**Memoria Examen Parcial 3 Ingeniería Web**

*Francisco José del Río Domínguez*

# 1 – DISEÑO DE LA BASE DE DATOS

A continuación, se detallan las principales entidades del modelo y sus relaciones, que

serán la base para la interacción de la API.

## **1.1 - Entidades y Atributos**

Las entidades elegidas son las siguientes:

### 1.1.1 - Tareas

Las tareas en las que se puede colaborar, descritas por las siguientes características:

* **responsable**: dirección de email del usuario responsable de la tarea (el que crea la tarea).
* **descripción**: título o descripción breve de la tarea (hasta 50 caracteres).
* **habilidades**: una serie de habilidades (términos) adecuadas para participar en la tarea.
* **segmentos**: duración estimada de la tarea (en segmentos de 1 hora de trabajo).

### 1.1.2 - Colaboradores

Representa a los usuarios de la aplicación, con las siguientes características:

* **email**: dirección de email del colaborador.
* **nombre**: nombre del usuario.
* **habilidades**: una serie de habilidades (términos) que posee el colaborador.

Como se ha indicado, varios colaboradores participan en el desarrollo de una tarea, cada uno de ellos llevando a cabo uno de los segmentos de la misma. Un colaborador puede participar en varias tareas. Una misma persona puede figurar como responsable de varias tareas.

Para el almacenamiento de información, utilizarás una base de datos no relacional. Por tanto, los elementos del modelo conceptual pueden dar lugar a un número mayor o menor de entidades de base de datos, con más o menos atributos, dependiendo de la base de datos elegida y el diseño creado para representarlos.

## 1.2 - URI / Credenciales

Las credenciales para añadir la conexión con la base de datos son:

**username**: 0610915314

**password**: AEUQKaus89306OeC

La URI para acceder es la siguiente:

mongodb+srv://0610915314: AEUQKaus89306OeC@parcial3.vbji9.mongodb.net/

# 2 – TECNOLOGÍAS UTILIDADAS

La base de datos que he utilizado es **MongoDB**, que es una base de datos NoSQL

orientada a documentos que almacena datos en un formato similar a JSON (BSON).

Para el desarrollo de las **APIs REST**, he empleado **Python**, más concretamente su

librería **FastAPI**.

Para el desarrollo del frontend, he utilizado **React**, un popular framework de JavaScript para la creación de interfaces de usuario dinámicas y modernas. El proyecto fue configurado con **Vite**, una herramienta rápida y ligera que acelera el proceso de desarrollo.

Para la gestión de imágenes, he empleado **Cloudinary**, una solución en la nube que facilita el almacenamiento, transformación y entrega de imágenes optimizadas. Los mapas interactivos en la aplicación se implementaron utilizando **OpenStreetMap**, una plataforma de mapeo de código abierto que proporciona datos geográficos precisos y personalizables. Finalmente, para la autenticación de usuarios, he integrado **OAuth**, un estándar de autorización que garantiza la seguridad y privacidad de los datos de acceso de los usuarios.

## 3 – Despliegue

***GitHub***: https://github.com/tiko717/parcial3Web

El despliegue del proyecto puede realizarse tanto en local como en la nube, permitiendo flexibilidad en su ejecución. A continuación, se detalla el procedimiento para ambas opciones:

### 3.1- Despliegue en local

El repositorio de GitHub contiene dos carpetas principales: **server** (para el backend) y **client** (para el frontend), ambas incluidas en el zip del campus. A continuación, se describen los pasos para ejecutar el proyecto en local:

1. **Backend**:
   * Navega a la carpeta server en tu terminal.
   * Asegúrate de que las dependencias de Python están instaladas.

*pip install -r requirements.txt*

* + Ejecuta el archivo app\_localhost.py para iniciar el servidor backend:

*python app\_localhost.py*

* + Por defecto, el backend estará disponible en el puerto **8000**.

1. **Frontend**:
   * Navega a la carpeta client en tu terminal.
   * Si la carpeta no está configurada, crea un nuevo proyecto de Vite en la carpeta deseada con el siguiente comando:

npm create vite@latest

* + Seguir los pasos e instalar las dependencias necesarias:

(Nombre client -> React -> Javascript)

*cd client*

*npm install*

*npm run dev* (Para comprobar que va bien)

* + Pasa la carpeta client a la creada e iniciar el servidor de desarrollo con:

*npm run dev*

* + Esto desplegará el frontend en un puerto asignado por Vite (por defecto, **5173**). La URL se muestra en la terminal tras ejecutar el comando.

### 3.2- Despliegue en la nube

El proyecto también puede desplegarse en la nube utilizando **Vercel**, una plataforma para el alojamiento de aplicaciones web frontend y backend.

**URL backend**: https://parcial3-web-server.vercel.app/

**URL frontend**: https://parcial3-web-client.vercel.app/

1. **Backend**:
   * El backend se encuentra desplegado en un proyecto de Vercel, accesible mediante la URL proporcionada.
2. **Frontend**:
   * El frontend se encuentra desplegado en un segundo proyecto de Vercel, vinculado al repositorio de GitHub correspondiente. Al acceder a la URL proporcionada, se puede interactuar con la interfaz de usuario del proyecto.

El despliegue en Vercel asegura una mayor disponibilidad del proyecto y elimina la necesidad de configuraciones locales complejas, facilitando el acceso y la interacción desde cualquier dispositivo con conexión a internet.