diseñada para la creación, manipulación y análisis de gráficos y redes complejas.[\[17\]](#)

4.2.8. Sqlalchemy

SQLAlchemy es un ORM (Object-Relational Mapper) para Python que permite interactuar con bases de datos SQL utilizando un enfoque basado en objetos en lugar de sentencias SQL puras.[\[18\]](#)

4.3. IDE

4.3.1. Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente gratuito desarrollado por Microsoft, popular por su soporte para múltiples lenguajes, extensiones, integración con Git y capacidades avanzadas de depuración.[\[19\]](#)

4.4. Herramientas de Gestión de Proyectos

4.4.1. Microsoft Project

Microsoft Project es una herramienta de gestión de proyectos desarrollada por Microsoft. Permite a los gestores de proyectos planificar, organizar y realizar un seguimiento de tareas, cronogramas y recursos. A través de sus funciones, como la creación de diagramas de Gantt, la asignación de recursos, y el seguimiento del progreso de tareas, ayuda a optimizar la planificación y ejecución de proyectos complejos. Además, es muy utilizado para gestionar presupuestos y controlar la carga de trabajo de los equipos.[\[20\]](#)

4.4.2. Visual Paradigm

Visual Paradigm es una herramienta CASE (Computer-Aided Software Engineering) que ofrece un conjunto completo de herramientas de modelado visual. Se utiliza para crear diagramas UML, modelado de procesos de negocio (BPMN), y otros diagramas para ayudar en el diseño y la planificación de sistemas complejos. Es muy popular por sus capacidades de generar documentación y colaborar en equipo, proporcionando también herramientas para la gestión de requisitos, desarrollo ágil y gestión de bases de datos.[\[21\]](#)

4.4.3. EZestimate

EZestimate es una herramienta diseñada para la estimación de costos y planificación de proyectos. Se utiliza en la fase de inicio de proyectos para hacer estimaciones rápidas y precisas de tiempo, recursos y costos, permitiendo a los gestores tomar decisiones informadas sobre la viabilidad del proyecto y su planificación. Aunque menos conocida que otras herramientas CASE, EZestimate es valorada por su simplicidad y precisión en proyectos de software.[\[22\]](#)

4.5. Herramientas de Generación de Documentación

4.5.1. Sphinx

Sphinx es una herramienta de documentación en Python que convierte archivos en formato reStructuredText en múltiples formatos de salida, como HTML, PDF y ePub. Es ampliamente utilizada en proyectos de software, especialmente en el ecosistema de Python, para generar documentación técnica. Sphinx es capaz de integrar con otras herramientas y frameworks como ReadTheDocs, lo que facilita la creación de documentación en línea accesible. Su capacidad de manejar grandes proyectos y generar índices automáticos lo hace una opción popular entre los desarrolladores de software.[\[23\]](#)

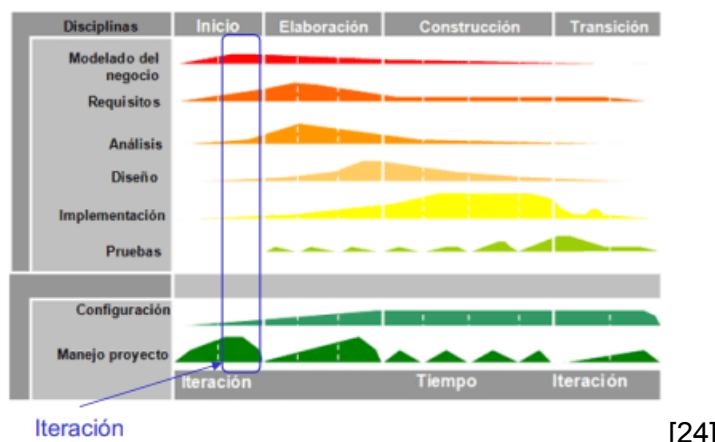
5. Aspectos Relevantes del Desarrollo

En este apartado se procederá a detallar las distintas fases del desarrollo de la plataforma.

5.1. Marco de Trabajo

La aplicación se desarrolla siguiendo el Proceso Unificado, un marco de desarrollo de software que organiza el trabajo en diferentes fases para garantizar un enfoque estructurado y metódico. Este proceso se divide en cuatro fases principales: Inicio, Elaboración, Construcción y Transición.

1. **Fase de Inicio:** En esta etapa se establecen los objetivos generales del proyecto, se definen los requisitos iniciales y se realiza una evaluación de riesgos. Es crucial asegurar que el proyecto tiene una base sólida antes de avanzar.
2. **Fase de Elaboración:** Aquí se profundiza en el análisis y diseño del sistema. Se define la arquitectura del software, se afina la estimación de recursos y tiempos, y se elaboran los planes detallados para la fase de construcción. Esta fase permite reducir la incertidumbre técnica y organizativa.
3. **Fase de Construcción:** En esta etapa se desarrolla el software de acuerdo con los requisitos y la arquitectura definidos en las fases anteriores. Es la fase donde se produce la mayor parte del código y se realizan pruebas iniciales para garantizar la funcionalidad del sistema.
4. **Fase de Transición:** Finalmente, en esta fase, se entrega el software a los usuarios finales. Incluye actividades de despliegue, capacitación y soporte inicial, asegurando que el sistema esté listo para operar en el entorno de producción.



[24]

Figura 1. Diagrama MT

5.2. Estimación de la Duración del Proyecto

Para este apartado ha sido necesario 3 puntos principales:

- Factores de complejidad del entorno
- Factores de complejidad técnica
- Dificultad de los casos de uso y de los actores

Después de asignar la complejidad a los actores, casos de uso, factores de complejidad técnica y de entorno, se procede a ingresar estos valores en la herramienta EZEstimate para calcular la estimación del esfuerzo. Para este cálculo, es necesario establecer la cantidad de horas por punto de caso de uso, que en este proyecto es de 4.

Los factores de complejidad se explican en el anexo I y los parámetros introducidos y la estimación obtenida se pueden observar en las siguientes figuras.

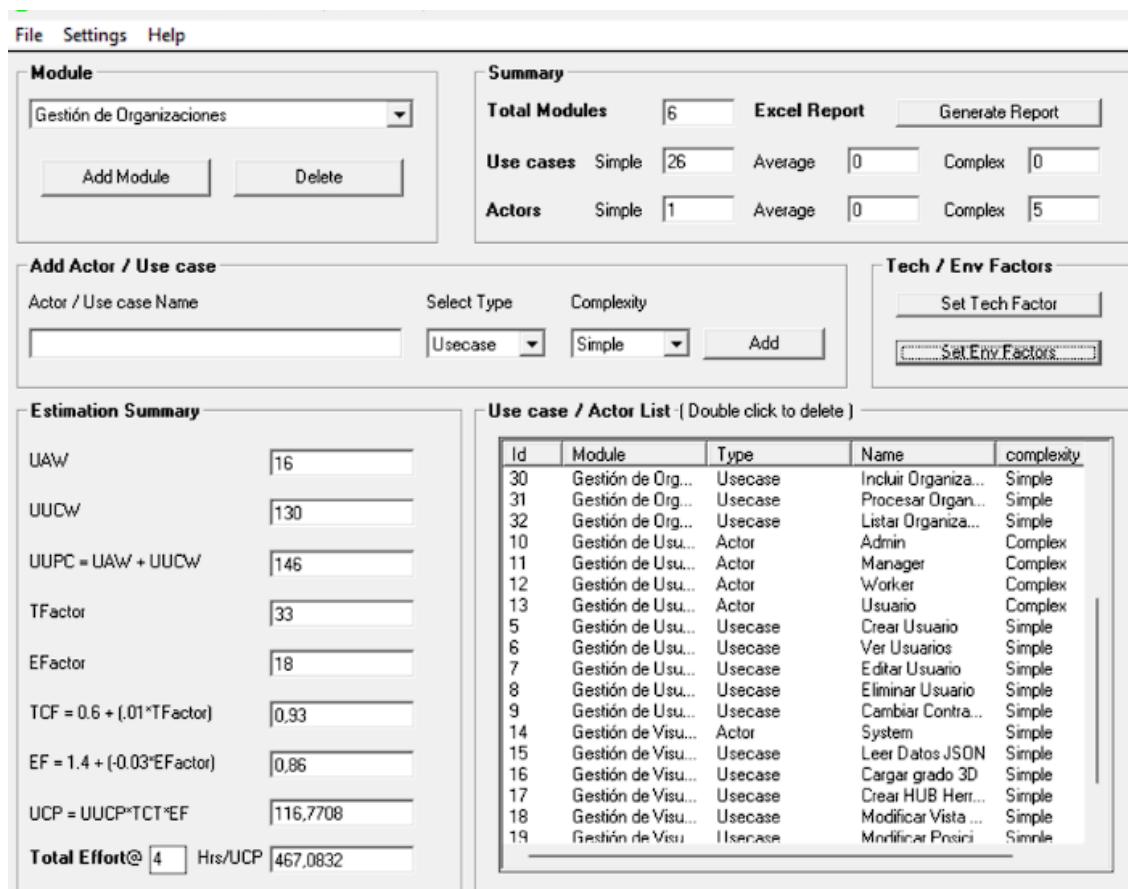


Figura 2. EZ

Set Technical Complexity	
Technical complexity factors	
Factor	Relevance
Distributed system	3
Response / Throughput performance objectives	2
End-user efficiency	1
Complex internal processing	4
Reusable code	1
Easy to install	4
Easy to use	4
Portable	3
Easy to change	2
Concurrent	3
Includes security features	2
Third party access	0
Special user training facilities required	0

Figura 3. Technical complexity

Set Environmental Factors	
Environmental factors	
Factor	Relevance
Familiar with Rational unified process	2
Application experience	0
Object oriented experience	3
Lead analyst capability	0
Motivation	5
Stable requirements	4
Part-time workers	0
Difficult programming language	1

Figura 4.Efort

5.3. Planificación Temporal

En este punto se muestra una vista de las iteraciones de Microsoft Projects con este proyecto:

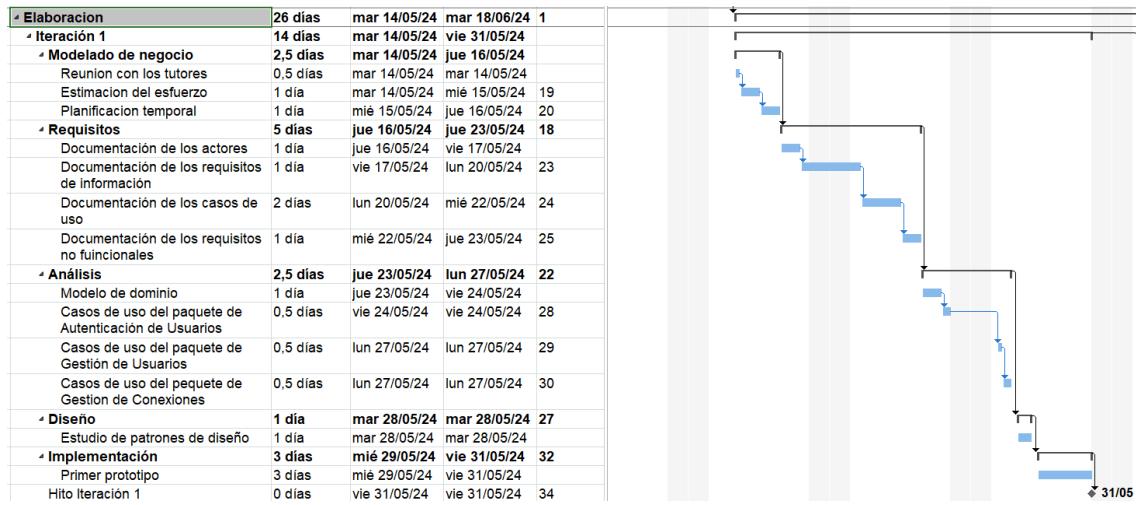


Figura 5. Projects

5.4. Especificación de Requisitos

A continuación, se incluyen ejemplos representativos de cada etapa en la especificación de dichos requisitos.

5.4.1. Actores del Sistema

Se identifican los distintos actores que interactúan con el sistema. Dado que se trata de un sistema complejo con conexión a la red, varios actores deben intervenir para garantizar la correcta gestión y funcionamiento de una aplicación web integral. Estos actores son:

Worker: Usuario encargado de visualizar sus conexiones con otros compañeros dentro de su empresa.

Admin: Responsable de gestionar los usuarios de la aplicación web y de incorporar nuevos datos en la base de datos.

Manager: Usuario que supervisa a todos los empleados y examina las conexiones en un grafo avanzado en 3D para la gestión del equipo.

<<System>>: Parte digital que actualiza la base de datos, gestiona las conexiones entre empleados, y maneja las llamadas a las APIs necesarias.

Usuario sin Login: Persona que aún no ha iniciado sesión en el sistema.

Usuario: Cualquier persona que interactúa con el sistema según el rol que desempeñe.

Ejemplo de la tabla del actor Manager:

ACT-03	Manager
Versión	1.0
Autores	YOEL PÉREZ CARRASCO
Fuentes	PracticalSII2022-2023.pdf Guia Realización.pdf
Descripción	El actor humano responsable de visualizar a todos los integrantes de la empresa y de examinar las conexiones en el grafo avanzado en 3D, facilitando así una gestión eficaz de los empleados.
Comentarios	Sin comentarios adicionales.

Tabla 1. ACT-03

5.4.2. Objetivos del Sistema

En este apartado se identifica los distintos objetivos de que deberá cumplir el sistema:

Gestión de usuarios: Implementar un sistema que permita administrar usuarios con roles diferenciados (admin, manager, worker).

Gestión de Visualización 3D: Desarrollar una funcionalidad para visualizar las relaciones entre empleados mediante grafos en 3D.

Gestión de API GitHub: Automatizar la recolección de datos de contribuciones en GitHub y almacenarlos en la base de datos.

Gestión de Autenticación: Garantizar la autenticación segura de los usuarios mediante tokens JWT para proteger el acceso a la aplicación web.

Gestión de Organizaciones: Gestionar la organización a la que pertenecen tanto los usuarios del sistema como los importados de GitHub.

Gestión de Conexiones: Gestionar y calcular las conexiones entre trabajadores de la misma empresa en la base de datos.

Ejemplo de tabla de Gestión de “API GitHub” .en el anexo II de un objetivo del sistema:

OBJ-03	Gestión de API GitHub
Versión	1.1. -> 04/08/2024
Autores	YOEL PÉREZ CARRASCO
Fuentes	PracticalSII2022-2023.pdf Guia Realización.pdf
Subobjetivos	Sin subobjetivos adicionales.
Descripción	Automatizar la recolección de datos de contribuciones en GitHub y almacenarlos en la base de datos.
Importancia	Necesario
Urgencia	Alta
Estabilidad	Con posibilidad de cambio
Comentarios	Sin comentarios adicionales.

Tabla 2. OBJ-03

5.4.3. Requisitos de Información

En este apartado se identifica los distintos requisitos de Información de que deberá cumplir el sistema:

- **Información de los managers:** El sistema almacenará información sobre los usuarios con rol de manager, incluyendo sus datos personales y de autenticación.
- **Información de los workers:** El sistema almacenará información sobre los usuarios con rol de worker, incluyendo sus datos personales y de autenticación.
- **Información de la organización:** El sistema almacenará la información relacionada con la organización a la que pertenecen los usuarios, incluyendo su nombre y datos asociados.
- **Datos JSON:** El sistema deberá convertir e interpretar la información de las conexiones del backend a formato JSON para ser utilizado en el frontend.
- **Información de la Autenticación:** El sistema gestionará la autenticación de los usuarios mediante tokens JWT para generar un contexto de rol en la aplicación web.

Ejemplo de tabla de Requisito de Información extraido del Anexo II:

IRQ-05	Información de la Autenticación	
Versión	1.0	
Autores	YOEL PÉREZ CARRASCO	
Fuentes	PracticalSII2022-2023.pdf Guia Realización.pdf	
Objetivos Asociados	OBJ-01-Gestión de Usuarios.	
Requisitos Asociados		
Descripción	El sistema procesa la autenticación del usuario que se ha logueado en la app web a través de token jwt, para generar un contexto de rol.	
Datos Específicos	1. Auth Token jwt	
Tiempo De Vida	Medio	Máximo
	4 años	10 años
Ocurrencias Simultáneas	Medio	Máximo
	100 Tuplas	1000 Tuplas
Importancia	Necesario	
Urgencia	Elevada	
Estabilidad	Sin posibilidad de cambio	
Comentarios	Se recoge toda la información asociada a los usuarios	

Tabla 3. IRQ-05

5.4.4. Requisitos No Funcionales

- **Multinavegador:** El sistema podrá ejecutarse en cualquier navegador.

- **Compatibilidad futuro con escritorio:** El sistema podrá ejecutarse tanto en navegador como en escritorio en el futuro.
-
- **Control de Errores:** El sistema deberá tener mecanismos para manejar errores tanto en navegador como en escritorio.
-
- **Cumplimiento RGPD:** El sistema deberá cumplir con el Reglamento General de Protección de Datos (RGPD) al gestionar información.
-
- **Disponibilidad:** El sistema deberá estar activo las 24 horas del día, los 7 días de la semana.
- **Requisitos mínimos del sistema:** El sistema deberá funcionar con un mínimo de 4 GB de RAM, 500 MB de disco duro y ser compatible con Microsoft Windows 98 o versiones superiores.

A continuación, se muestra una tabla de ejemplo del Requisito No Funcional “MultiNavegador”;

RNF-01	Multinavegador
Versión	1.0
Autores	YOEL PÉREZ CARRASCO
Fuentes	PracticalSII2022-2023.pdf Guia Realización.pdf
Objetivos Asociados	OBJ-01-Gestión de Usuarios.
Requisitos Asociados	RF-10- iniciar sesión
Descripción	El sistema podrá ejecutarse en cualquier navegador.
Importancia	Necesario
Urgencia	Alta
Estabilidad	Con posibilidad de cambio
Comentarios	Sin Comentarios

Tabla 4. RNF-01

5.4.5. Requisitos Funcionales

Teniendo en cuenta las distintas complejidades que pueden tomar cada uno de los actores, se estipulan tres valores: Simple, Intermedio y Complejo.

Usuario No Login: Persona que interactúa con la interfaz gráfica sin iniciar sesión.

Worker: Persona que interactúa con la interfaz gráfica como trabajador del sistema.

Admin: Persona que interactúa con la interfaz gráfica con permisos administrativos.

Manager: Persona que interactúa con la interfaz gráfica con funciones de gestión.

Usuario: Persona que interactúa con la interfaz gráfica como un usuario registrado.

<<System>>: Un sistema que se comunica con la aplicación a través de una API.

A continuación se muestra un ejemplo de la tabla de actores para Admin:

ACT - 003	Admin
Complejidad	Complejo
Comentarios	Persona que interacciona con interfaz gráfica

Tabla 5. ACT-03

La complejidad de los casos de uso se clasifica según el número de transacciones que involucran: hasta 3 se consideran simples, entre 4 y 7 son intermedios, y más de 7 son complejos. Este método facilita estimar el esfuerzo necesario para desarrollar cada caso de uso basado en el nivel de interacción.

Caso de Uso
UC-001 Registrar Usuario
UC-002 Logout
UC-003 Login Usuario
UC-004 Cambiar Contraseña
UC-005 Crear Usuario
UC-006 Ver Usuarios
UC-007 Editar Usuario
UC-008 Eliminar Usuario
UC-009 Leer Datos JSON
UC-010 Cargar grafo 3D
UC-011 Crear hub Herramientas
UC-012 Modificar Vista Visualización 3D
UC-013 Modificar Posicionamiento Cámara
UC-014 Ver ayuda
UC-015 Ver Gráfico De Empresa
UC-016 Autenticar API

UC-017 Obtener Datos API
UC-018 Procesar datos Github
UC-019 Ver Conexiones
UC-020 Exportar CSV Conexiones
UC-021 Generar CSV Conexiones
UC-022 Realizar Conexiones
UC-023 Procesar Conexiones
UC-024 Incluir Organización
UC-025 Procesar Organizaciones de Usuarios Github
UC-026 Listar Organizaciones

Tabla 6. Tabla de UC

Se muestra a continuación un ejemplo de una de las tablas de la lista, concretamente la del caso de uso Listar Organizaciones, cuyo número es el 26:

UC-26	Listar Organizaciones	
Versión	v1.0	
Autores	YOEL PÉREZ CARRASCO	
Fuentes	PracticalSII2022-2023.pdf Guia Realización.pdf	
Dependencias	Organizaciones disponibles	
Descripción	El sistema permitirá listar las organizaciones disponibles en la plataforma.	
Precondición	El usuario debe estar autenticado.	
Secuencia Normal	Paso	
	1	El usuario solicita la lista de organizaciones.
	2	El sistema muestra las organizaciones registradas.
Postcondición	La lista de organizaciones se muestra correctamente.	
Excepciones	Paso	
	1	Error al listar organizaciones: mostrar mensaje de error.
Rendimiento	Paso	
	-	La lista de organizaciones debe cargarse en menos de 3 segundos.
Frecuencia	Frecuente según las necesidades de gestión de organizaciones.	

Esperada	
Importancia	Media
Urgencia	Media
Estabilidad	PD
Comentarios	Sin comentarios

Tabla 7. UC-26

Seguidamente, un ejemplo del diagrama del paquete de visualización 3D

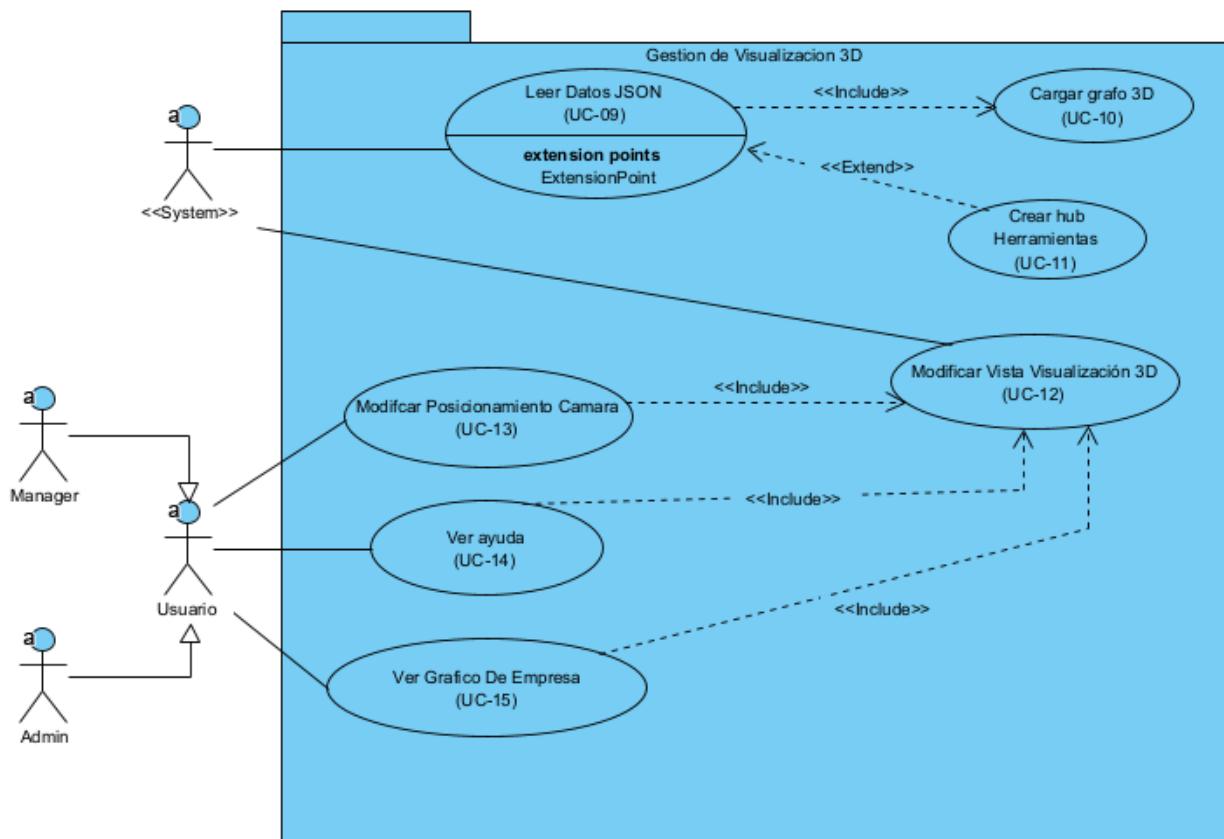


Figura 6. Paquete 3D

Finalmente se muestra el diagrama de paquetes:

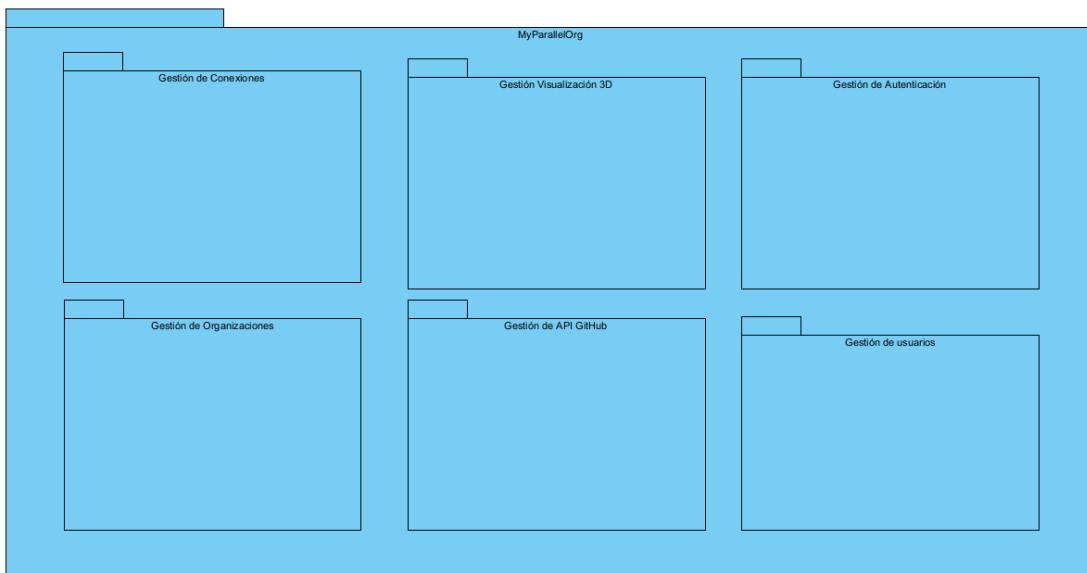


Figura 7. diagrama de paquetes

5.5. Diseño del Sistema

5.5.1. Patrón Modelo-Vista-ViewModel (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]

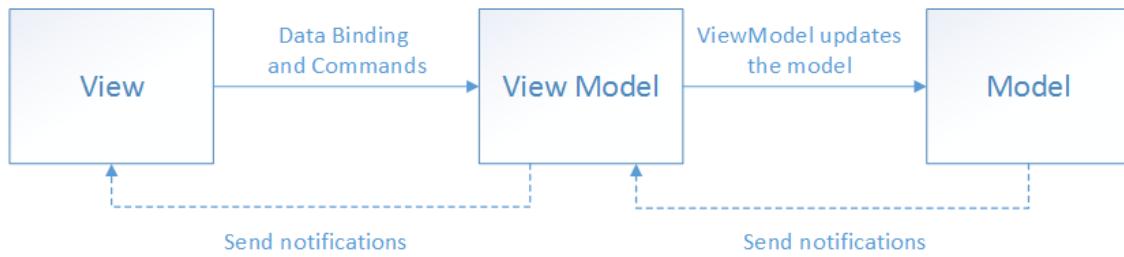


Figura 8. Modelo - Vista - Vista/Modelo

5.5.2. Patrón DAO

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.

El patrón DAO se organiza en:

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

DataAccessObject: Abstacta 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]

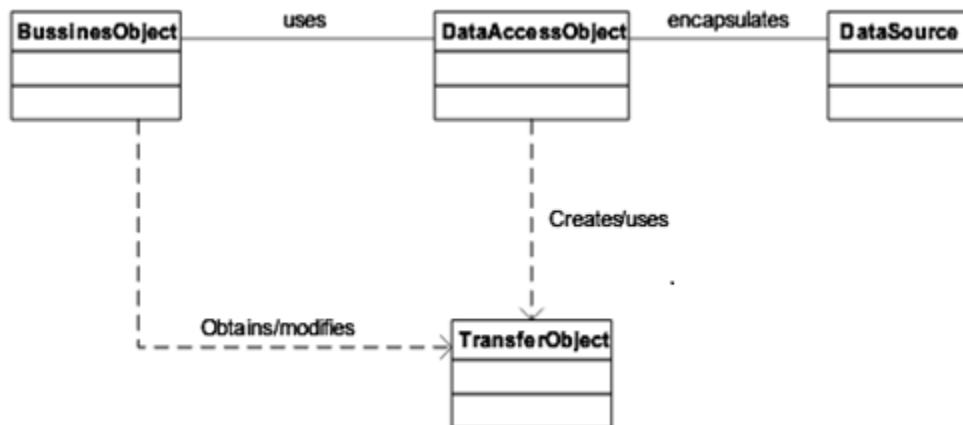


Figura 9: Patrón DAO

5.5.3. Subsistemas de Diseño

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

- **Back-End:** Este subsistema contiene la lógica y la gestión de los datos del sistema. Los datos se almacenan en el paquete Model, representa la estructura y las relaciones de los datos. La especificación del patrón DAO se encuentra en el paquete Back-EndDAO, 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:

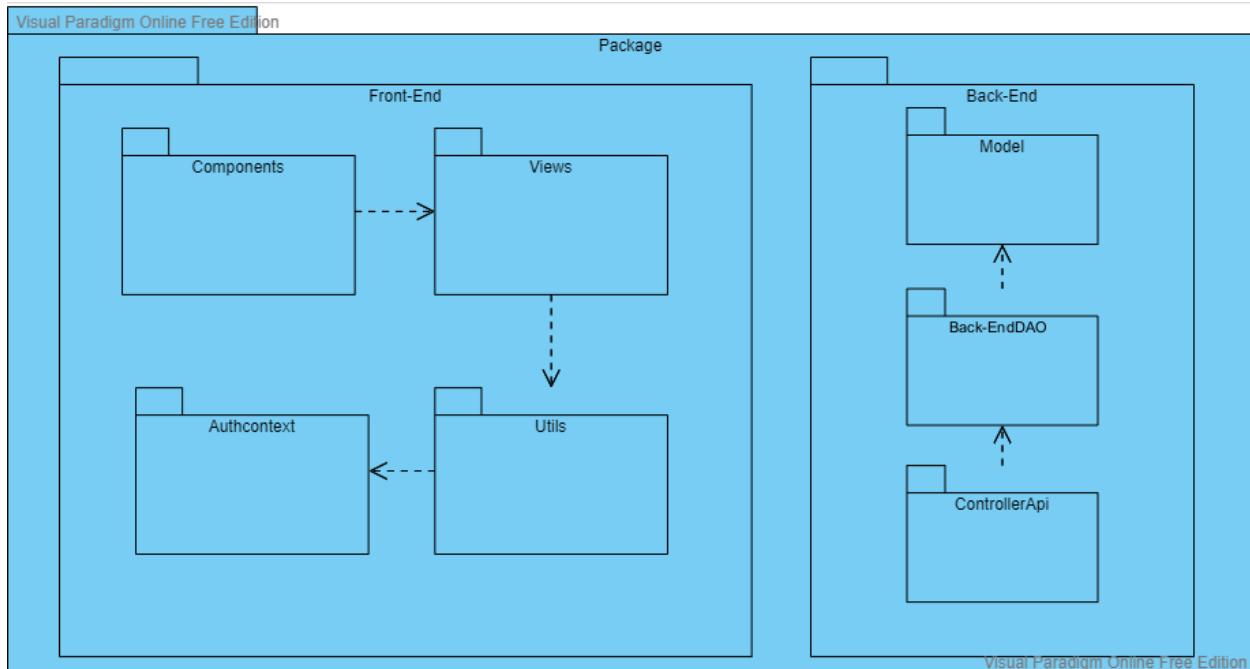


Figura 10. Subsistemas de Diseño

5.5.4. Clases de Diseño

En este apartado se especifican las clases de diseño del sistema.

El Back-End presenta un subsistema controlador, un Back-EndDAO y un modelo. Sus clases de diseño se observan en las siguientes figuras.

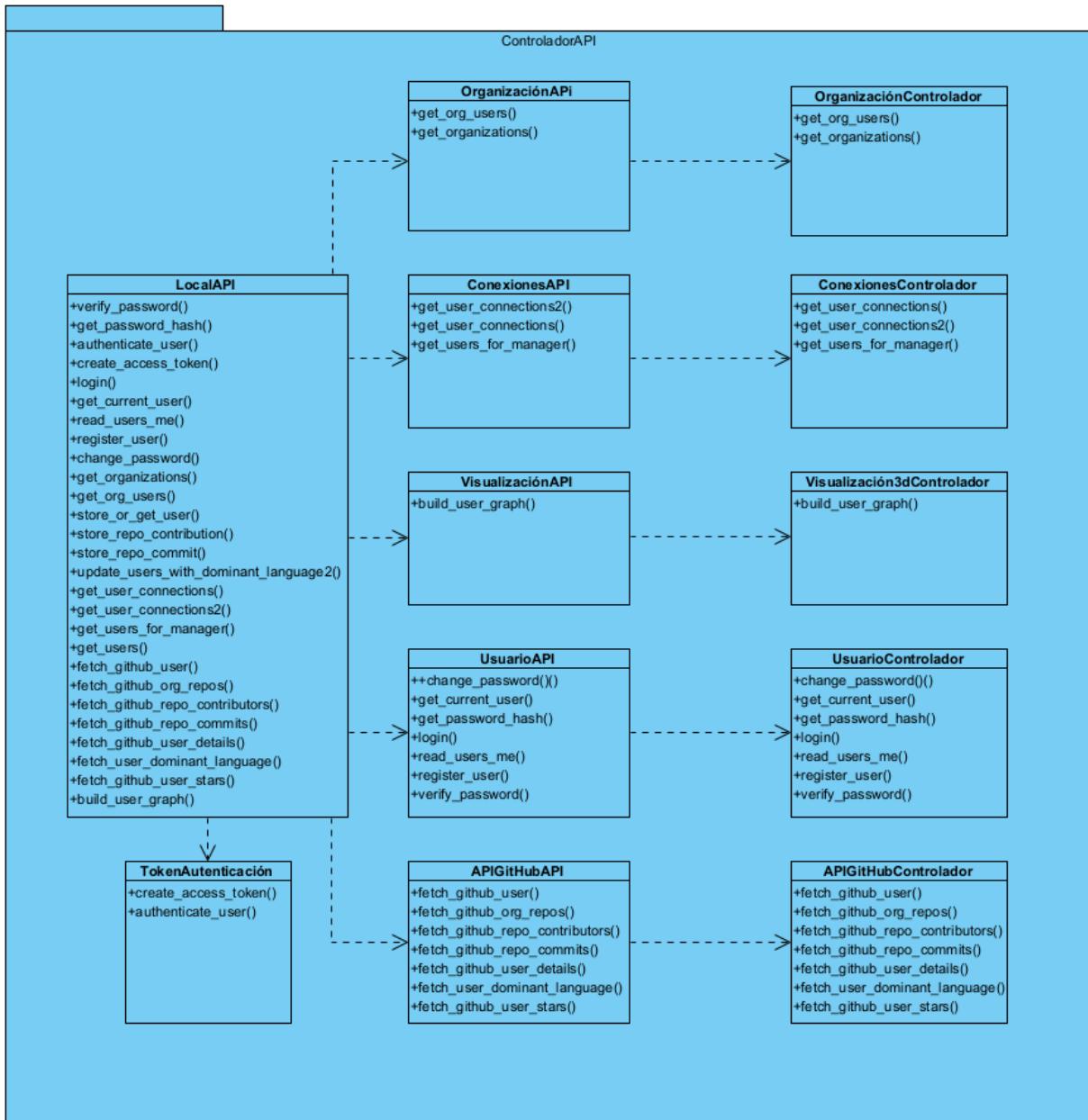


Figura 11.subsistema controlador

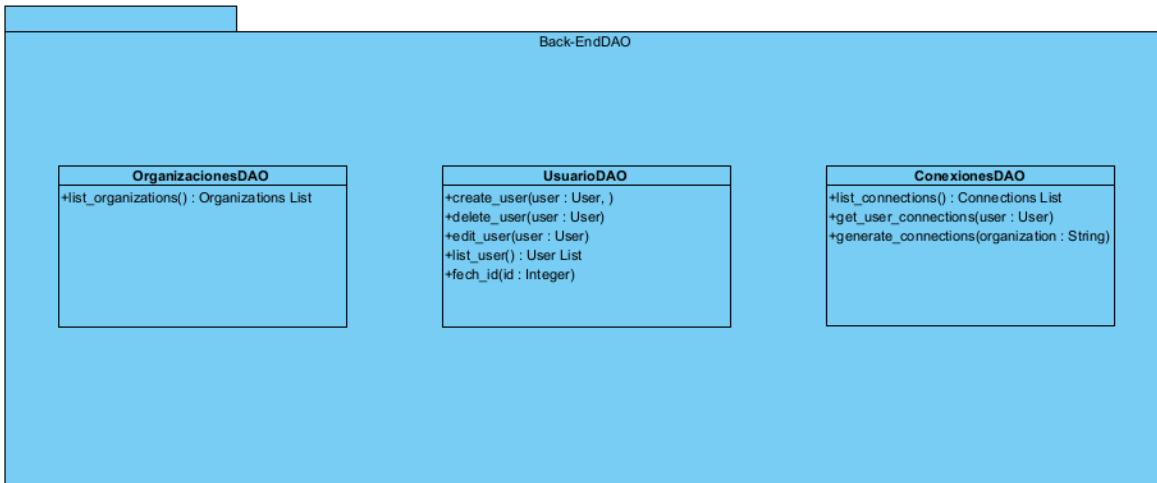


Figura 12.subsistema Back-EndDAO

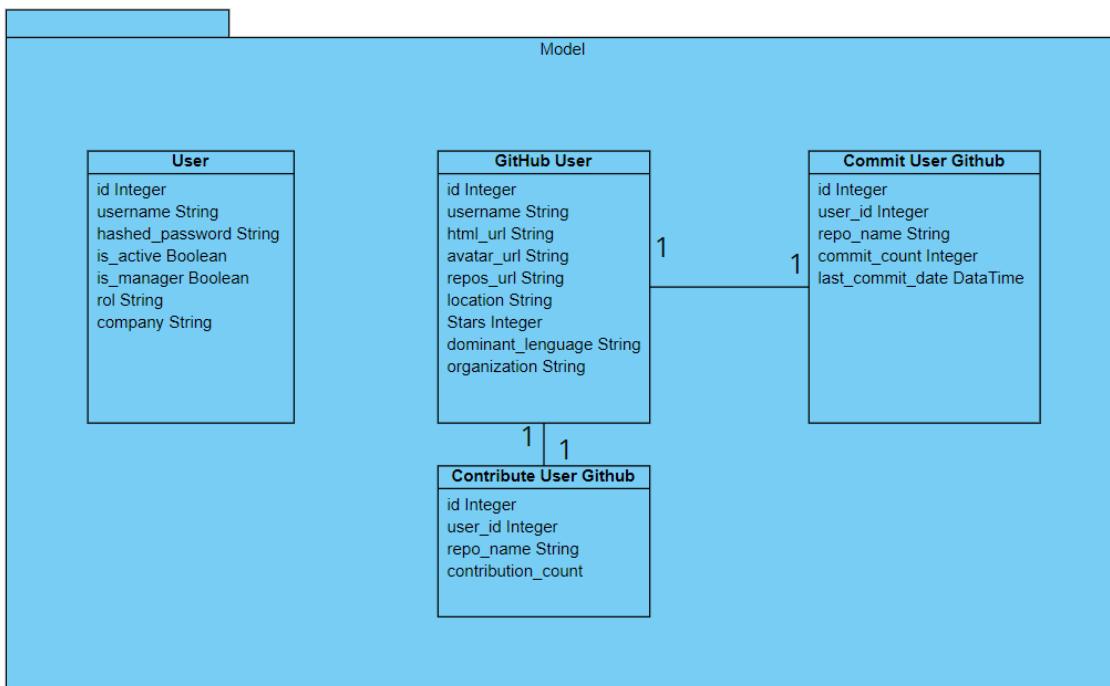


Figura 13.subsistema model

Respecto al Front-End, este se divide en los subsistemas de components, views, authcontext y utils. Sus clases de diseño son:

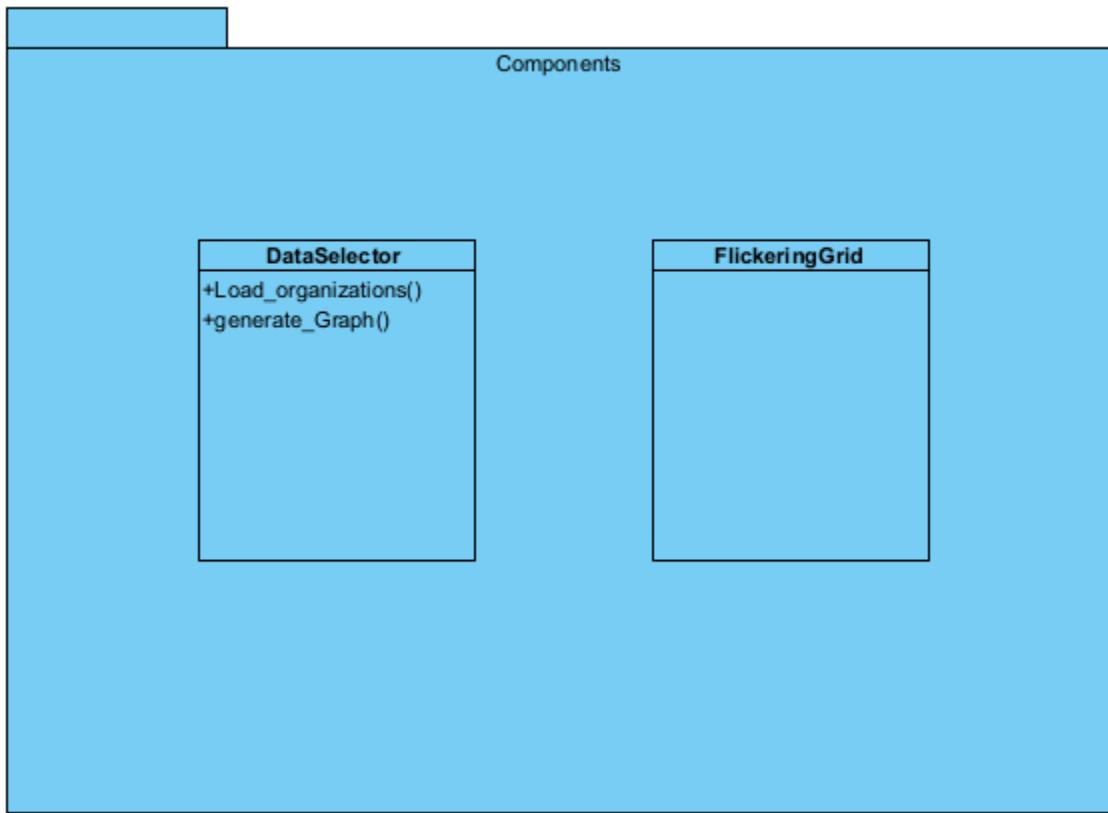


Figura 14.subsistema vista components

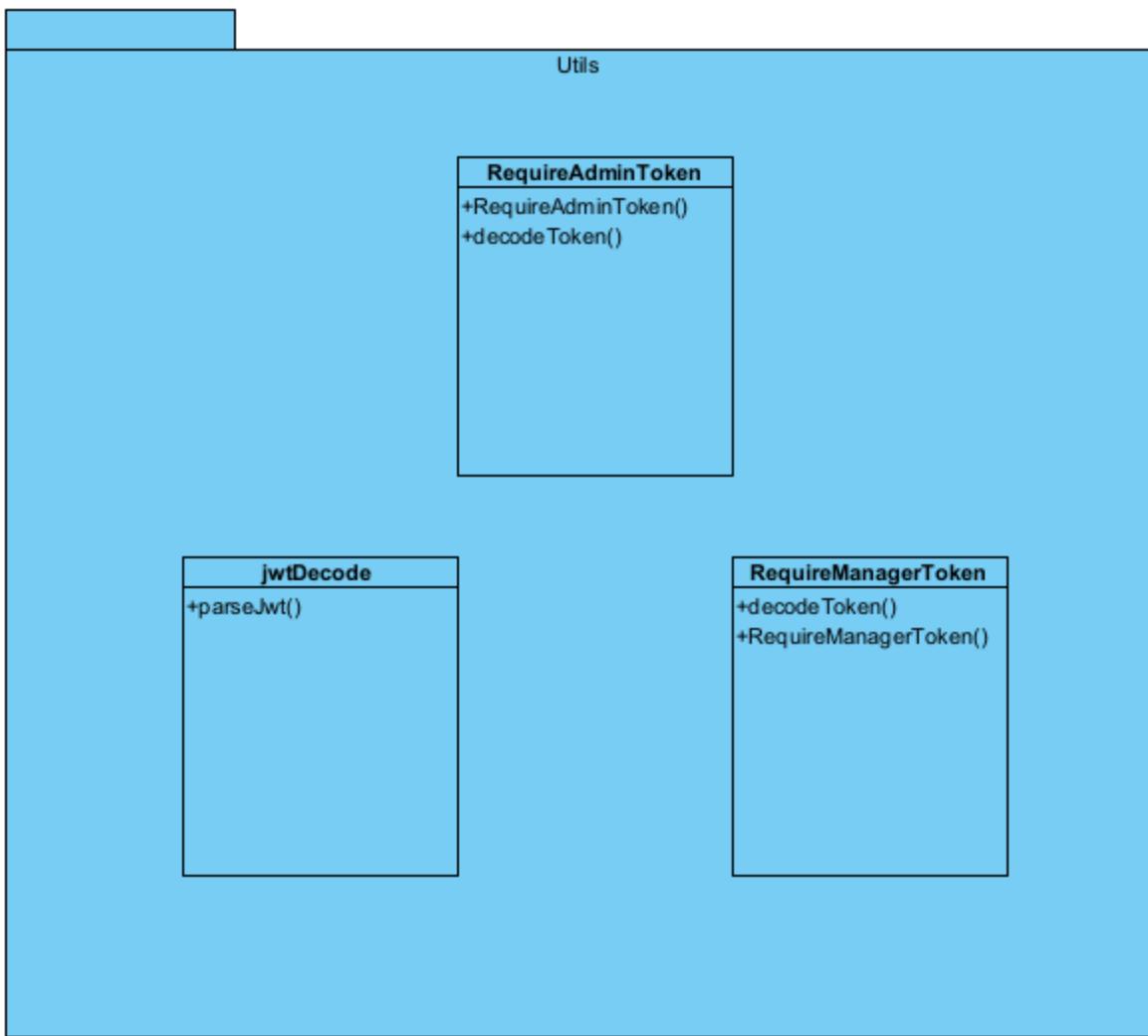


Figura 15.subsistema vista utils

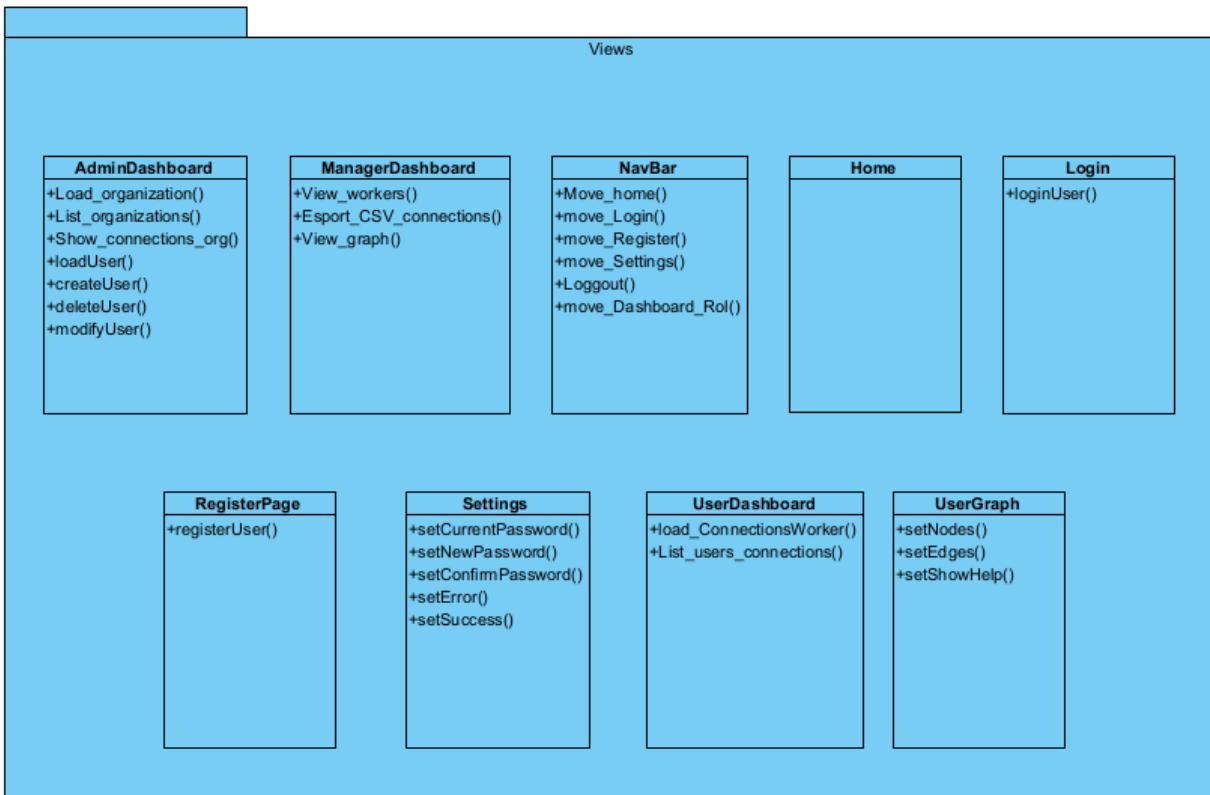


Figura 16.subsistema vista Views

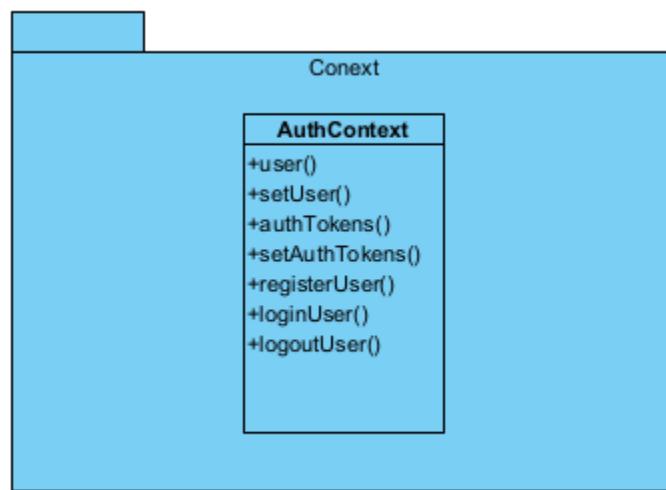


Figura 17.subsistema vista Context

5.5.5. Realización de Casos de Uso (Diseño)

En este apartado se muestran las interacciones de la plataforma y los mensajes de la misma, de una forma más concreta y cercana al producto final. Se adjuntan 3 ejemplos de diagramas, uno de secuencia y 2 de estado:

- **Listar Organizaciones**
- **Iniciar Sesión**
- **Cerrar Sesión**

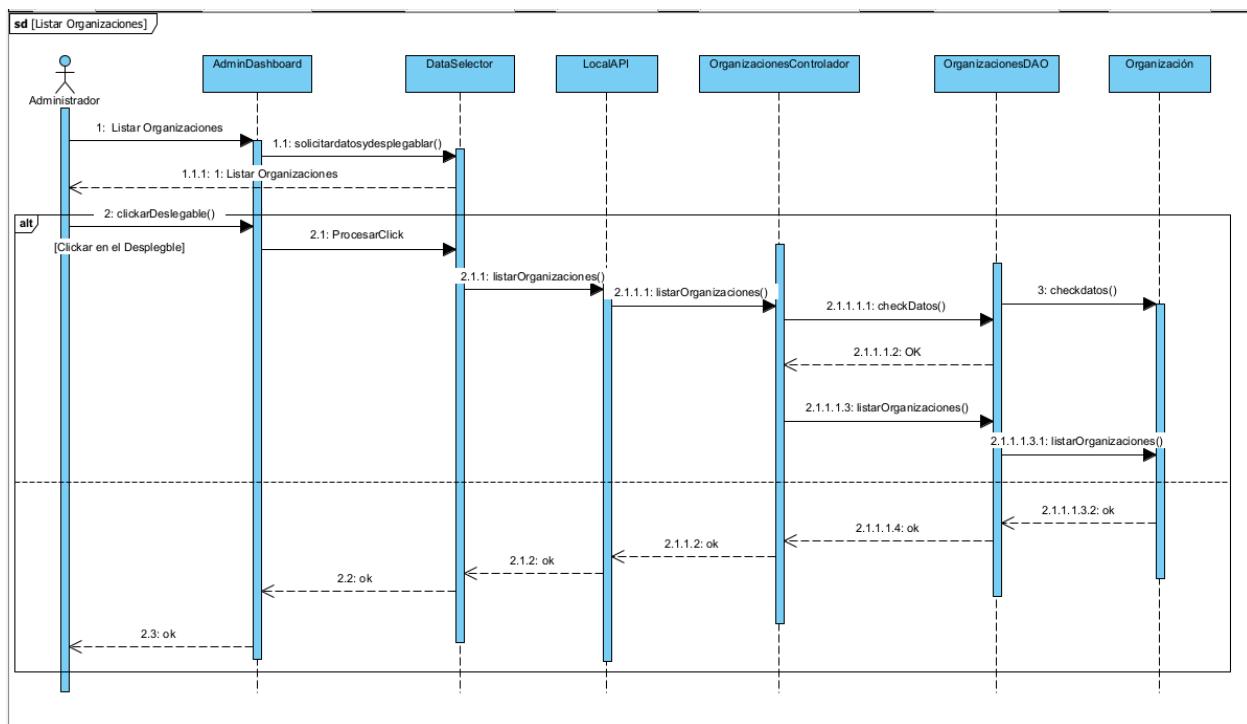


Figura 18. Diagrama de Secuencia

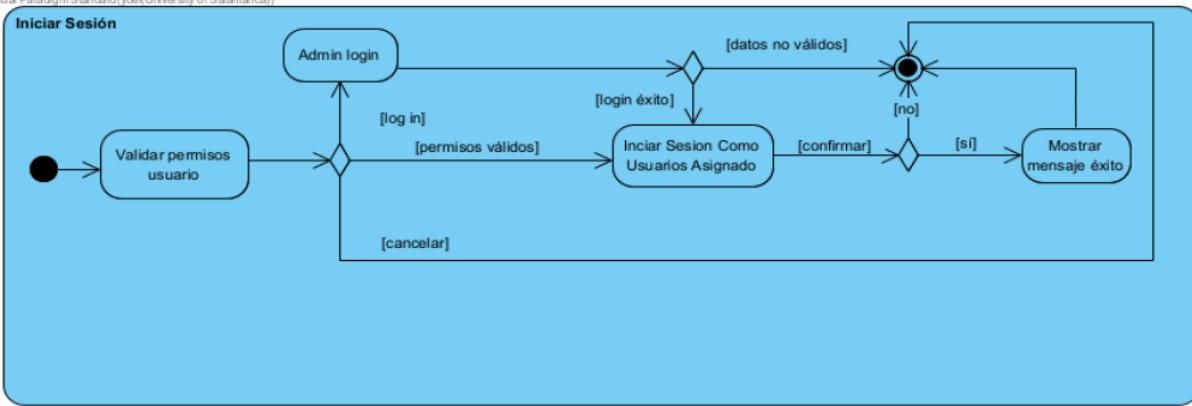


Figura 19. Diagrama de estados I

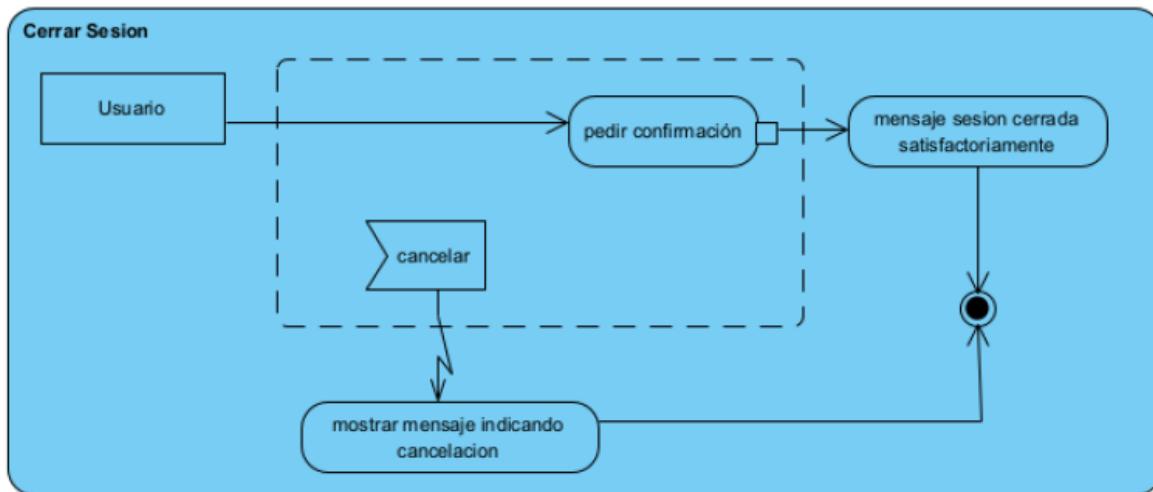


Figura 20. Diagrama de estados II

5.5.6. Diseño de la Base de Datos

En este apartado se muestra las tablas de las bases de datos junto a las relaciones que tienen entre ellas:

Diagrama Diseño Base De Datos

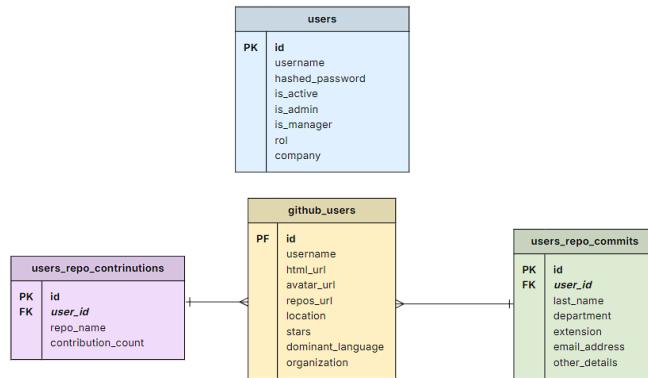


Figura 21. Diseño de la Base de Datos

5.5.7. Modelo de Despliegue

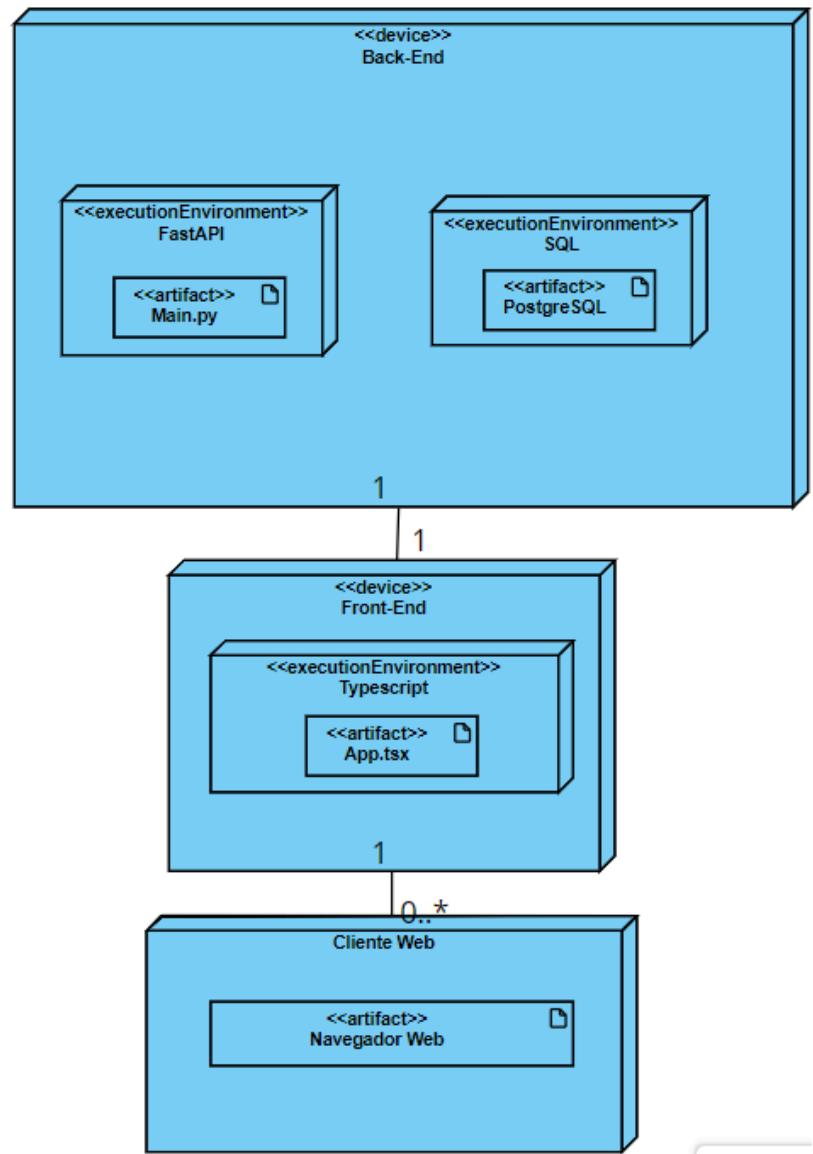


Figura 22. Diseño de la Base de Datos

5.6. Implementación

Durante esta fase, se interpretaran todos los datos recogidos en la fase de diseño. Para llevar a cabo esta etapa, se emplean las técnicas y herramientas descritas en el [apartado de técnicas y herramientas del sistema](#) del documento. La implementación se organiza en tres componentes principales:

Front-End: En esta parte el front-end se encargará de gestionar toda la vista que se le presenta al usuario, así como toda la funcionalidad de conexión e interpretación de datos con el back-end a traves del protocolo HTTP

Back-End: La parte de la implementación más importante del back-end radicara en la instalación de la API loca, como de las llamadas a la API de Github para la recolección de datos.

5.7. Pruebas

Las pruebas son una parte fundamental dentro del ciclo de vida software puesto que en esta se pondrá a prueba la completa funcionalidad deseada, y si no fuera así habría que volver a generar otra iteración en producción o en el peor de los casos, volver a etapas de análisis.

Se han realizado pruebas tanto completas con el sistema montado al completo, como pruebas unitarias para así verificar su correcta integración.

La prueba más importante realizada en este sistema, se trata de la carga de una empresa con muchos repositorios, esta empresa es “facebookexperimental”, la cuál cuenta con más de 500 supuestos trabajadores. Al ser una cantidad de trabajadores tan grande se ha puesto a prueba la optimización de recursos del sistema, como así el tratamiento de datos por parte de ReaGraph[25].

A continuación, se mostrará una captura de la visualización 3D, donde se observa con nitidez las buena carga de datos y sus conexiones pertinentes:



Figura 22. Prueba 3D

5.8. Funcionalidad del Sistema

En esta sección se presenta un recorrido a través de la aplicación web del sistema propuesto, con el objetivo de facilitar y familiarizar al usuario con su uso. Se detallan las principales funcionalidades que el sistema pone a disposición del usuario, permitiendo que cada rol acceda a diferentes opciones según sus permisos. El sistema contempla cinco roles, por lo que esta sección está organizada de acuerdo con estos roles:

Usuario No Logueado: Puede explorar funciones básicas de la aplicación, sin acceso a funcionalidades avanzadas.

Worker: Accede a tareas específicas y colaborativas, como visualizar sus propias conexiones y contribuciones dentro de la organización.

Manager: Tiene la capacidad de gestionar grupos de trabajo, ver y organizar conexiones entre empleados, así como supervisar la actividad del equipo.

Usuario: Posee acceso general a las funcionalidades estándar de la aplicación, como la visualización de datos y participación en proyectos.

Administrador: Administra la totalidad del sistema, gestionando roles, permisos, y la configuración global de la aplicación.

Para obtener más información detallada sobre cada funcionalidad, consulte el **Anexo V: Manual de Usuario**.

5.8.1. Instalación y Configuración

La instalación y configuración se encuentran en la parte de la documentación técnica Anexo IV de esta documentación, en este caso se optó por utilizar un aplicación de formateo y conversión de documentación de códigos fuente Sphinx, por lo se podra encontrar publicada en Read The Docs en este enlace:

- [Enlace Instalación del entorno APP](#)

5.8.2. Inicio

Esta pantalla incluye un botón denominado "Comenzar", que dirige al usuario al formulario de inicio de sesión:



Figura 23. Vista Inicio Bienvenida

La sección de "Visualización de redes" incluye un botón "Ver detalles", que permite al usuario acceder a información detallada sobre esta característica. Asimismo, se ofrece acceso a la documentación de ReaGraph, que es la herramienta utilizada para la visualización de redes, y a la documentación sobre el despliegue con Docker, donde se explica el proceso de implementación:



Figura 24. Vista Inicio Utilidades y Referencias

5.8.3. Autenticación y Registro

En este apartado se comentarán las distintas formas que tendrá el usuario sin loguearse todavía para poder acceder a la plataforma.

5.8.3.1. Registro de Usuarios

Este proceso se lleva a cabo mediante un formulario que incluye cuatro campos esenciales: rol, nombre de usuario, contraseña y organización. Para completar el registro, es necesario ingresar la información correspondiente en cada uno de estos campos y, posteriormente, hacer clic en el botón "Registrar".

El sistema almacenará los datos y creará la cuenta de usuario. Es importante destacar que el rol seleccionado en esta etapa definirá las vistas y funcionalidades disponibles para el usuario dentro de la aplicación:

The screenshot shows a registration form titled "Registrarse". On the left, there is a large, dense network graph consisting of numerous small circles connected by thin blue lines. To the right of the graph, the form fields are arranged vertically:

- "Elige tu rol" (Choose your role) with a dropdown menu labeled "Rol".
- A text input field labeled "Usuario".
- A text input field labeled "Contraseña".
- A text input field labeled "Organización".

Below the fields is a black rectangular button labeled "Registrarse". At the bottom of the form, there is a link "¿Ya tienes una cuenta? [Iniciar Sesión](#)". At the very bottom, there are two small links: "Terms of use" and "Privacy policy".

Figura 25. Vista Registro

En este último campo “Organización”, se deberá ajustar con exactitud en nombre de la organización con el se haya registrado dicha organización en la plataforma de Github:

Organización

Registrarse

Figura 26. Vista Registro Organización

5.8.3.2. Login de Usuarios

Para acceder a la plataforma, se presenta un formulario que solicita el nombre de usuario y la contraseña, junto con un botón de "Iniciar sesión". Al ingresar las credenciales registradas previamente, el sistema redirige al usuario al panel de control (dashboard), donde se mostrarán las opciones y funcionalidades disponibles según el rol asignado al usuario en la plataforma:



Figura 27. Vista Login

A esta vista podremos acceder mediante el botón de “Iniciar Sesión” anclado en el NavBar.

5.8.4. Vista del Worker

Cuando un usuario con el rol de **Worker** inicia sesión en MyParallelOrg y accede a la sección de "Tus Conexiones en la Empresa", es posible que se encuentre con diferentes escenarios dependiendo de su estatus dentro de la base de datos.

Si la organización de la persona que ha accedido a la plataforma está incluido en la base de datos y contiene conexiones con otros trabajadores podrá ver algo así:

The screenshot displays a dashboard titled "Tus Conexiones en la Empresa". It features a 3x3 grid of profile cards. Each card contains a user icon, a name, and a "View Profile" button. The names and icons are as follows:

- Row 1: disconnect3d (blue circular icon), dgrnbrg-meta (green checkered icon), pbacer (brown circular icon)
- Row 2: mshawcroft (person at desk icon), psaab (two people icon), mauriciolempke (purple plus icon)
- Row 3: davejiang (orange person icon), vitorfalcao (pink plus icon)

Figura 28. Vista Dashboard de Worker

De lo contrario, si el usuario no tiene ninguna conexión registrada en la base de datos, verá un mensaje en la pantalla que dice: "Ohhhh vaya..., parece que no tienes todavía ningún compañero por aquí..." .

Este mensaje indica que el usuario no está vinculado a ningún otro miembro de la organización en la red colaborativa:

The screenshot shows a dashboard titled "Tus Conexiones en la Empresa". Below the title, there is a single line of text: "Ohhhh vaya..., parece que no tienes todavía ningun compañero por aqui..." .

Figura 29. Vista Dashboard de Worker sin conexiones

5.8.5. Vista del Manager

La interfaz del Manager dentro de la aplicación permite gestionar y visualizar las conexiones de empleados y detalles individuales de cada trabajador:

The screenshot shows the Manager dashboard interface. At the top, there is a header bar with the title "MyParallelOrg" on the left and navigation links "Inicio", "Dashboard", and "qwe" on the right. Below the header, the main content area is titled "Conexiones de Empleados". It features a section titled "Trabajadores de tu Empresa" containing six employee profiles arranged in a 2x3 grid. Each profile card includes a circular profile picture, the employee's GitHub handle, their programming language, their GitHub star count, and a "Ver Perfil de Github" button.

Employee Handle	Lenguaje	Reputación GitHub (Stars)	Action
adam-yang	C++	1	Ver Perfil de Github
llvm-beanz	C++	9	Ver Perfil de Github
tristanlabelle	C++	43	Ver Perfil de Github
jaebaek	C++	14	Ver Perfil de Github
jeffnn	C++	0	Ver Perfil de Github
marcelorl	JavaScript	0	Ver Perfil de Github

Figura 30. Vista Dashboard de Manager

En la parte superior se presenta una sección titulada "Conexiones de Empleados", que ofrece opciones para ver los grupos de trabajo y exportar las conexiones en formato CSV :

Conexiones de Empleados

[Ver Grupos De Trabajo](#)[Exportar CSV](#)

Figura 31. Vista Dashboard de Manager Utilidades

Debajo de esta sección se muestra una lista de trabajadores de la empresa, con información clave como la foto de perfil, el nombre de usuario, el lenguaje de programación principal y la reputación en GitHub medida en estrellas.

Cada tarjeta de empleado incluye un botón para ver el perfil de GitHub, lo que permite acceder directamente al perfil de GitHub del trabajador seleccionado:

Trabajadores de tu Empresa

 adam-yang Lenguaje: C++ Reputación GitHub (Stars): 1 Ver Perfil de Github	 llvm-beanz Lenguaje: C++ Reputación GitHub (Stars): 9 Ver Perfil de Github	 tristanlabelle Lenguaje: C++ Reputación GitHub (Stars): 43 Ver Perfil de Github
 jaebaek Lenguaje: C++ Reputación GitHub (Stars): 14 Ver Perfil de Github	 jeffnn Lenguaje: C++ Reputación GitHub (Stars): 0 Ver Perfil de Github	 marcelolr Lenguaje: JavaScript Reputación GitHub (Stars): 0 Ver Perfil de Github

Figura 32. Vista Dashboard de Manager Trabajadores

5.8.6. Barra de Navegación (Navbar)

Un componente crucial en la plataforma es el NavBar, el cuál acompañará al usuario durante toda la interacción con la plataforma.

Esta utilidad será clave para la navegación y orientación por la diferentes vistas de la plataforma, así contando con diferentes formas de visualización en torno si el usuario ha sido logueado ya o no en la plataforma.



Figura 33. Vista NavBar

5.8.6.1. Logotipo y Enlaces Principales

En la parte izquierda del NavBar se encuentre el nombre corporativo de la plataforma junto con el logotipo elegido, el cuál es interactivo para poder volver a la visualización de inicio:

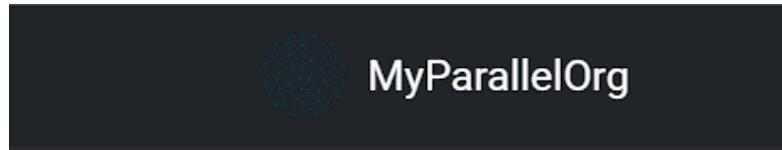


Figura 34. Vista NavBar

Estado de la vista del NavBar para un usuario todavía no autenticado dentro de la plataforma:

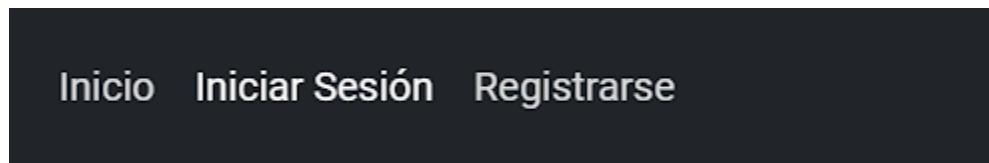


Figura 35. Vista NavBar No Login

Estado de la vista del NavBar para un usuario autenticado dentro de la plataforma, donde se podrá observar el nombre de usuario de la persona en cuestión:

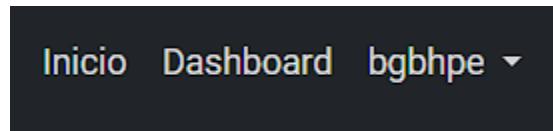


Figura 36. Vista NavBar Login

5.8.6.2. Opciones de Usuario (Ajustes y Cerrar Sesión)

Dentro del NavBar, en el apartado del usuario logueado podremos desplegar un hub de opciones personalizadas para cada usuario, llamados Ajustes y cerrar sesión

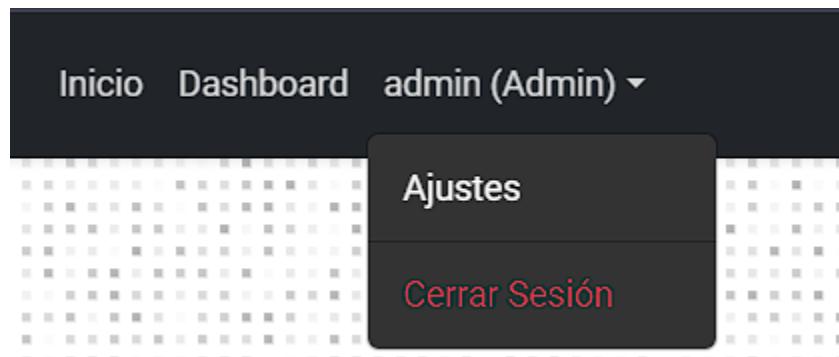


Figura 37. Vista NavBar Desplegable

Estos botones adheridos a un desplegable en el NavBar, realizar **dos acciones**:

Acceder a la pantalla de Ajustes donde podremos ver las estadísticas del consumo de medios de la APP y también podremos cambiar la contraseña del usuario actual logueado.

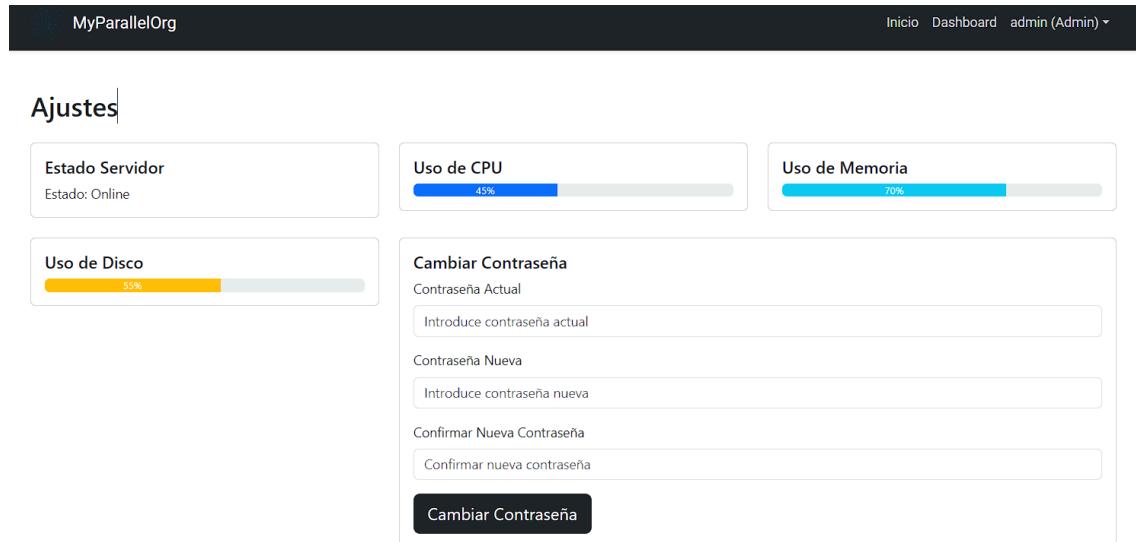


Figura 38. Vista Ajustes

Cerrar sesión del sitio eliminando el token de autenticación que hacía que nuestra sesión siguiese abierta durante un tiempo determinado:

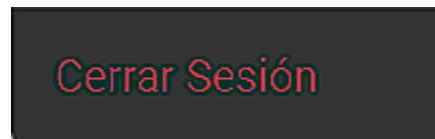


Figura 39. Vista Cerrar sesión

5.8.7. Visualización Gráfica 3D

El pilar fundamental de la plataforma se centra en el procesamiento y visualización de los datos clusterizados relacionados con las conexiones entre los trabajadores de la empresa.

Al acceder a esta visualización, ya sea como administrador o manager, se muestran los nodos, que representan a los trabajadores, conectados entre sí en función de las afinidades generadas entre ellos. Estas conexiones permiten identificar y crear grupos de trabajo más eficientes, facilitando una gestión óptima de los equipos dentro de la organización:

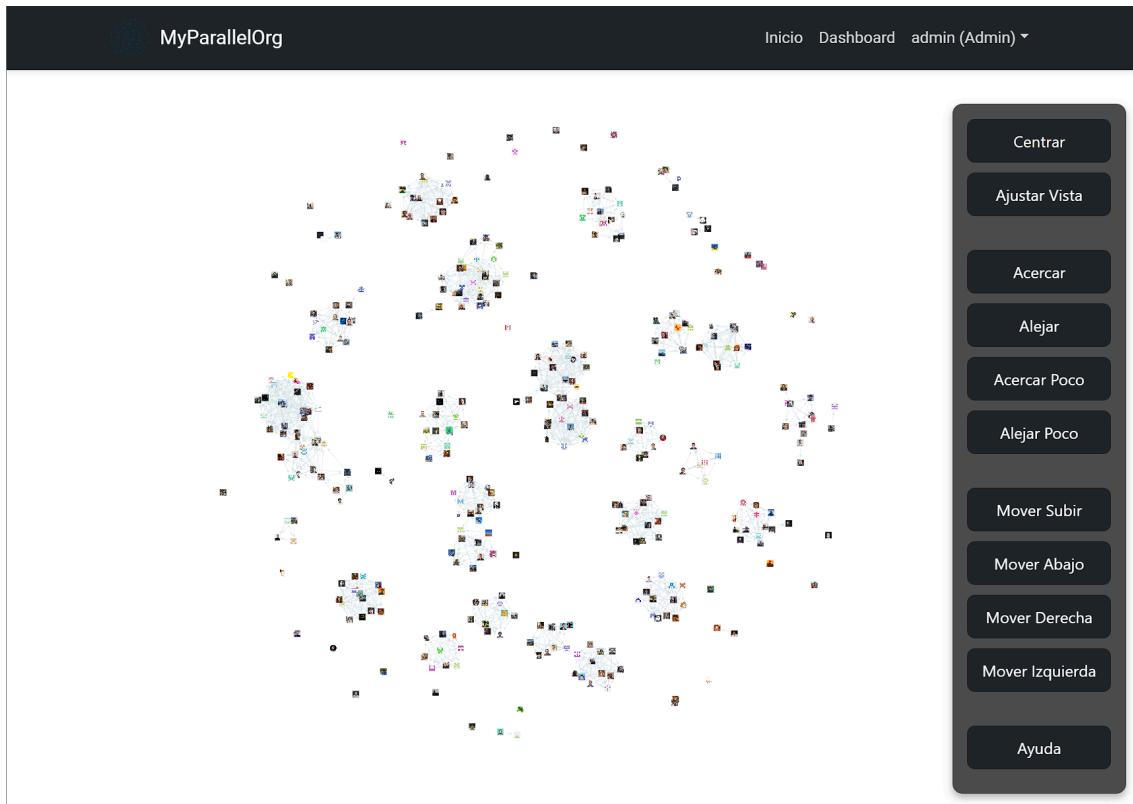


Figura 40. Vista Visualización 3D

5.8.7.1. Nodos y Conexiones

Al acercarnos a las agrupaciones creadas en la visualización, podremos observar con mayor detalle cómo se han formado los grupos de trabajo eficientes y quiénes los integran. Los nodos en la visualización representan a cada trabajador y cuentan con especificaciones como el avatar utilizado en su perfil de GitHub, así como su nombre de usuario. Además, la plataforma permite interactuar directamente con los nodos; al hacerlo, el sistema redirige al perfil de GitHub del trabajador correspondiente, ofreciendo una integración fluida entre la visualización y el análisis de las contribuciones del empleado en su repositorio.

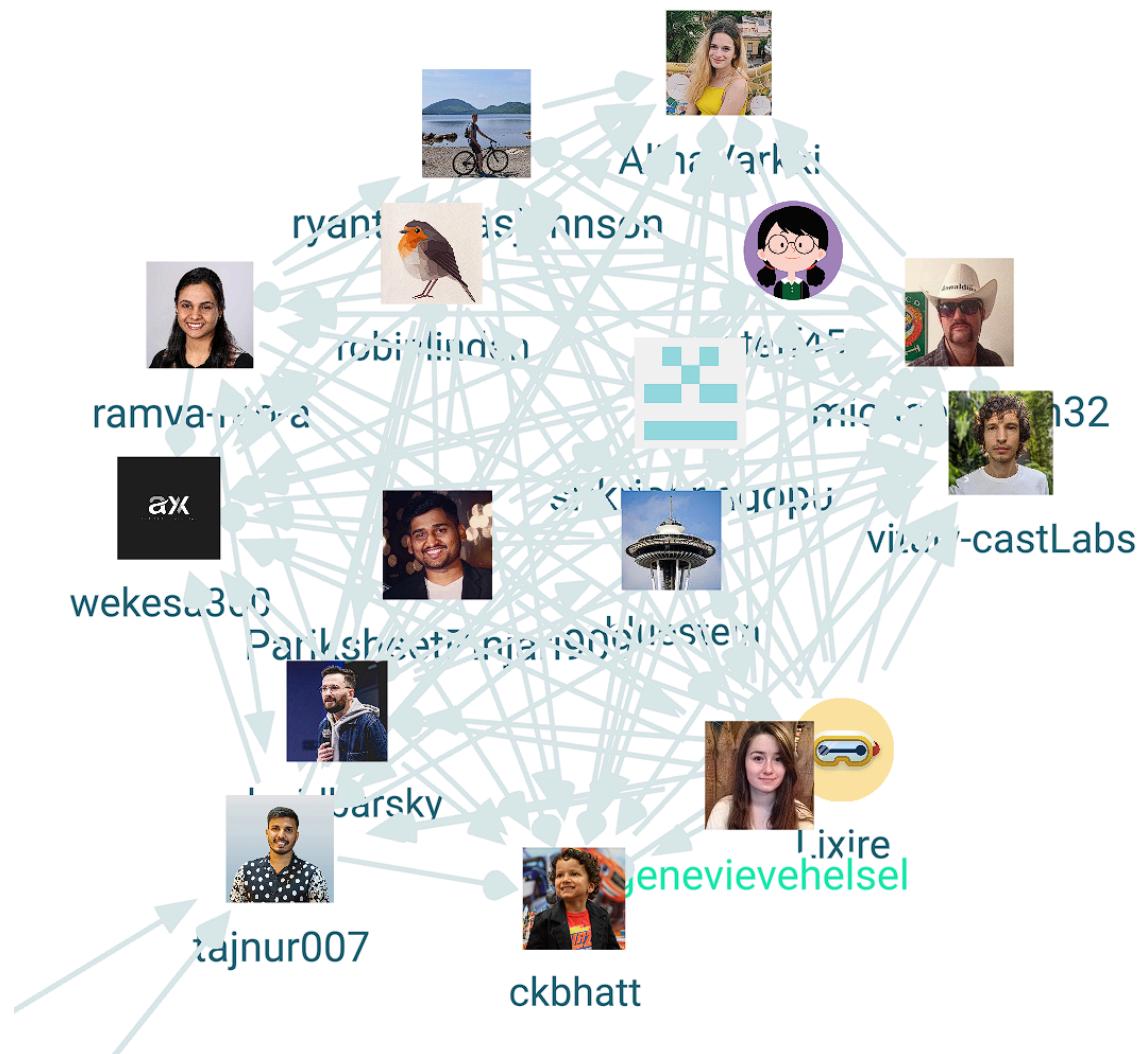


Figura 41. Vista Nodos

5.8.7.2. Controles de Navegación

El hub de configuración de posicionamiento de cámara dentro de la visualización 3D del grafo de conexiones permite al usuario controlar la posición y el enfoque de la cámara en relación con el grafo que se está visualizando:

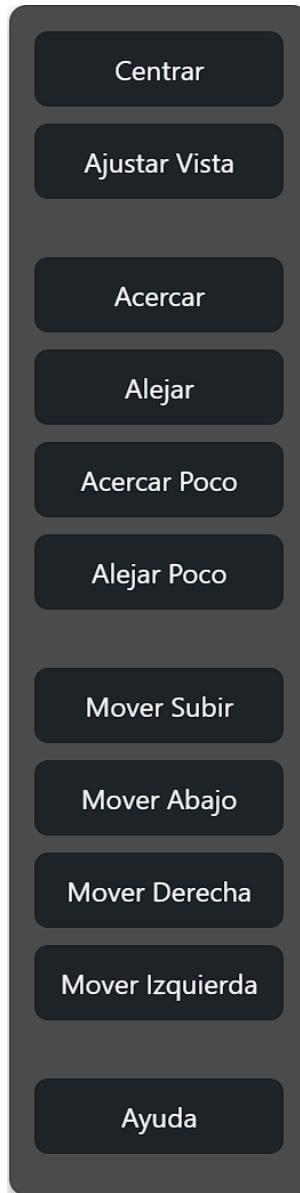


Figura 42. Vista Hub Herramientas Visualización

Centrar: Restablece la cámara para centrar la visualización en el grafo, llevando el enfoque al punto central del grafo.

Ajustar Vista: Ajusta automáticamente la vista para que todo el grafo sea visible dentro del marco de la pantalla, optimizando el ángulo y la distancia de la cámara.

Acercar: Acerca la cámara hacia el grafo, permitiendo una vista más cercana de los nodos y aristas.

Alejar: Aleja la cámara del grafo, proporcionando una vista más amplia de toda la estructura.

Acercar Poco: Realiza un acercamiento menor, permitiendo un ajuste fino de la distancia de la cámara hacia el grafo.

Alejar Poco: Realiza un alejamiento menor, ajustando la distancia de la cámara de manera más precisa.

Mover Subir: Desplaza la cámara hacia arriba, modificando el punto de vista vertical sobre el grafo.

Mover Abajo: Desplaza la cámara hacia abajo, modificando el punto de vista vertical en sentido contrario.

Mover Derecha: Desplaza la cámara hacia la derecha, alterando el ángulo horizontal del grafo.

Mover Izquierda: Desplaza la cámara hacia la izquierda, cambiando el ángulo horizontal en la dirección opuesta.

Ayuda: Proporciona información o asistencia sobre cómo utilizar las opciones de configuración de la cámara en la visualización 3D del grafo.

5.8.8. Vista de Administrador

La vista del Administrador es una herramienta que permite a los administradores gestionar de manera eficiente las organizaciones y los usuarios dentro de la plataforma:

The screenshot shows the 'Admin Dashboard' interface. At the top, there's a header with the title 'Admin Dashboard' and a sub-instruction: '¡Bienvenido al panel de administración! Aquí puedes gestionar los usuarios de la plataforma.' Below this, a section titled 'Seleccionar Datos' with the subtitle 'Utiliza el selector de datos para filtrar o gestionar datos específicos.' contains a dropdown menu labeled 'Empresa' with the placeholder 'Selecciona una Empresa'. A large blue button labeled 'Continuar' is positioned below the dropdown. To the right, there's a section titled 'Añadir Nueva Organización' with a sub-instruction 'Nombre de la nueva organización' and a text input field labeled 'Introduce el nombre'. A green button labeled 'Añadir Organización' is at the bottom of this section. At the bottom of the dashboard, there are two main sections: 'Listar Usuarios' (with a sub-instruction 'Ver todos los usuarios registrados.' and a blue 'Ver Usuarios' button) and 'Crear Usuario' (with a sub-instruction 'Añadir un nuevo usuario a la plataforma.' and a green 'Crear Usuario' button).

Figura 43. Vista Dashboard Administrador

5.8.8.1. Gestión de Organizaciones

Dentro de la administración donde podemos gestionar los usuarios, también podremos visualizar todos las visualizaciones 3D de grafos de conexiones de las distintas empresas cargadas en la plataforma, después de seleccionar cualquiera de ellas, bastará con interactuar con el botón “Continuar”, para ver la visualización 3D:

:

Seleccionar Datos
Utiliza el selector de datos para filtrar o gestionar datos específicos.

Seleciona La Empresa Con La Que Visualizar

Empresa

Seleciona una Empresa

Continuar

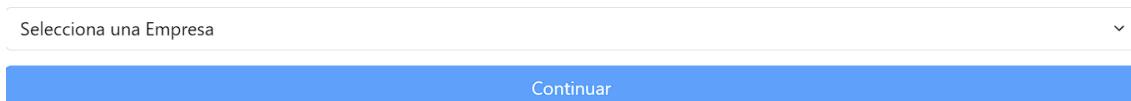


Figura 44. Vista Dashboard Administrador Seleccionar Organización

5.8.8.2. Carga de Nuevas Organizaciones

Otras de las funcionalidades que tiene la vista del admin es la capacidad de cargar nuevas empresas en la plataforma, esta debe coincidir exactamente con el nombre de referencia de la organización en GitHub, si no, no se cargará correctamente en la plataforma:

Añadir Nueva Organización

Nombre de la nueva organización

Introduce el nombre

Añadir Organización



Figura 45. Vista Dashboard Administrador Cargar Organización

5.8.9. Gestión de Usuarios

El dashboard proporciona accesos directos para realizar operaciones CRUD en los usuarios del sistema. Estas opciones permiten a los administradores gestionar la base de usuarios de manera eficiente.

5.8.9.1. Listar Usuarios:

La sección de "Listar Usuarios" permite a los administradores visualizar todos los usuarios registrados en el sistema. Esta lista es fundamental para revisar y gestionar la información de cada usuario, incluyendo su rol y la organización a la que pertenece.

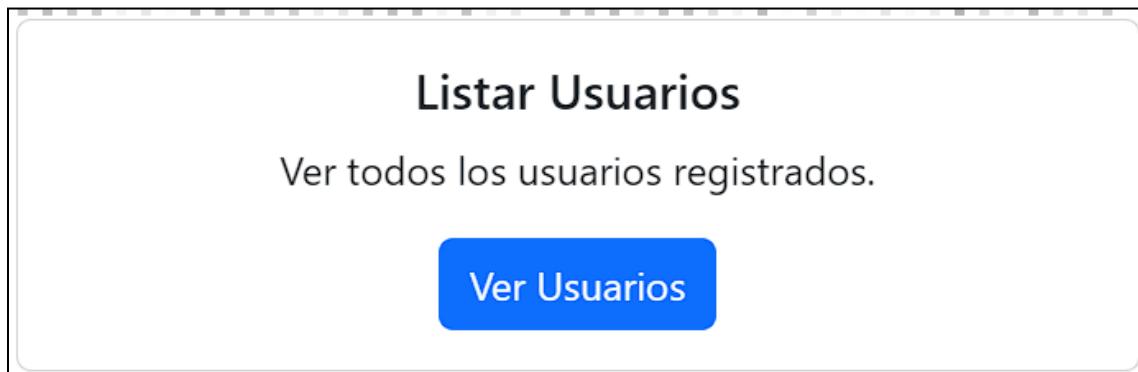


Figura 46. Vista Dashboard Administrador Listar Usuarios

5.8.9.1.1. Interfaz de Listado:

La lista de usuarios se muestra en una tabla que incluye varias columnas clave. Cada usuario está identificado por un ID, que es un identificador único en el sistema. La columna Usuario presenta el nombre registrado del usuario, mientras que la columna Rol indica el rol asignado, ya sea worker, manager o admin:

Lista de Usuarios del Sistema				
ID	Usuario	Rol	Organización	Acciones
6	yyyyyy	worker	EpicGames	<button>Editar</button> <button>Eliminar</button>
7	Yoel Pérez	manager	EpicGames	<button>Editar</button> <button>Eliminar</button>

Figura 47. Vista Listar Usuarios

También se muestra la Organización a la que pertenece cada usuario. Finalmente, en la columna Acciones, se encuentran los botones que permiten editar o eliminar al usuario.

5.8.9.1.2. Acciones Disponibles:

Editar: Permite modificar la información del usuario. Al hacer clic en el botón "Editar", se abre un formulario donde se pueden actualizar los datos como nombre, rol, y organización. Despues de hacer los cambios, asegúrate de guardar para que se apliquen.

Eliminar: Permite borrar un usuario del sistema. Al hacer clic en "Eliminar", se te pedirá confirmar la acción para evitar eliminaciones accidentales. Una vez confirmada, el usuario se eliminará de la base de datos permanentemente.

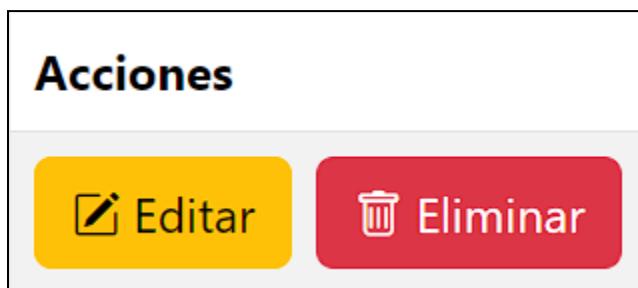


Figura 48. Vista Listar Usuarios Acciones

5.8.10. Crear Usuario:

El formulario de "Crear Usuario" es utilizado para añadir nuevos usuarios a la plataforma. Esta herramienta es esencial para garantizar que todos los empleados o miembros de una organización estén correctamente registrados y asignados a sus roles dentro del sistema.

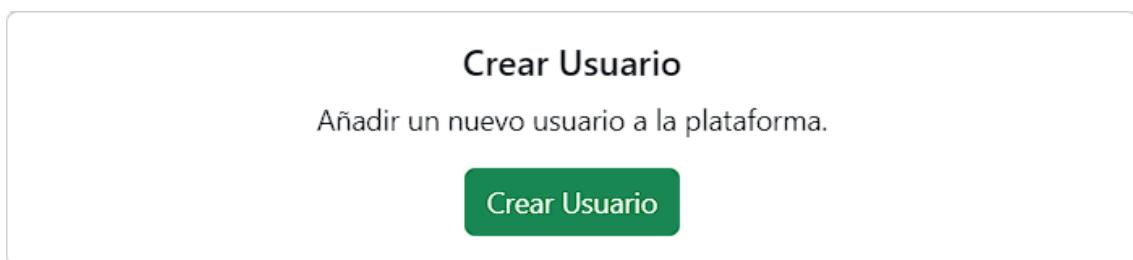


Figura 49. Vista Dashboard Administrador Crear Usuario

5.8.10.1. Interfaz de Creación de Usuario:

El formulario está diseñado para capturar la información necesaria para crear un nuevo usuario en la plataforma. Se debe ingresar un nombre de usuario que identificará al nuevo miembro, seguido de una contraseña que protegerá su cuenta.

The screenshot shows a user interface for creating a new user. At the top, there's a header bar with the logo 'MyParallelOrg' and navigation links 'Inicio', 'Dashboard', and 'admin (Admin)'. Below the header, the main title is 'Crear Usuario Nuevo En El Sistema'. The form consists of four input fields: 'Usuario' (User) with placeholder 'Introduce usuario', 'Contraseña' (Password) with placeholder 'Introduce contraseña', 'Nombre de la Organización' (Organization Name) with placeholder 'Introduce nombre de la empresa', and 'Rol' (Role) with placeholder 'Selecciona rol'. A blue button labeled 'Crear Usuario' (Create User) is located at the bottom left of the form area.

Figura 50. Vista Crear Usuario

Además, es necesario especificar el nombre de la organización a la que pertenece el usuario y seleccionar su rol dentro de la misma, ya sea como Worker, Manager o Admin. Una vez que todos estos campos han sido completados correctamente, se puede hacer clic en el botón "Crear Usuario" para añadir el nuevo usuario al sistema.

6. Limitaciones del Sistema

La limitación más importante a la que se enfrenta el sistema es cuando una empresa u organización no cuenta con un número suficiente de trabajadores o cuando su dataset es extremadamente grande. Suponiendo que nos ponemos en el caso de que no cuenta con lo mínimos, la generación del grafo, no sería suficiente para obtener unas hipótesis razonables.

Este problema se agrava si los componentes de la empresa no tienen ningún tipo de interacción o conexión, ya que el propósito del sistema es identificar y mostrar relaciones entre empleados.

Implementar este límite ayudaría a evitar que organizaciones con insuficientes interacciones o datos sean procesadas por el sistema, garantizando así que el análisis sea efectivo y útil.

7. Conclusiones y Líneas de Trabajo Futuras

7.1. Conclusiones

Una vez finalizada toda la fase de implementación y pruebas, se puede decir que entremos en la fase de mantenimiento y monitorización software, una vez que ya estamos en esta fase es donde se pueden sacar conclusiones afianzadas de nuestros logros y derrotas.

Haciendo un balance general y retomando los objetivos iniciales, podemos concluir que la implementación ha alcanzado muchos de los hitos establecidos. El sistema ha demostrado ser capaz de gestionar grandes volúmenes de datos, conectar a empleados de manera eficaz y ofrecer una visualización 3D clara de las relaciones laborales a través de GitHub. Además, la integración de herramientas como Docker ha proporcionado una base sólida y fiable para garantizar la estabilidad y escalabilidad del sistema, lo que nos ha permitido trabajar con mayor confianza sabiendo que el entorno es seguro y fácilmente replicable.

Sin embargo, también he tenido complicaciones. Algunos de los problemas que surgieron durante el desarrollo, como la optimización de conexiones entre usuarios en base a criterios complejos o la falta de datos suficientes en ciertos escenarios, subrayan la necesidad de futuras mejoras y ajustes. Estos desafíos nos han proporcionado valiosas lecciones sobre la importancia de la planificación y la adaptación de la infraestructura para hacer frente a imprevistos. En conjunto, tanto los logros como las dificultades nos han permitido fortalecer el sistema y plantear líneas de trabajo futuras que aseguren una mejora continua.

7.2. Líneas de Trabajo Futuras

7.2.1. Back-End

7.2.1.1. Mejoras en las conexiones entre usuarios:

Esta mejora traería una complejidad extra al procesamiento de los datos y a su vez de procesamiento de la visualización de las conexiones. La principal mejora que se puede hacer a la conexiones por la heurística utiliza, no es más

7.2.1.2. Migración a una infraestructura cloud:

Migrar el sistema a una infraestructura cloud, lo que mejoraría la disponibilidad continua, la escalabilidad y la capacidad de gestionar grandes volúmenes de datos en tiempo real.

7.2.2. Front-End

7.2.2.1. Incorporación de notificaciones y chat:

Incorporar un sistema de notificaciones y un chat interno para facilitar la comunicación y colaboración en tiempo real entre los usuarios.

8. Bibliografía

[1] API. (2024). En *Wikipedia, la enciclopedia libre*.

<https://es.wikipedia.org/w/index.php?title=API&oldid=162005312>

[2] GraphEverywhere, E. (2020, agosto 21). ¿Qué es el Clustering? | Detección de Comunidades. *GraphEverywhere*.

<https://www.grapheverywhere.com/que-es-el-clustering/>

[3] Atlassian. (s. f.). *Git commit | Atlassian Git Tutorial*. Atlassian.

Recuperado 5 de septiembre de 2024, de

<https://www.atlassian.com/es/git/tutorials/saving-changes/git-commit>

[4] Grafo (tipo de dato abstracto). (2021). En *Wikipedia, la enciclopedia libre*.

[https://es.wikipedia.org/w/index.php?title=Grafo_\(tipo_de_dato_abstracto\)&oldid=138019502](https://es.wikipedia.org/w/index.php?title=Grafo_(tipo_de_dato_abstracto)&oldid=138019502)

[5] JavaScript With Syntax For Types. (s. f.). Recuperado 5 de septiembre de 2024, de <https://www.typescriptlang.org/>

[6] HTML. (2024). En *Wikipedia, la enciclopedia libre*.

<https://es.wikipedia.org/w/index.php?title=HTML&oldid=161854547>

[7] ¿Qué es CSS? - CSS en español. (s. f.). Recuperado 5 de septiembre de 2024, de <https://lenguajecss.com/css/introduccion/que-es-css/>

[8] Axios (sitio web). (2024). En *Wikipedia, la enciclopedia libre*.

[https://es.wikipedia.org/w/index.php?title=Axios_\(sitio_web\)&oldid=159432337](https://es.wikipedia.org/w/index.php?title=Axios_(sitio_web)&oldid=159432337)

[9]Bootstrap (framework). (2024). En *Wikipedia, la enciclopedia libre*.
[https://es.wikipedia.org/w/index.php?title=Bootstrap_\(framework\)&oldid=160017323](https://es.wikipedia.org/w/index.php?title=Bootstrap_(framework)&oldid=160017323)

[10]*FastAPI*. (s. f.). Recuperado 5 de septiembre de 2024, de
<https://fastapi.tiangolo.com/>

[11]*Npm | Home*. (s. f.). Recuperado 5 de septiembre de 2024, de
<https://www.npmjs.com/>

[12]PostgreSQL. (2024). En *Wikipedia, la enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=PostgreSQL&oldid=161450636>

[13]Docker (software). (2024). En *Wikipedia, la enciclopedia libre*.
[https://es.wikipedia.org/w/index.php?title=Docker_\(software\)&oldid=161987879](https://es.wikipedia.org/w/index.php?title=Docker_(software)&oldid=161987879)

[14]*JSON Web Token Libraries—Jwt.io*. (s. f.). Recuperado 5 de septiembre de 2024, de <https://jwt.io/libraries>

[15]*Welcome to Pydantic—Pydantic*. (s. f.). Recuperado 5 de septiembre de 2024, de <https://docs.pydantic.dev/latest/>

[16]NetworkX. (2024). En *Wikipedia, la enciclopedia libre*.
<https://es.wikipedia.org/w/index.php?title=NetworkX&oldid=162276428>

[17]*SQLAlchemy*. (s. f.). Recuperado 5 de septiembre de 2024, de
<https://www.sqlalchemy.org>

[18]Visual Studio Code. (2024). En *Wikipedia, la enciclopedia libre*.
https://es.wikipedia.org/w/index.php?title=Visual_Studio_Code&oldid=162215034

[19] *Software de administración de proyectos | Microsoft Project*. (s. f.). Recuperado 5 de septiembre de 2024, de <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>

[20] *Ideal Modeling & Diagramming Tool for Agile Team Collaboration*. (s. f.). Recuperado 5 de septiembre de 2024, de <https://www.visual-paradigm.com/>

[21] Sphinx (generador de documentación). (2023). En *Wikipedia, la encyclopédia libre*. [https://es.wikipedia.org/w/index.php?title=Sphinx_\(generador_de_documentaci%C3%B3n\)&oldid=156159727](https://es.wikipedia.org/w/index.php?title=Sphinx_(generador_de_documentaci%C3%B3n)&oldid=156159727)

[22] *Install Tailwind CSS with Create React App—Tailwind CSS*. (s. f.). Recuperado 5 de septiembre de 2024, de <https://tailwindcss.com/docs/guides/create-react-app>

[25] *Reaviz/reagraph*. (2024). [TypeScript]. REAVIZ. <https://github.com/reaviz/reagraph> (Obra original publicada en 2022)

[25] Moreno García, M. N. (2023). *Gestión de proyectos del software*.