Universidad Nacional de Colombia - sede Bogotá
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas e Industrial
Curso: Ingeniería de Software I

Estudiantes:

Laura Vanesa Alvarez Lafont
Santiago Andres Fetecua Pulgarin
Nicolas Andres Bolanos Fernandez
Juan Pablo Corredor Castaneda

Selección de tecnología

1. Desarrollo general del proyecto

1.1. Lenguaje 🐍

El proyecto fue desarrollado en Python, un lenguaje ampliamente adoptado en el ámbito académico y profesional por su sintaxis clara, su gran ecosistema de librerías y su capacidad para facilitar un desarrollo ágil y mantenible.

Para la creación de la interfaz gráfica se emplearon las librerías Tkinter y CustomTkinter, las cuales permiten construir aplicaciones de escritorio con interfaces nativas y componentes visuales personalizables. Esta elección se debió tanto a la familiaridad previa del equipo con estas herramientas como a la experiencia adquirida en proyectos anteriores de aplicaciones locales, lo que optimizó significativamente los tiempos de desarrollo.

Antes de optar por Tkinter, se evaluó el uso de Django, una opción robusta para aplicaciones web. Sin embargo, dado que el proyecto estaba orientado específicamente a ejecutarse de forma local y en un entorno de escritorio, se descartó el desarrollo web.

1.2. Herramientas complementarias 🧰

Para garantizar la funcionalidad completa del sistema y mantener un alto estándar de seguridad y modularidad, se integraron las siguientes librerías y herramientas:

- **StarUML:** utilizada para modelar y documentar los casos de uso, contribuyendo a la claridad en la definición de los requisitos funcionales y en la comunicación técnica dentro del equipo.

- **sqlite3:** utilizado como motor para la creación de la base de datos y el manejo externo a la aplicación de los datos.

- **SQLAlchemy:** Esta herramienta de ORM brinda soporte para manejar la información contenida en la base de datos sin depender del motor (en nuestro caso, SQLite)

- **Pillow:** librería destinada al procesamiento y manipulación de imágenes, que permitió incorporar funcionalidades gráficas adicionales sin comprometer el rendimiento.

- **bcrypt:** utilizada para la gestión segura de contraseñas mediante técnicas de hash, garantizando un almacenamiento robusto y reduciendo el riesgo de exposición de credenciales.

- **sib_api_v3_sdk (SendinBlue):** implementada para el envío automatizado de correos electrónicos, incluyendo confirmaciones y notificaciones a los usuarios finales.

- **python-dotenv:** empleada para el manejo de variables de entorno, permitiendo separar la configuración sensible del código fuente y facilitar la adaptabilidad y el mantenimiento del proyecto en diferentes entornos locales.

El uso de estas herramientas contribuyó a mantener un desarrollo organizado y controlado, siguiendo buenas prácticas como el versionamiento de código y la configuración centralizada.

2. Base de datos 💾

La base de datos se diseñó mediante **MySQL Workbench**, herramienta que proporciona una interfaz visual intuitiva para el modelado de datos y la definición del esquema relacional, mediante herramientas como la selección adecuada de tipos de datos, de las propiedades de los campos y de las relaciones entre tablas.

Aprovechando las facilidades que brinda esta herramienta, se elaboró un diagrama entidad-relación (ER) detallado que permitió identificar de manera precisa las entidades, atributos y relaciones necesarias para satisfacer los requerimientos funcionales definidos en la documentación del proyecto. Posteriormente, mediante la función de Forward Engineering, se generó el código SQL de manera automática, asegurando consistencia entre el diseño conceptual y la implementación física de la base de datos. A partir de ese código orientado a Mysql se utilizó la herramienta de inteligencia artificial chat GPT para adaptarlo a **SQLite**, que es la herramienta que va a ser usada para la base de datos del proyecto.

Por último, se diseñaron modelos con **SQLAlchemy** para separar la lógica del motor de la base. Cada tabla diseñada en sqlite corresponde a una clase del ORM capaz de reflejar las operaciones que le apliquen al archivo .db del proyecto.