

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

Anexo IV. Documentación técnica de programación



TRABAJO DE FIN DE GRADO  
INGENIERÍA INFORMÁTICA

Yoel Pérez Carrasco

Dirigido por Fernando De La Prieta Pintado y Walter Vinci

Salamanca, septiembre 2024

## Tabla de contenidos:

<b>1. Introducción.....</b>	<b>4</b>
<b>2. Estructuración.....</b>	<b>4</b>
2.1. Frontend.....	4
2.1.1. Componentes React (src/components):.....	4
2.1.2. Contextos y Hooks Personalizados (src/context y src/utills):.....	4
2.1.3. Vistas (src/views):.....	4
2.1.4. Estilos (src/styles):.....	5
2.1.5. Configuración de la Aplicación (src):.....	5
2.2. Backend.....	5
2.3. Docker.....	5
<b>3. Alojamiento Documentación Técnica.....</b>	<b>6</b>
3.1. Frontend.....	6
Figura 1. Imagen vista documentación frontend.....	6
3.2. Backend y Docker.....	7
Figura 2. Imagen vista documentación backend y Docker.....	7

Tabla de tablas:

No contiene tablas este Anexo.

Tabla de figuras:

Figura 1. Imagen vista documentación frontend.....	6
Figura 2. Imagen vista documentación backend y Docker.....	7

# 1. Introducción

En este anexo se presenta la organización del proyecto y se proporciona una explicación detallada del código, complementada con herramientas de generación de documentación. El anexo se divide en las siguientes secciones:

**Estructuración:** Se describe la organización del proyecto en sus diferentes directorios, carpetas y fuentes.

**Alojamiento Documentación Técnica:** Se proporciona una descripción detallada del código para facilitar su comprensión.

## 2. Estructuración

### 2.1. Frontend

#### 2.1.1. Componentes React (`src/components`):

`DataSelector.tsx`

`FlickeringGrid.tsx`

#### 2.1.2. Contextos y Hooks Personalizados (`src/context` y `src/utils`):

`AuthContext.tsx`

`RequireAdminToken.tsx`

`RequireManagerToken.tsx`

`jwtDecode.tsx`

#### 2.1.3. Vistas (`src/views`):

`Home.tsx`: Página Principal

`LoginV2.tsx`: Página de Inicio de Sesión

`UserDashboard.tsx`: Dashboard del Usuario

`ManagerDashboard.tsx`: Dashboard del Manager

`AdminDashboard.tsx`: Dashboard del Administrador

`UserGraph.tsx`: Visualización del Grafo de Usuarios

#### 2.1.4. Estilos (`src/styles`):

`Navbar.css`: Estilos para la Barra de Navegación

`App.css`, `index.css`: Estilos Globales

#### 2.1.5. Configuración de la Aplicación (`src`):

`App.tsx`: Componente Principal

`index.tsx`: Punto de Entrada del Frontend

## 2.2. Backend

`main.py`: Punto de Entrada del Backend

`crud.py`: Operaciones CRUD

`database.py`: Configuración de la Base de Datos

`models.py`: Definición de Modelos de Datos

`requestgithub.py`: Interacciones con APIs Externas

## 2.3. Docker

Todos los ficheros fuentes necesarios para levantar las imagenes y contenedores necesarios poner en marcha el server API y la base de datos de Postgres se encuentran en tres ficheros en la raiz del proyecto estos son:

- `docker-compose.yml`
- `Dockerfile`
- `requeriments.txt`

## 3. Alojamiento Documentación Técnica

Se ha utilizado Sphinx para la creación de la documentación técnica del proyecto. Sphinx es una herramienta muy flexible que permite generar documentación en formato HTML a partir de archivos en Markdown.

Se ha generado a mayores un carpeta dentro de la documentación donde se almacenarán los html géneros como backup de la documentación técnica “./Documentacion/Tecnica”.

### 3.1. Frontend

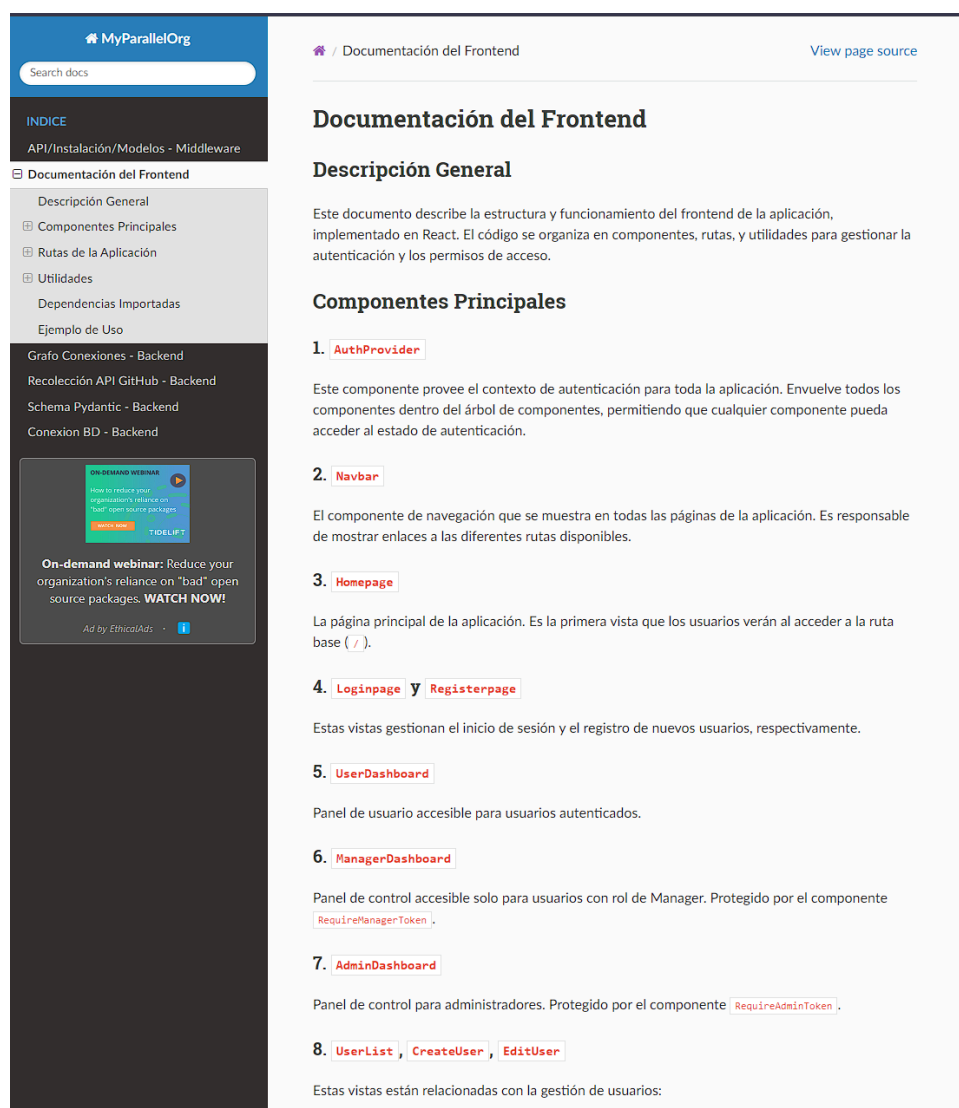


Figura 1. Imagen vista documentación frontend

Enlace Read The Docs de la documentación del Frontend:

- [Frontend](#)

## 3.2. Backend y Docker

The screenshot displays the documentation for 'API/Instalación/Modelos - Middleware' on the MyParallelOrg website. The left sidebar contains a search bar and a table of contents with links to various sections like 'Instalación', 'Configuración', and 'Endpoints'. The main content area features a title, a brief description of the project as a FastAPI-based API, and a 'Tabla de Contenidos' (Table of Contents) listing topics such as 'Instalación', 'Configuración', 'Configuración de Docker', 'Modelos de Base de Datos', 'Autenticación y Autorización', 'Endpoints', 'Manejo de Errores', and 'Logging'. Below this, the 'Instalación' (Installation) section provides a list of steps: cloning the repository, installing dependencies (with a code block for 'pip install -r requirements.txt'), configuring the database, and starting the application (with a code block for 'uvicorn main:app --reload'). The 'Configuración' (Configuration) section follows, with a 'Variables de Entorno' (Environment Variables) subsection listing 'SECRET\_KEY', 'ALGORITHM', and 'ACCESS\_TOKEN\_EXPIRES\_MINUTES'.

Figura 2. Imagen vista documentación backend y Docker

Enlace Read The Docs de la documentación del Frontend:

- [Backend y Docker](#)