David Escrivá

Documento de Análisis - Proyecto CRM por Consola

Solución

El objetivo del proyecto era desarrollar un sistema CRM conectando una base de datos y que por consola permitiera registrar usuarios, facturas asociadas a estos y mostrar información tanto personal, de facturas, como financiera a través de consultas. En este caso, se ha optado por una arquitectura modular, separando cada funcionalidad en un archivo independiente dentro de la carpeta `src/crm` para así hacer más fácil la escalabilidad de cada función que si estuvieran todas las funciones en un mismo archivo. Estas funcionalidades pueden ser: crear un usuario, crear una factura, mostrar ciertos datos de usuarios o facturas, etc.

Estos archivos independientes se acaban encontrando en el archivo `main.py`, que es el punto de entrada del CRM y el que muestra el menú interactivo al usuario para poder realizar las diferentes acciones. Se ha utilizado PostgreSQL como sistema de gestión de bases de datos, ya que es una opción más robusta y profesional para entornos reales.

Además, se ha utilizado Docker para contenerizar tanto la aplicación como la base de datos, lo que permite un despliegue más limpio, controlado y replicable del entorno de desarrollo.

Uno de los principales retos fue la integración de Docker, ya que inicialmente no sabía cómo usarlo para levantar una base de datos PostgreSQL. Me costó entender la sintaxis de los archivos docker-compose.yml y cómo gestionar volúmenes, puertos y dependencias entre contenedores.

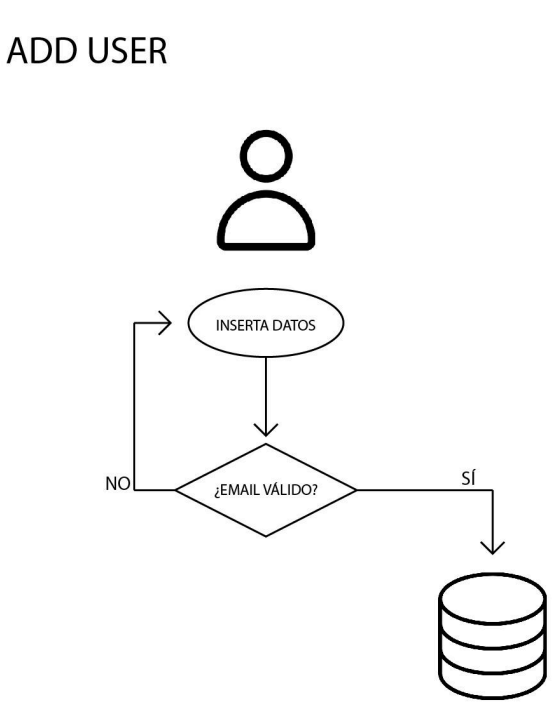
Otro reto fue la creación de tests. Aunque tenía experiencia en Python, escribir pruebas automatizadas con pytest no me resultaba del todo natural al principio. Este proyecto me ayudó a mejorar mi capacidad para testear código y refactorizar funciones de manera más mantenible.

Justificación de tipos de datos

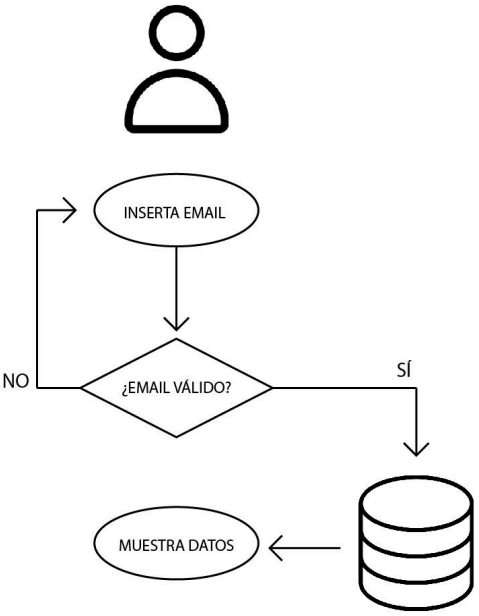
Se han utilizado los siguientes tipos de datos para las tablas de la base de datos:

- INTEGER (Serial): Utilizado en los campos id para las claves primarias, con autoincremento para identificar de forma única cada registro.
- VARCHAR: Se utiliza en campos en los que se va a insertar una cadena de caracteres con una longitud máxima. Se ha considerado que los datos como nombre, descripción, dirección… eran una buena opción para este tipo de dato.
- DATE: Se usa en los campos de fechas, asignando de manera automatica la fecha en la que se insertan los datos con NOW().
- REAL: Se emplea en el campo monto de las facturas, ya que puede incluir decimales y la documentación de SQLite recomienda usar este tipo.
- CHECK: Se ha utilizado para restringir los valores posibles del estado de una factura.

Diagramas de flujo



SHOW INVOICES



ADD INVOICE

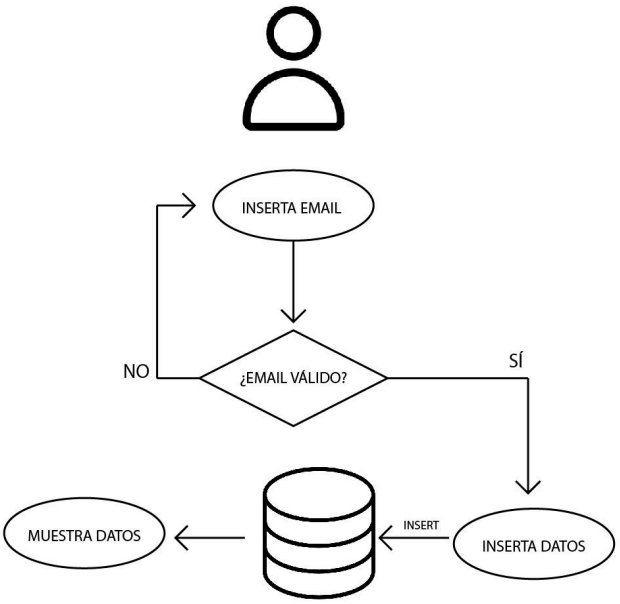


Diagrama Entidad-Relación