



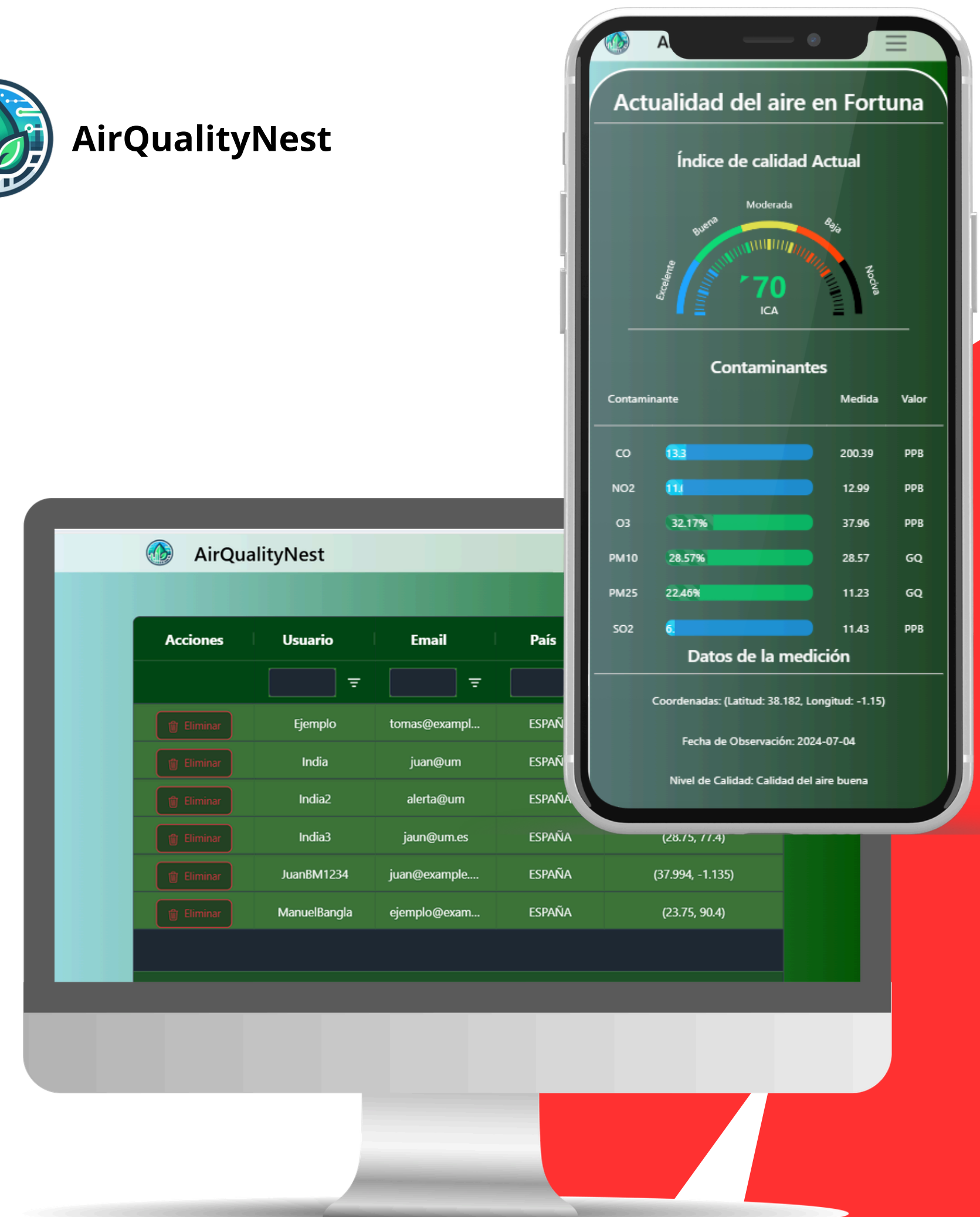
TFG UNIVERSITARIO

DESARROLLO DE UNA APLICACIÓN WEB Y MÓVIL PARA LA MONITORIZACIÓN, NOTIFICACIÓN, VISUALIZACIÓN Y PREDICCIÓN DE LA CALIDAD DEL AIRE.

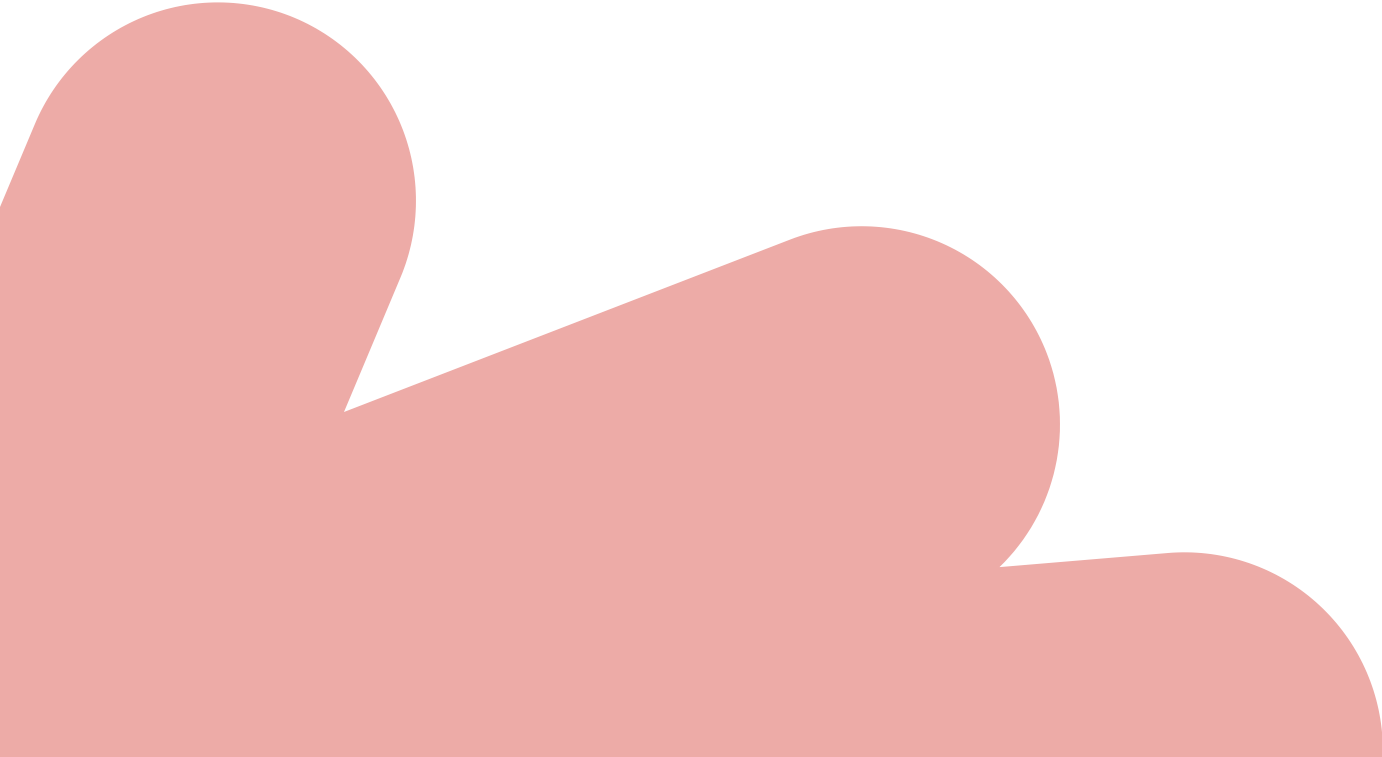
Por:
Juan Bernal Miralles

Tutoras:
Raquel Martínez España,
María Del Carmen Garrido
Carrera

Fecha:
Miércoles, 10 de julio
de 2024



Índice de CONTENIDOS

An abstract, organic red shape is located in the bottom-left corner of the slide, partially overlapping the white background.

1 . Introducción

2 . Estado del arte

3 . Diseño del trabajo realizado

4 . Desarrollo e implementación de la aplicación

5 . Pruebas y despliegue

6 . Conclusiones y vías futuras



SECCIÓN 1

INTRODUCCIÓN

Problema de rigurosa actualidad

Uno de los mayores riesgos ambientales que existen para la salud es el empeoramiento de la contaminación del aire. Este riesgo provoca cada año 4,2 millones de muertes prematuras.

Avances tecnologicos

Las ciudades están experimentando un gran avance implementando nuevas tecnologías. Bajo este concepto surgen las ciudades inteligentes que llegan de la mano de sistemas IoT, big data, inteligencia artificial, etc.

FIWARE nació bajo el paraguas del VII Programa Marco de la Unión Europea en colaboración público-privada para el desarrollo del Internet del Futuro. Su principal dominio es el de ciudades inteligentes, aunque se aplica a más campos.

Solución

El desarrollo de una aplicación web multiplataforma para la monitorización, notificación, visualización y predicción de la calidad del aire.

APLICACIONES SIMILARES

Especializadas en calidad del aire

IQAir y Plume Labs son dos aplicaciones multiplataformas especializadas en la calidad del aire, que ofrecen un sistema de visualización y adquisición de dispositivos para la monitorización.



De carácter general

Accuweather y Aemet son dos aplicaciones multiplataformas enfocadas en la visualización del clima. Estas tecnologías agregan la calidad del aire como apartado.



TECNOLOGÍAS Y HERRAMIENTAS



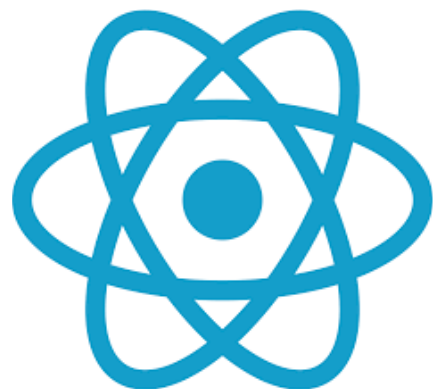
Python

Es un lenguaje muy utilizado para el desarrollo de software y el machine learning. Esta tecnología será utilizada para el modelo de predicción



Spring

Es un framework para construir aplicaciones empresariales en Java. Y usada en el desarrollo de la API de calidad.



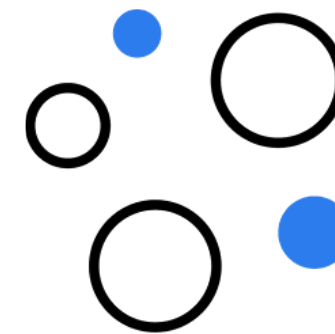
React

Es una librería de JavaScript aplicada al proyecto para el desarrollo de interfaces de usuarios a través de componentes reutilizables.



MySQL

Es un sistema de administración de bases de datos relacional. Es la base de datos utilizada para la aplicación junto a MongoDB. Esta es necesaria para FIWARE.



API Air Quality de Google

Proporciona datos sobre la calidad del aire en tiempo real. Esta API es la encargada de abastecer de datos a la aplicación.

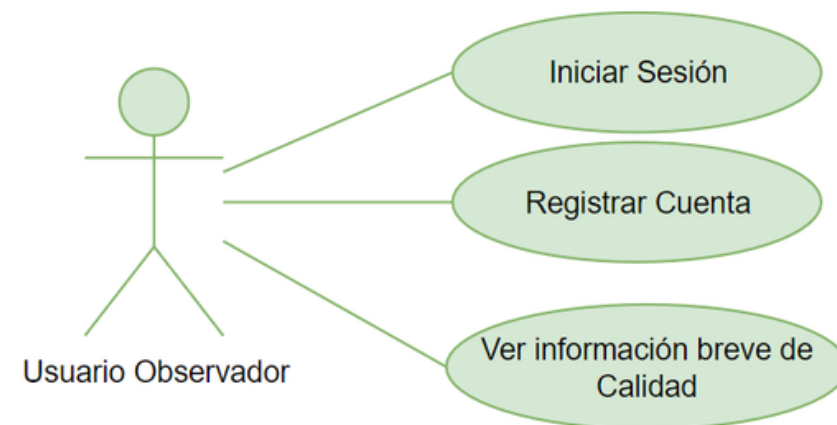


VS Code y Docker

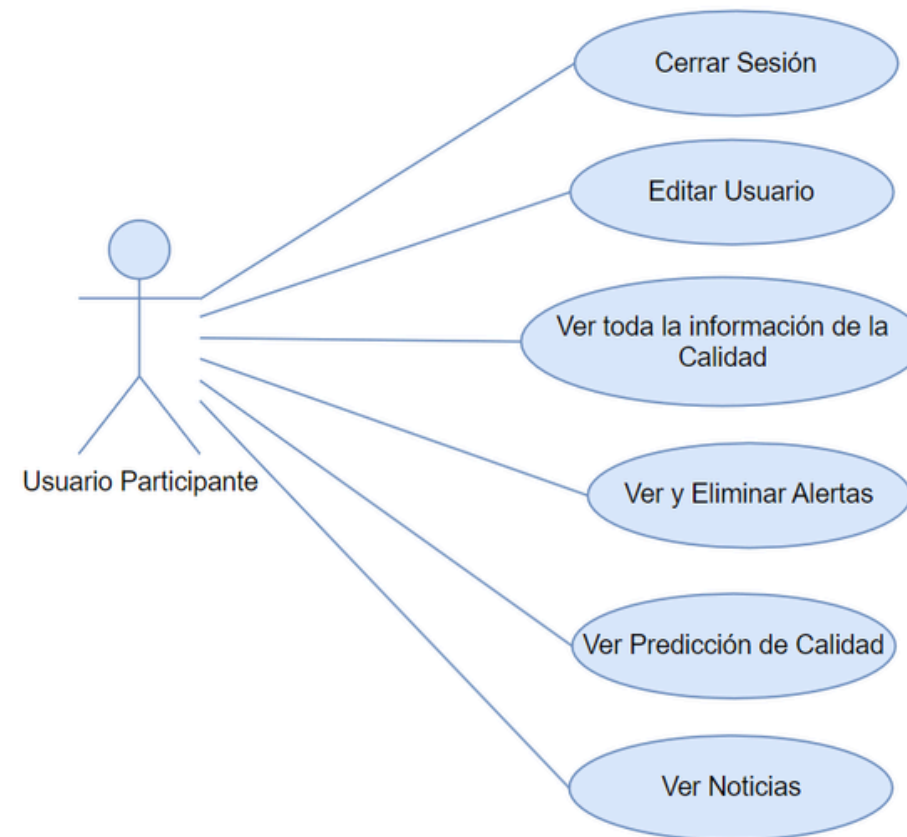
VS Code ha sido elegido como entorno de desarrollo que admite todas las tecnologías utilizadas. El sistema Docker es el responsable del despliegue de los componentes.

REQUISITOS DE LA APLICACIÓN

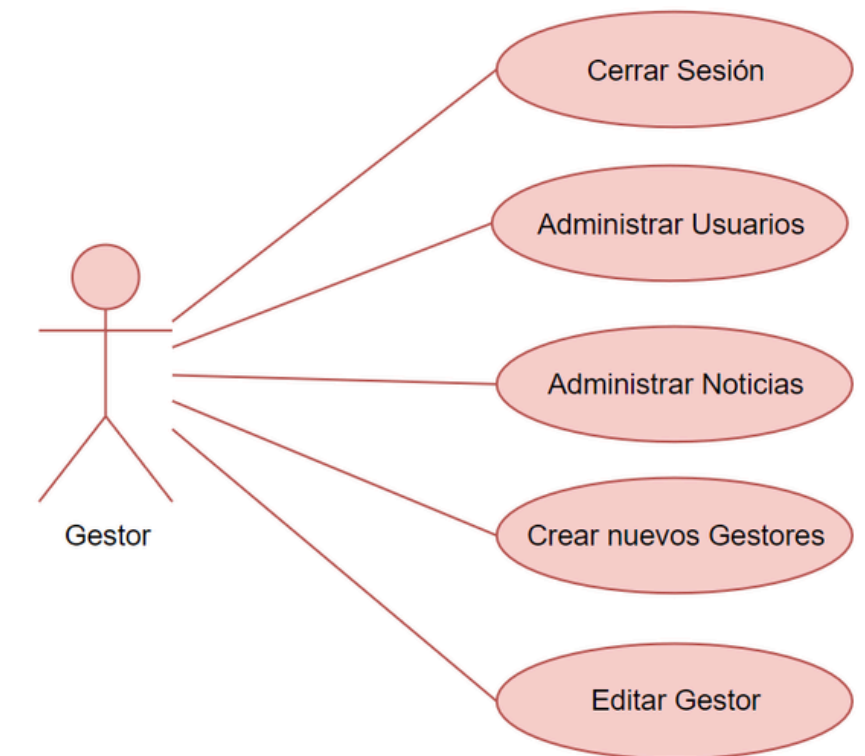
En la aplicación existen tres tipos de actores. Los requisitos de cada actor son representados por medio de casos de uso en lenguaje natural.

**Usuario Observador**

Este usuario no necesita estar registrado en la aplicación. La única función disponible, además de iniciar y registrarse en la herramienta, es consultar un resumen de la calidad del aire en su ubicación.

**Usuario Participativo**

Es el usuario principal de la aplicación que debe estar registrado. Sus principales funciones son ver la calidad del aire para su ubicación, administrar sus alertas, ver las noticias para su país y la predicción de calidad del aire para los próximos 7 días.

**Gestor**

Es el encargado de administrar las noticias y los usuarios participativos. También tiene la capacidad de crear nuevos usuarios gestores.

FIWARE

FIWARE es un ecosistema de innovación y emprendimiento abierto basado en las tecnologías del internet del futuro. Una de sus principales utilidades consiste en recopilar y administrar información de contexto. Este ecosistema cuenta con una serie de componentes que tiene como principal el Context Broker. Para interactuar cuenta con una API restful FIWARE Interfaces de Servicio de Nueva Generación (NGSI).



Orion Context Broker

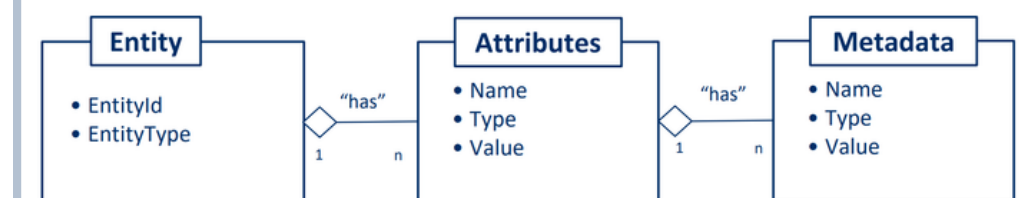
Es el principal y único componente obligatorio de cualquier plataforma o solución desarrollada con FIWARE.

Entities

GET	/v2/entities	List Entities
POST	/v2/entities	Create Entity
GET	/v2/entities/{entityId}	Retrieve Entity
DELETE	/v2/entities/{entityId}	Remove Entity

NGSIV2

Es una API que permite la gestión de información de contexto.



Modelo NGSI

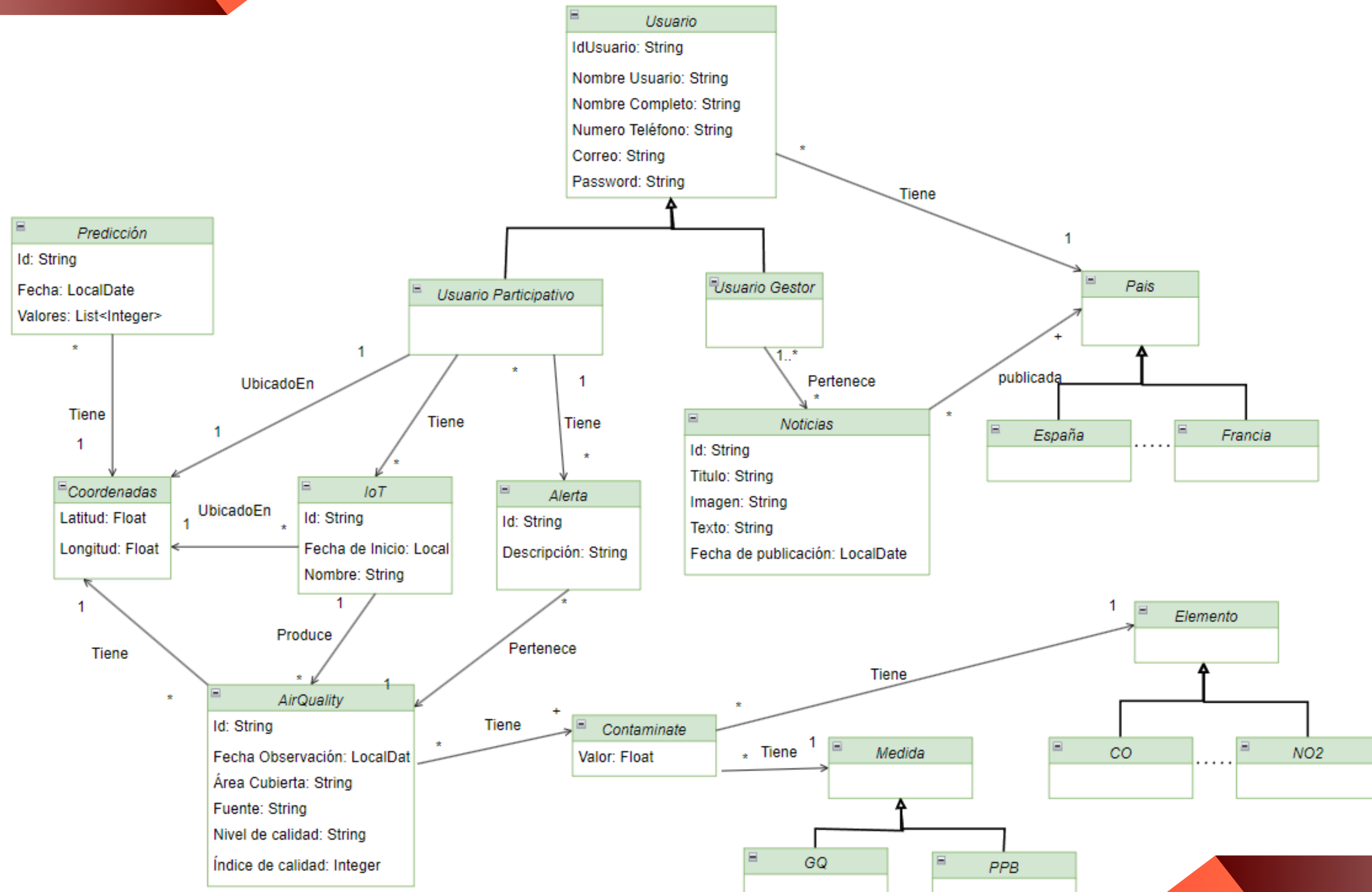
Los datos de contexto en FIWARE son representados a través de estructuras de datos genéricos.

AirQualityObserved

AirQualityObserved

Es el modelo seleccionado para representar la calidad del aire en un lugar y momento determinado.

MODELADO DEL DOMINIO



ARQUITECTURA DE LA APLICACIÓN

Componentes

La aplicación consta de cuatro componentes principales:

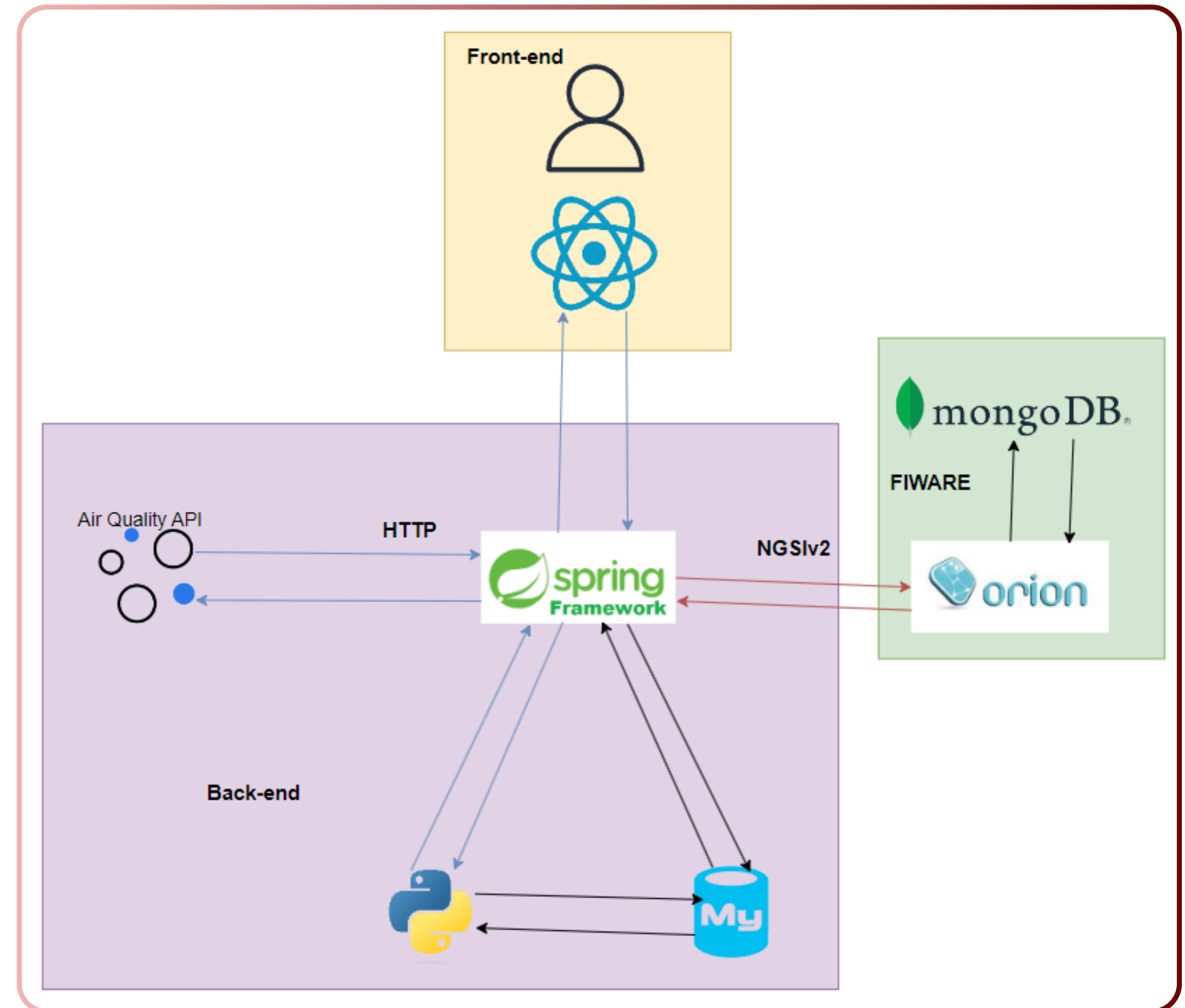
- Front-end: Esta desarrollado en React. Es la interfaz de los usuarios con el sistema.
- API de calidad: El componente principal encargado de la comunicación entre los demás elementos y de la lógica de negocio. Se encuentra implementado en Spring.
- API de predicción: Encargado de obtener la predicción de la calidad. Se halla en programado en Python.
- FIWARE: Responsable de los datos de contexto de la aplicación.

Arquitectura limpia

Para el proyecto se ha seguido los métodos de la arquitectura limpia. Se usa los principios SOLID y patrones de diseño para separar las responsabilidades y una estructura modular.

Arquitectura REST

Para la creación de las APIs se ha seguido la arquitectura REST. Esta arquitectura ha permitido crear APIs fáciles de usar, escalables, confiables, interoperables y que ofrecen una buena experiencia para los desarrolladores.



PROTOTIPO DE LA APLICACIÓN

Se han desarrollado los prototipos de las ventanas de registro, estado actual de la calidad del aire y predicción de la misma. Estos prototipos se han desarrollado con la herramienta Canva.



SECCIÓN 4

DESARROLLO E IMPLEMENTACIÓN DE LA APLICACIÓN

DESARROLLO DE LA API DE CALIDAD DEL AIRE

Es una API REST desarrollada con el framework de Spring. Es la responsable de las comunicaciones y la lógica principal de la aplicación. En ella se encuentran los end-point utilizados por el front-end, un temporizador para la obtención de datos, la seguridad de la aplicación, entre otras.

La arquitectura de la API está enfocada en una arquitectura por capas siguiendo los principios de arquitectura limpia y SOLID. Las capas en las que se divide este componente son las siguientes:

1

Capa de Presentación

2

Capa de Persistencia

3

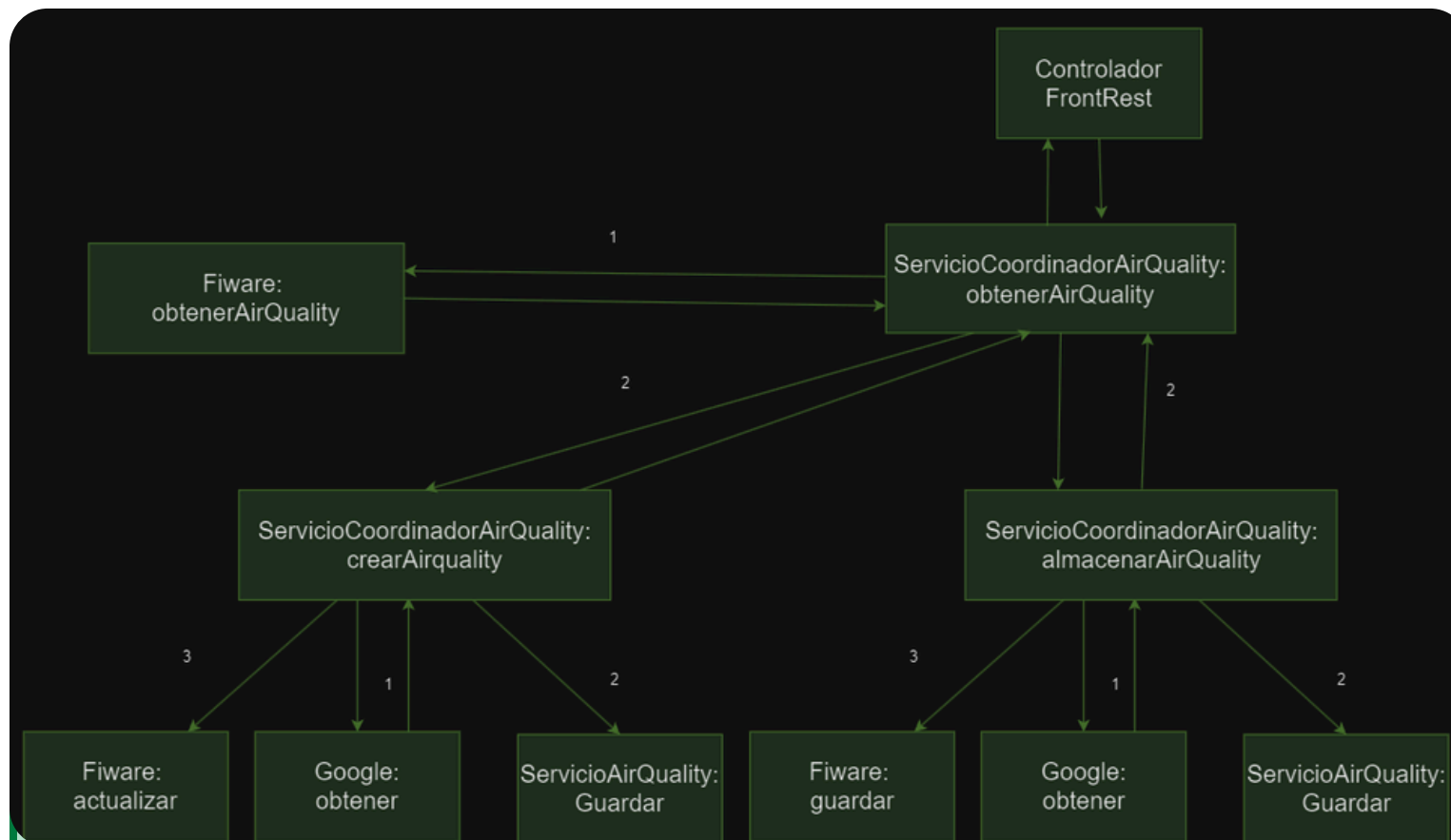
Capa de Dominio

4

Capa de Soporte

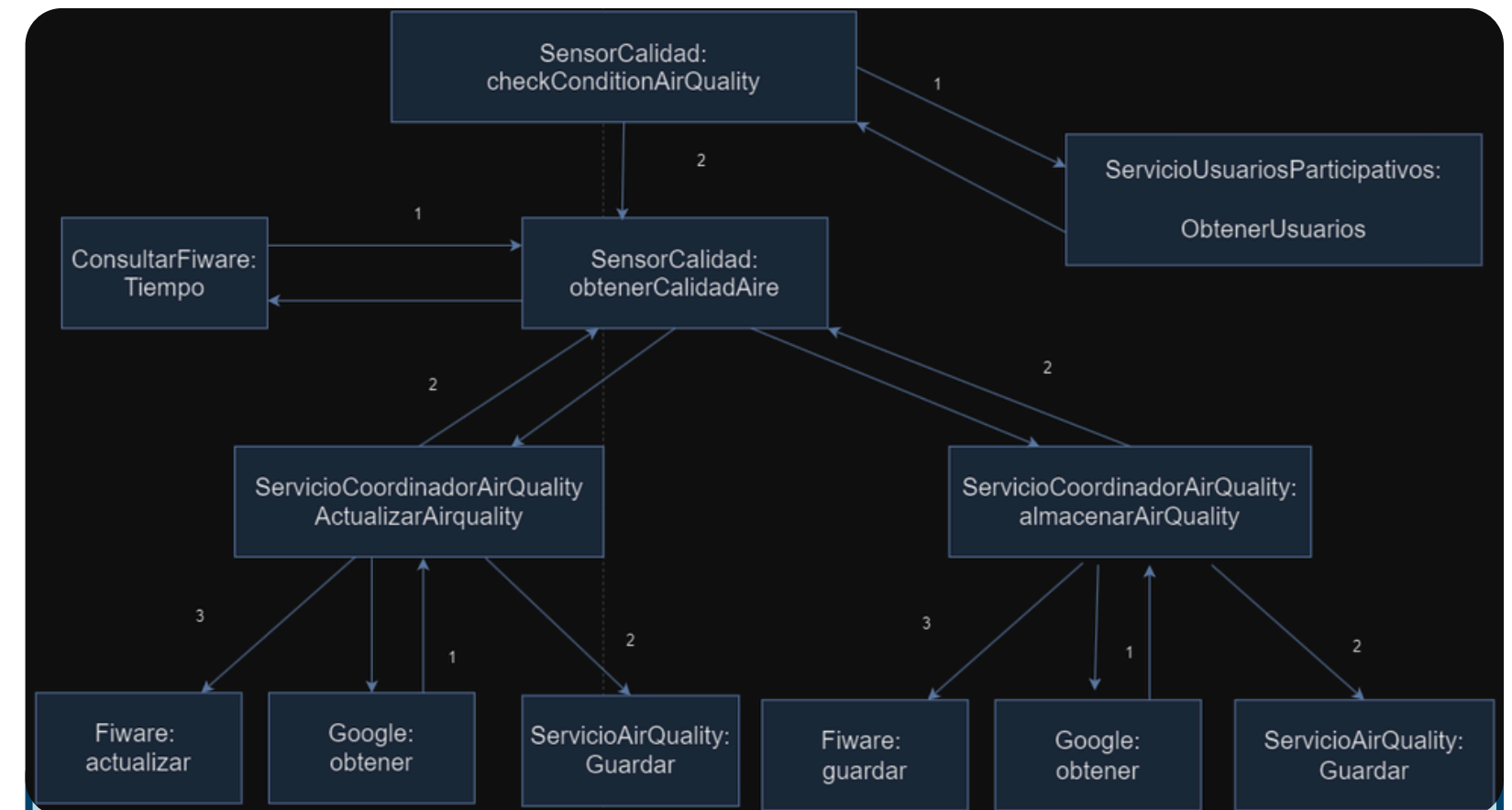
5

Capa de Lógica de Negocio



Obtención de calidad del aire

Para obtener la calidad del aire por parte de los usuarios de la aplicación se comprueba si la medición existe en FIWARE y no es más antigua que un día. Si no se cumple, se obtiene de la API de Google.



Temporizador de la calidad

Para tener un registro y mantener actualizados a los usuarios participativos se obtiene la calidad del aire en su posición cada hora.

DESARROLLO DE LA API DE PREDICCIÓN DE LA CALIDAD DEL AIRE

Descripción

Está desarrollada en Python haciendo uso de diferentes librerías. Esta ofrece un end-point REST para obtener la predicción de la calidad del aire de los próximos 7 días para una coordenada dada. Es importante destacar que este no es el elemento principal de la aplicación, sino uno complementario.

Librerías

Se han empleado diferentes librerías como Pandas (manipulación y análisis de datos), Flask (creación del end-point REST), Scikit-learn (creación un modelo RandomForestRegressor), entre otras.

Elementos principales

- Model.py: en este archivo se obtienen los datos, se seleccionan las características y el objetivo, se crea un modelo RandomForestRegressor, se entrena y se almacena.
- Api.py: Contiene la función predict que corresponde al end-point REST, el cual se encarga de llamar al modelo de predicción y obtener la calidad del aire de los próximos 7 días.

Análisis de datos y entrenamiento del modelo

Se ha desarrollado un pequeño análisis para verificar las mejores características y la calidad de la predicción.

Parámetros de entrada	MAE
Sin Ordenar 0 días	11,3356
Ordenado 0 días	7,7642
1 día anterior	5,7613
2 días anteriores	4,9886
3 días anteriores	6,454
4 días anteriores	5,2666
5 días anteriores	6,236
6 días anteriores	3,8459
7 días anteriores	7,1319
8 días anteriores	5,9296
9 días anteriores	9,9552

DESARROLLO DEL FRONTEND

Descripción

Es el elemento de la aplicación que permite a los usuarios interactuar con la funcionalidad de AirQualityNest. Se ha desarrollado en React manejando diferentes librerías. Los dos principales objetivos de la interfaz son: la sencillez y su facilidad de uso. También debe ser responsiva. Esta interfaz es una SPA con lo que reduce la carga de las páginas provocando una mejor experiencia. Este componente también cuenta con una autenticación por rutas.



Librerías

- **Vite:** Es un empaquetador de React que permite el desarrollo de forma sencilla y rápida.
- **Bootstrap:** Librería de componentes que permite crear interfaces web responsivas de forma rápida y sencilla.
- **React Router Dom:** Permite agregar un enrutamiento de páginas dinámicamente.

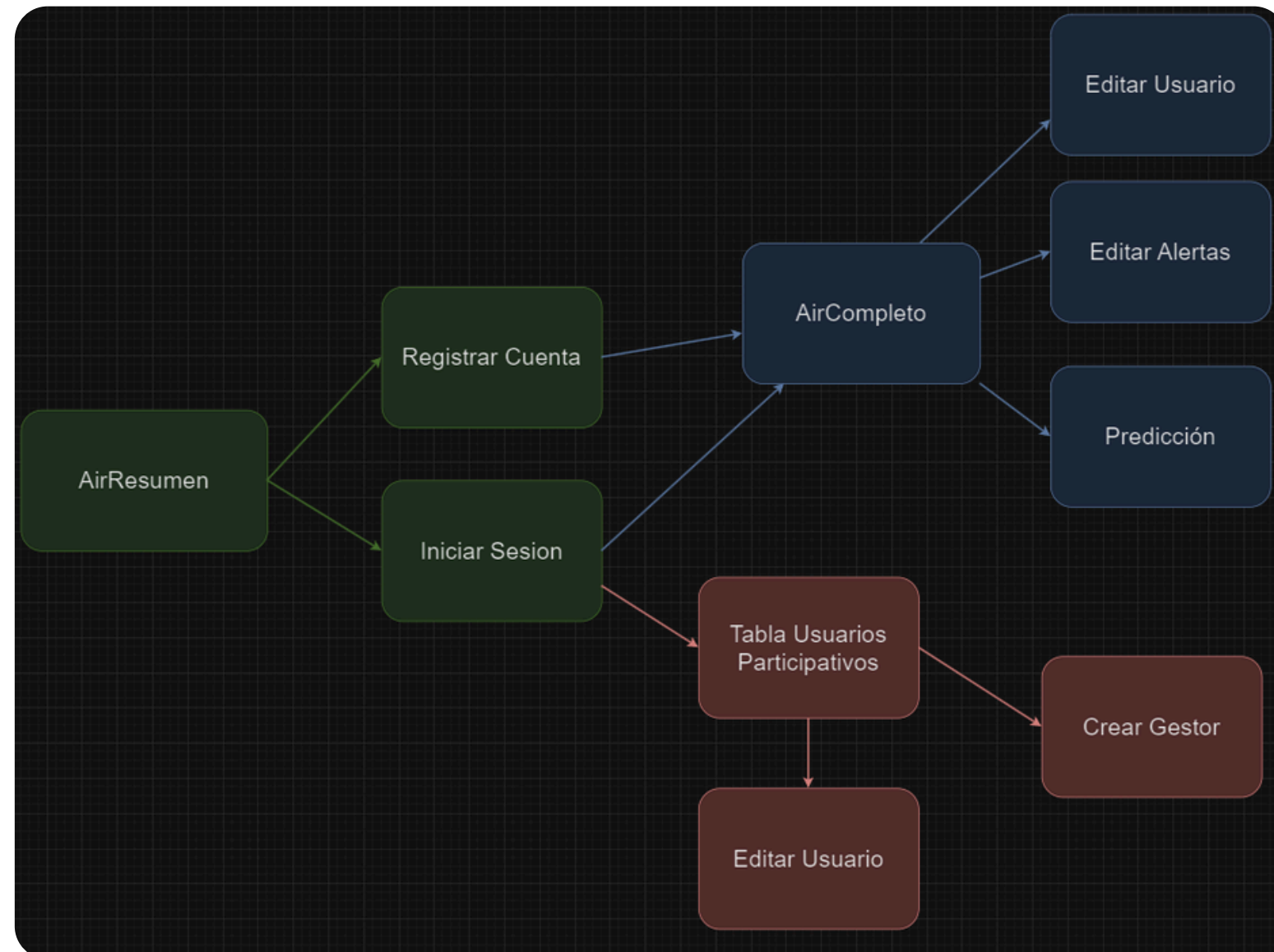
Estructura

Este componente se estructura en diferentes paquetes, siendo los más importantes :

- **Paquete component:** Contiene los componentes utilizados por otros elementos de la aplicación .
- **Paquete pages:** Compuesto de los componentes que son utilizados en las rutas.

MAPA DE NAVEGACION

La aplicación se divide en tres secciones principales, dependiendo de la autenticación necesaria.



No autenticado

1

Las páginas disponibles para este usuario son:

- **AirResumen:** Presenta la calidad del aire de hoy resumida en las coordenadas actuales del usuario.
- **Inicio de sesión**
- **Registro usuario participativo**

Autenticado como usuario

2

Para acceder a estas páginas el usuario debe estar autenticado:

- **AirCompleto:** Muestra la calidad del aire actual para la ubicación del usuario.
- **Predicción:** Permite la visualización de la predicción de la calidad del aire para su ubicación.
- **Alertas:** Lista de alertas del usuario.
- **Perfil:** Información personal del usuario. Es modificable.

Autenticado como gestor

3

Para acceder a estas páginas se necesita estar autenticado:

- **Tabla de usuarios:** Tabla con los usuarios participativos de la aplicación
- **Crear usuario Gestor:** Página para dar de alta a los gestores.
- **Perfil:** Información personal del gestor. Es modificable.

Pruebas

Se ha llevado a cabo diferentes pruebas de caja blanca durante el desarrollo de la aplicación.

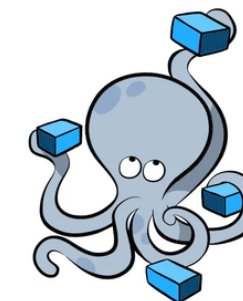
- Pruebas unitarias: realizadas con unittest y mock. Sirven para comprobar la correcta ejecución de los servicios de la aplicación de forma aislada.
- Pruebas de integración: utilizadas para comprobar el correcto funcionamiento de los componentes con los demás componentes de la aplicación.
- Pruebas de sistema: Validan la correcta ejecución de la aplicación en su conjunto.



Despliegue

Con la plataforma Docker se ha desplegado la aplicación. Para esto se ha creado un **docker-compose.yml** y un **Dockerfile** por cada componente desarrollado.

Los puertos que se han expuesto son el **4800** Front-end, **8080** API de calidad y **1026** servicio de Orion Context Broker.



CONCLUSIONES

Se ha podido alcanzar los principales objetivos iniciales del proyecto. La aplicación resultante permite monitorizar, visualizar, notificar y predecir la calidad del aire a los usuarios.

En el proyecto he podido aplicar la mayoría de los conocimientos adquiridos durante la carrera. Al principio del proyecto, había un reto muy sugerente y a la vez complicado, pero a medida que se avanzaba en el desarrollo, aumentaba la curiosidad de éste.

Esta versión de la aplicación es una inicial no final. Por ejemplo, no se ha podido implementar en el front la funcionalidad de noticias. Esta funcionalidad y alguna idea inicial se han quedado fuera del alcance del proyecto.

VÍAS FUTURAS

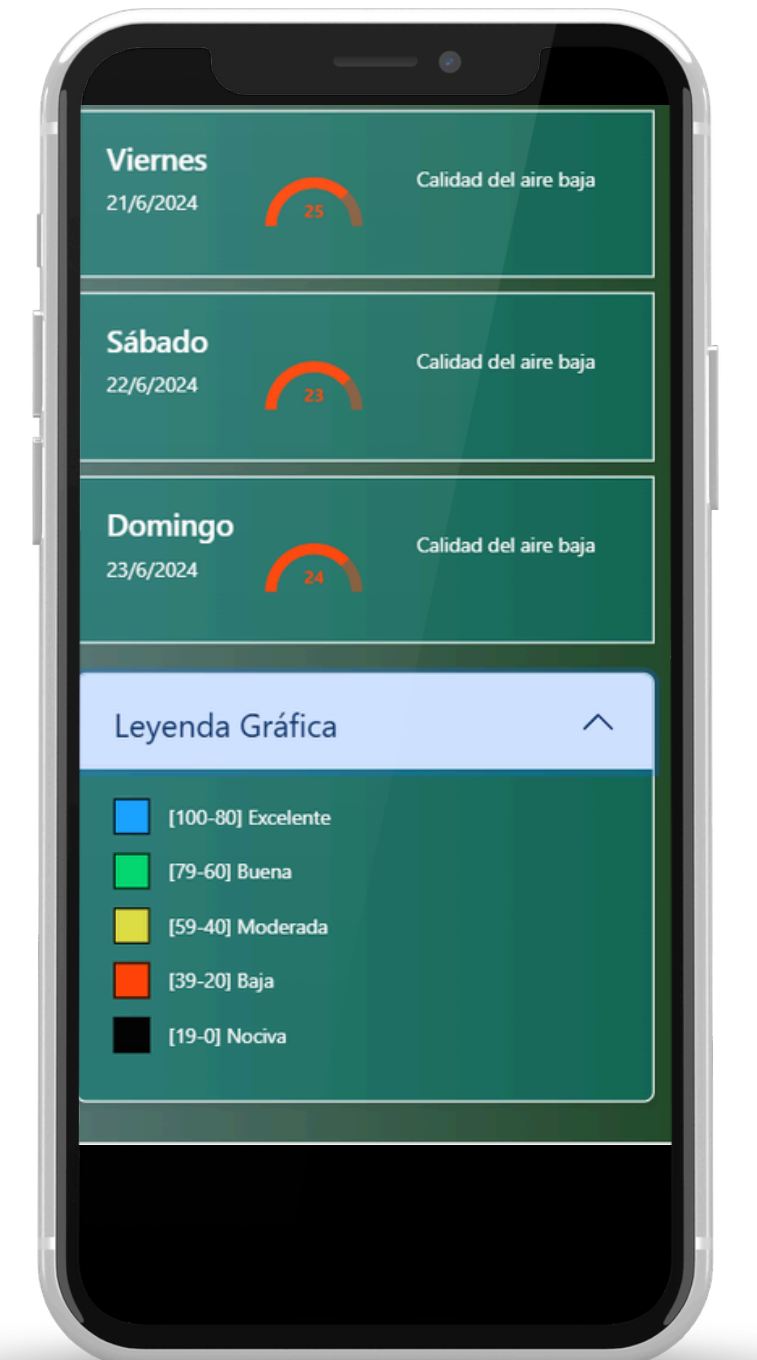
Este proyecto podría escalar a una aplicación más completa y con más funcionalidades. A continuación, se detallan algunas de las posibles vías futuras:

1. **Implementación del sistema de monitoreo de calidad del aire:** Esta sería una idea muy interesante, que consiste en dar la posibilidad a los usuarios participativos con el fin de adquirir monitores para establecerlos en sus domicilios.
2. **Mejora de la predicción:** Esta mejora perfectamente podría ser un TFG en sí mismo. Para mejorarla es necesario obtener más datos que los actuales.
3. **Obtención de datos en tiempo real:** Para poder escalar la aplicación es necesario modificar el método de la obtención de datos a una versión de pago o propia.



MUCHAS GRACIAS

Juan Bernal Miralles



ANÁLISIS DE OBJETIVOS Y PLAN DE TRABAJO

Análisis de objetivos

El proyecto se ha dividido en 12 objetivos principales. Estos a su vez se agrupan en 7 fases con 27 sub-tareas totales.

ListCalendarGanttTableTableroPanel+ Vista

Grupo: EstadoSubtareas: Expandir todoColumnasFiltrosMostrar elementos cerradosOcultar

REVISIÓN3+ Añadir Tarea

Nombre	Fecha límite	Estado	
▶ 🔗 Análisis del sistema y planteamiento de requisitos. 🔗 3	mar. 6	REVISIÓN	...
▶ 🔗 Obtención y estudio de API para la calidad del aire. 🔗 2	feb. 18	REVISIÓN	...
▶ 🔗 Investigacion y estudio de la tecnología FIWARE. 🔗 2	mar. 28	REVISIÓN	...
+ Añadir Tarea			

DESARROLLADA3+ Añadir Tarea

Nombre	Fecha límite	Estado	
▶ 🔗 Estudio de tecnologías y herramientas para el desarrollo de aplicaciones multiplataforma. 🔗 4	mar. 30	DESARROLL...	...
▶ 🔗 Diseño del modelo de datos 🔗 1	mar. 29	DESARROLL...	...
▼ 🔗 Diseño de la arquitectura de la aplicación. 🔗 2	abr. 4	DESARROLL...	...
🔗 Diseñar el componente back-end del sistema a partir de servicios REST.		DESARROLL...	...
🔗 Diseñar el componente front-end del sistema siguiendo un modelo responsivo.		INICIADA	...
+ Añadir Tarea			

INICIADA2+ Añadir Tarea

Nombre	Fecha límite	Estado	
▶ 🔗 Documentación de la aplicación multiplataforma. 🔗 4	may. 31	INICIADA	...
▶ 🔗 Desarrollo del back-end. 🔗 1	abr. 29	INICIADA	...

Planificación temporal

Para la planificación se ha utilizado la herramienta Clickup. En esta se ha podido trabajar con el tablero de tareas de kanban y el diagrama de Gantt.

