Universidad Nacional de Colombia - Sede Bogotá

Ingeniería de Software I

11/07/2025

Módulo: CLEAN CODE

Integrantes

- Derian Santiago Rojas Leiva
- Jhon Edison Prieto Artunduaga
- Juan David Castañeda Cárdenas
- Omar Darío Zambrano Galindo

Código Justificación:

Para la versión inicial de Walletive elegimos Python 3.12 junto con PyQt como tecnología principal de interfaz. Esta decisión nos brindó rapidez de creación considerando la facilidad de adopción por todo el equipo. Python se ajustó a nosotros porque ya lo manejábamos en otros trabajos y resultó sencillo de leer, probar y depurar; su sintaxis clara aceleró el desarrollo. En esencia, buscamos utilizar este lenguaje para implementar la lógica interna de manejo de datos internos sin uso de APIs externas. Por otro lado, PyQt ofreció ventanas para diseñar interfaces a aplicaciones de ordenador de manera sencilla y efectiva, ya que, según nuestra investigación, este Framework (Qt) es el más relevante en la actualidad para su ámbito.

Ilustración 1: PyQT.

Como almacén de datos optamos por SQLite (Ilustración 2). Este motor nos permitió guardar transacciones, metas y preferencias en un único archivo (walletive.db) sin configurar un servidor. La elección simplificó la instalación en los equipos de clase y evitó costes de mantenimiento. Dejamos planificada una migración futura a PostgreSQL para soportar

multiusuario y análisis más pesados; el esquema actual emplea tipos compatibles con ese motor, de modo que el cambio exigirá poco esfuerzo.

Ilustración 2: SQLite.

Para completar el entorno de desarrollo usamos:

- Creamos un venv para aislar las dependencias y empleamos pipreqs para mantener un requirements.txt fiable sin añadir paquetes innecesarios.
- Para las gráficas iniciales utilizamos Matplotlib y Pandas; además integramos PyQtGraph para incrustar visualizaciones en tiempo real directamente dentro de las ventanas PyQt.
- Gestionamos el código con Git, trabajando en ramas de característica revisadas mediante *pull requests* en GitHub. Una tubería en GitHub Actions crea el venv, ejecuta las pruebas y empaqueta la aplicación con PyInstaller en cada *push* a main.

En conjunto, estas elecciones cubrieron la necesidad del proyecto, se alinearon con la experiencia del grupo y respetaron los objetivos del curso de producir un prototipo funcional dentro del calendario académico.