**# ¿Que es agenda?**

Agenda es el proyecto final del curso de desarrollo visualización de datos con Python, tiene como objetivos.

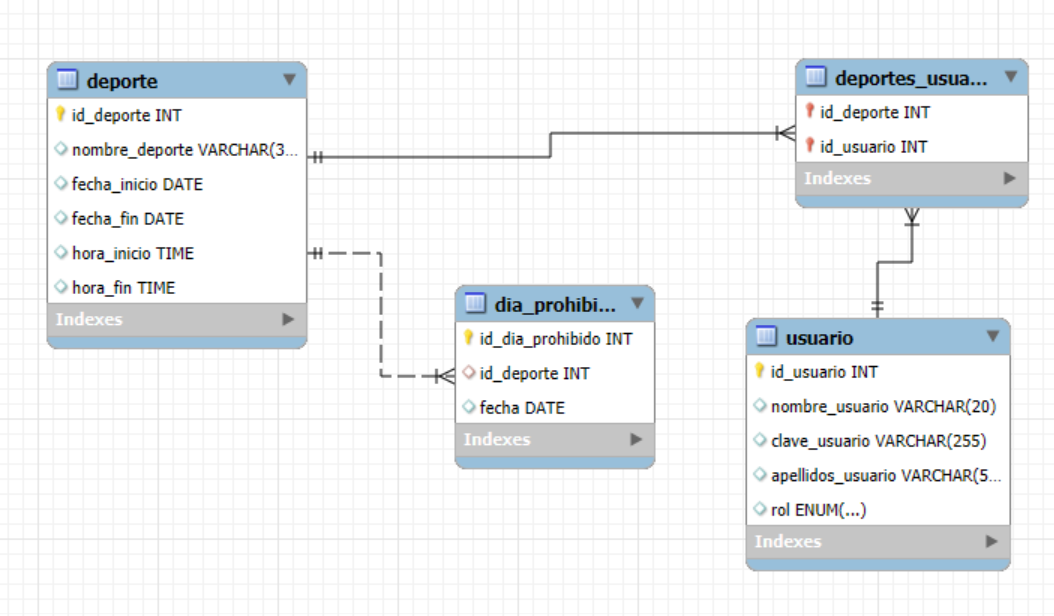
Este proyecto en github: <https://github.com/avanzatipolisto/agenda>

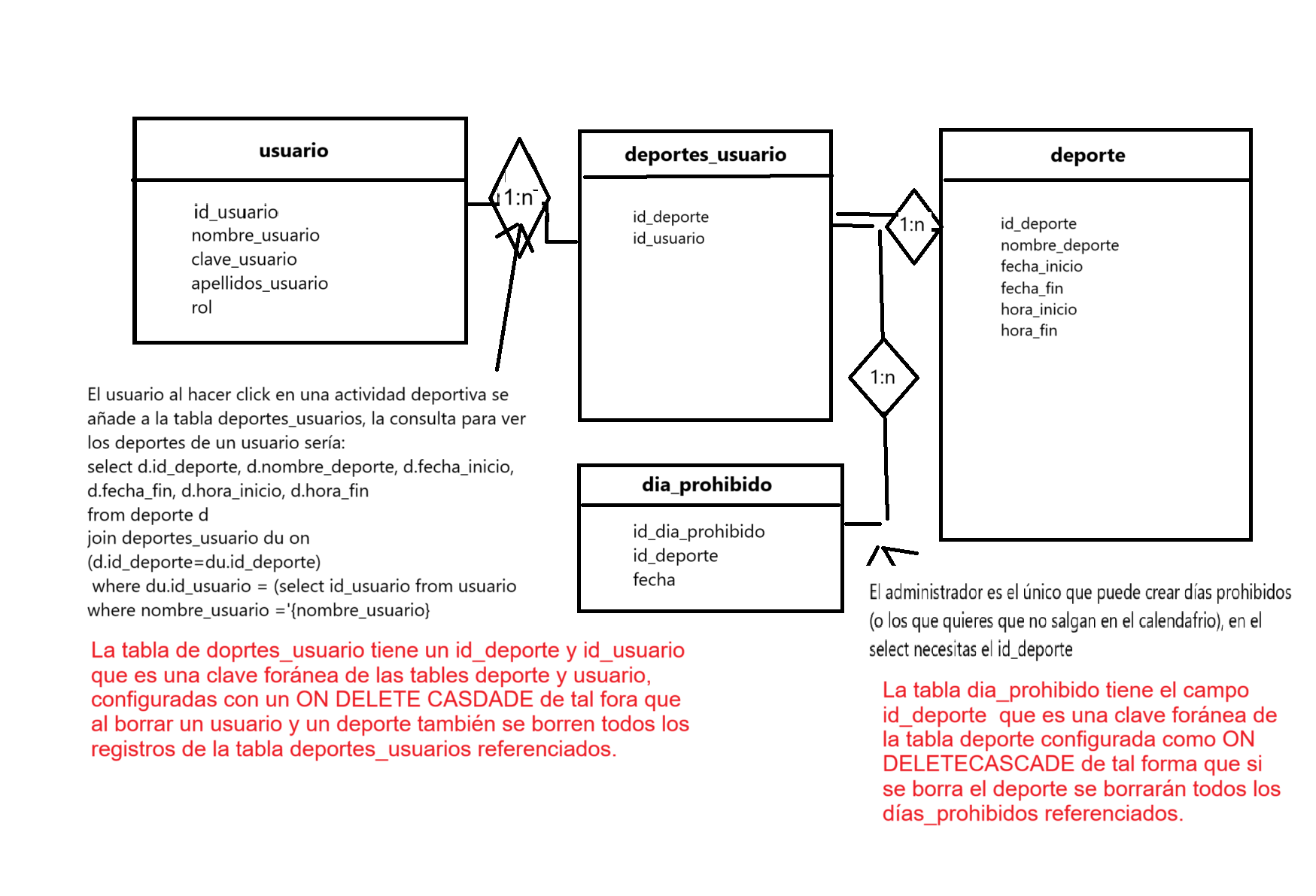
Y en render.com: <https://agenda-8s3p.onrender.com/>

Después de descargártelo, ya sabes:

1. python -m venv .venv
2. .venv\Scripts\activate
3. Pip install -r requirements.txt
4. Dentro de vscode pulsa Ctrl+Shift+p y selecciona el interprete Python (.venv)
5. Ve al main.py y pulsa el play
6. Pincha en login y accede como administrador con kike y carlos o como usuario como pablo y sergio, para todos la contraseña es 1234.

**Integración de Base de Datos: Utilizar una base de datos para almacenar y gestionar datos de forma estructurada.**

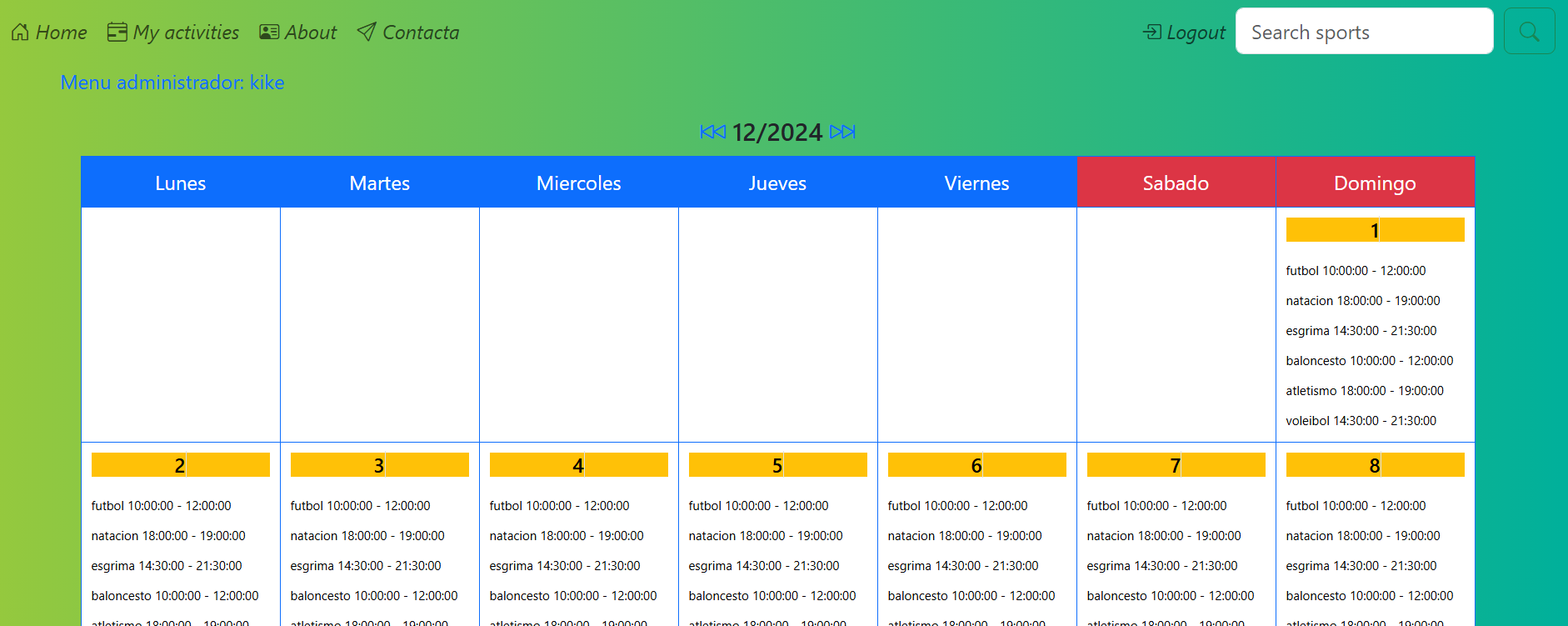




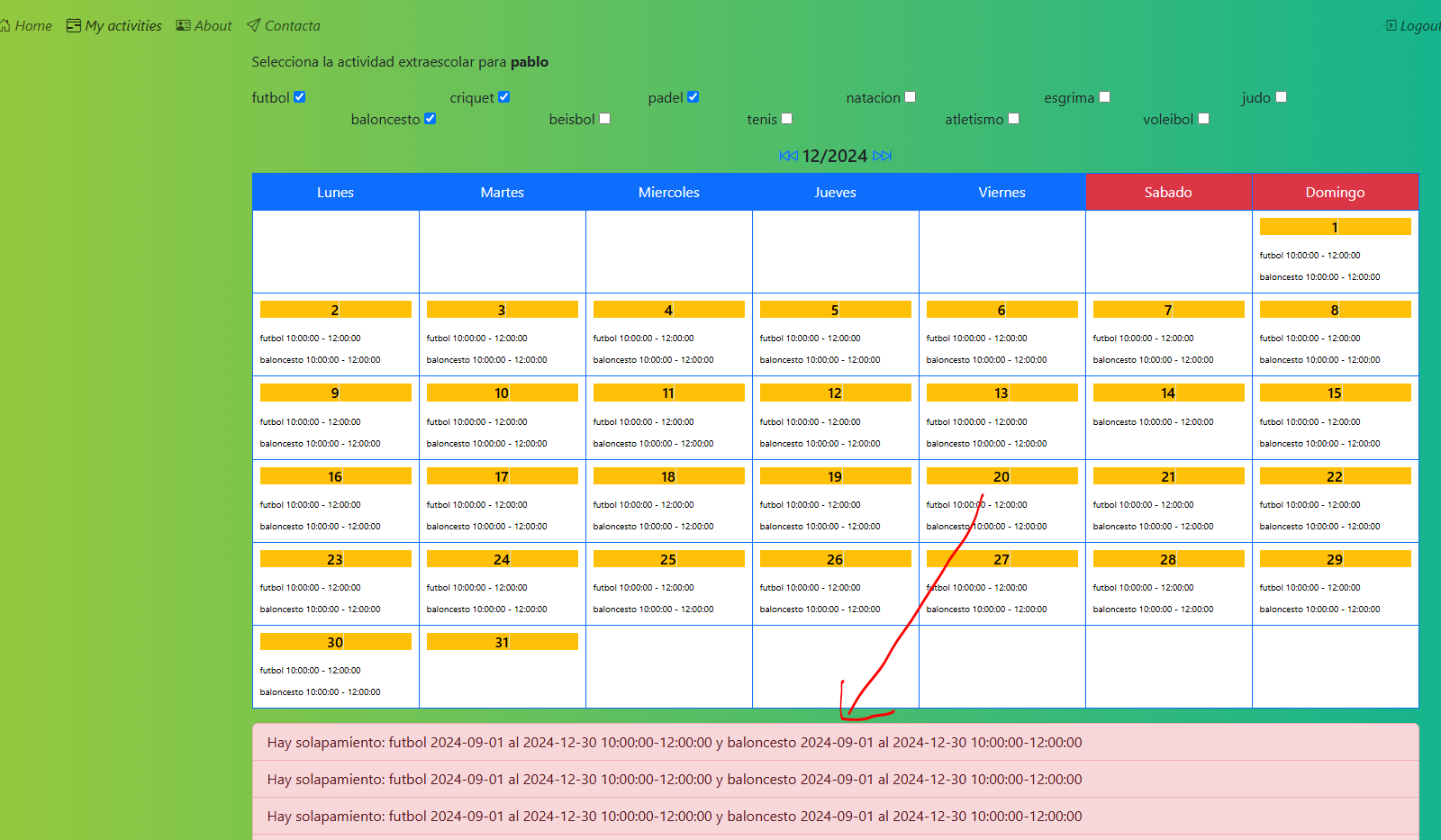
**Visualización Web: Crear una interfaz web donde se puedan visualizar los datos de manera interactiva.**

Agenda es una herramienta que te permite crear turnos para una actividad en un rango de fechas y horarios.

En la página de inicio se ven todos los turnos de las actividades mezcladas.



Cuando se logea y aparece el menú del usuario puede añadirse a sus actividades, si las actividades se superponen se le avisa.



Cada actividad tiene unos días prohibidos en los que no aparecerá en el calendario.



Puedes probarlos online en render.com: https://agenda-qptf.onrender.com

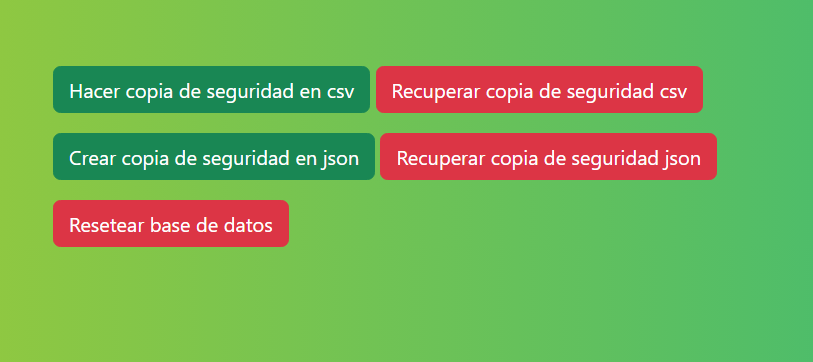
(puede ser que tarde en arrancar)

Su instancia gratuita dejará de funcionar debido a la inactividad, lo que puede retrasar las solicitudes 50 segundos o más.

Hemos trabajado con html, javascript, css (boostrap), flask, sesiones, formularios y validaciones pero no hemos creado ningún objeto propio y en la vista tocamos la lógica la base de datos,en el siguiente proyecto trabajaremos con el Model View Contoller (MVC)

**Extra: Manipulación de Ficheros: Incluir funcionalidades para leer y escribir ficheros de datos (por ejemplo, CSV, JSON, txt) y gestionarlos de manera eficiente.**

Para trabajar con archivos hemos implementado la funcionalidad de crear copias de seguridad:



**# Instrucciones de desarrollo**

1. Instala python desde aqui: <https://www.python.org/downloads/>

2. Crea el entorno virtual con el comando: python -m venv .venv

3. Activa el entorno virtual con el comando: .\venv\Scripts\activate

4. Pone el intérprete python del entorno virtual pulsando Ctrl+shift+p y escribe python interpreter, elige el python que tenga (.venv)

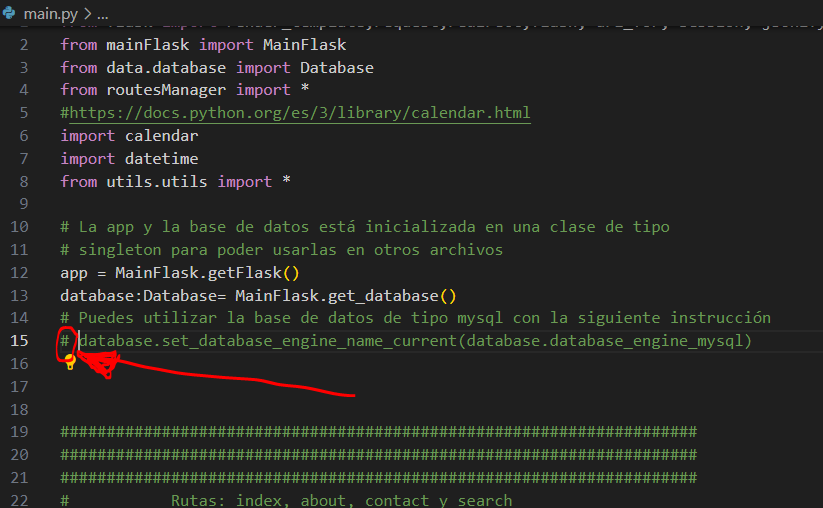
5. Dentro del entorno virtual escribe “pip install -r requirements” o:

    pip install flask

    pip install mysql-connector

6. Para probar el programa el usario kike y carlos son los administradores, pablo, loreto y sergio son los usuarios, la contraseña para todos es 1234.

7. Aunque ahora está configurado con mysql, puedes elegir trabajar con sqlite descomentádo la instrucción: database.set\_database\_engine\_name\_current(database.database\_engine\_mysql) de la línea 15 del archivo main.py:



Si seleccionas mysql tienes que descargar la versión que pesa 306.5M desde aquí: [https://dev.mysql.com/downloads/windows/installer/8.0.html](https://www.python.org/downloads/), esta versión te crea el servidor de mysql y el cliente de mysql con su usuario y contraseña de inicio de sesión.

Además, tendrás que ejecutar los comandos sql que se encuentran data/mysql-structure.sql para crear la base de datos.

También para que funciene hay que cambiar en authentication users de workbench, ejecutando en una nueva hoja SQL: ALTER USER 'yourusername'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'youpassword';

