Documento de Pruebas Unitarias Ecuplot Web

Julian Alejandro Cardenas Santiago, Carlos Mateo Cruz Ibarra

Universidad de Cundinamarca, Facultad de Ingeniería

Ingeniería de Sistemas y Computación

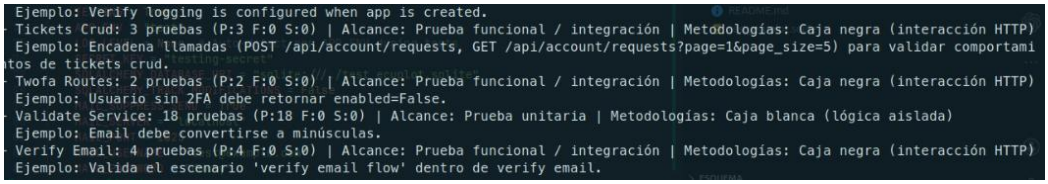
Ingeniería de Software 2

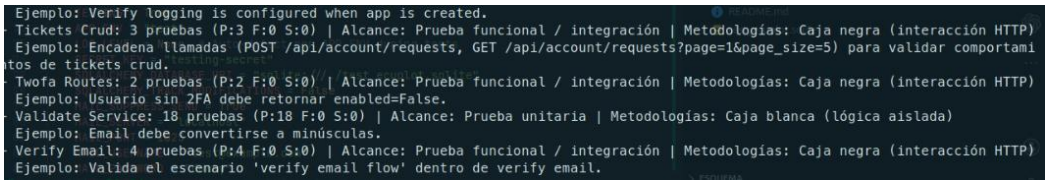
Ing. German Ortiz Diaz

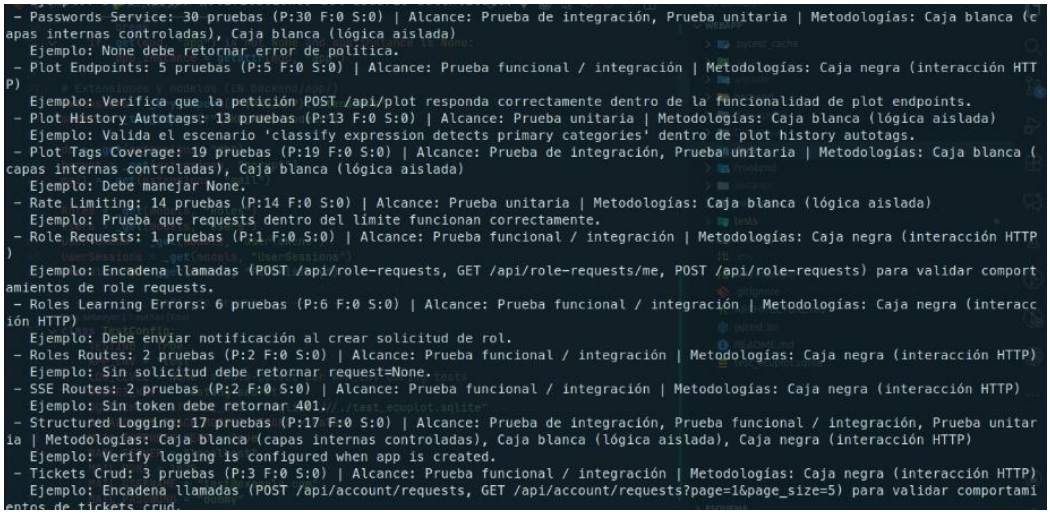
19 de noviembre del 2025

Introducción

En el proyecto EcuPlot Web se utiliza el framework pytest como herramienta principal para la ejecución de pruebas automatizadas. Pytest permite definir casos de prueba simples y legibles en Python, agruparlos por módulos (por ejemplo: servicios de contraseñas, endpoints de graficación, rutas de roles, validación de email y 2FA, exportación CSV/JSON, sistema de tickets, historial de gráficas, etc.) y ejecutarlos de forma masiva desde la línea de comandos, obteniendo un resumen claro de cuántas pruebas pasan, fallan o se omiten.







En la batería ejecutada se observan distintos tipos de pruebas y metodologías:

Pruebas unitarias sobre lógica aislada (caja blanca: capas internas controladas y lógica de validación).

Pruebas funcionales e integración sobre la API HTTP (caja negra: interacción real con endpoints como /api/plot, /api/role-requests, /api/account/requests, exportaciones, rutas de frontend y health checks).

Casos específicos para seguridad y autenticación (Passwords_Service, Twofa Routes, Validate_Service, Verify Email), control de tráfico (Rate Limiting) y consistencia de datos (History Mutations, Plot History Autoguess, Event Stream, Structured Logging).

Análisis de resultados

Los resúmenes mostrados para cada módulo indican patrones del tipo:

X pruebas (P:X F:0 S:0)

es decir:

P: número de pruebas pasadas.

F: número de pruebas fallidas (0).

S: número de pruebas saltadas (0).

El hecho de que en todos los servicios listados el valor de F sea 0 y S sea 0 significa que **todas las pruebas definidas se ejecutaron y pasaron correctamente**. Esto sugiere:

La lógica crítica de negocio (generación de gráficas, manejo de historiales, solicitudes de roles, tickets, etc.) se comporta según lo esperado en los escenarios probados.

Los mecanismos de seguridad (gestión de contraseñas, verificación de email, validación de 2FA) funcionan sin errores en las condiciones definidas en los casos de prueba.

La API HTTP y las rutas del frontend responden correctamente a las peticiones encadenadas que se usan en el flujo real de la aplicación (por ejemplo, creación de solicitudes, paginación de historiales, exportación de datos).

El logging estructurado y los servicios de monitoreo básicos se crean y configuran adecuadamente al iniciar la aplicación, lo que facilita depuración y trazabilidad.

Conclusiones

estos resultados muestran un estado estable de EcuPlot Web en el momento de la ejecución de las pruebas, con una buena cobertura sobre las áreas más sensibles (seguridad, endpoints de gráficas, gestión de cuentas y registros de actividad). Como trabajo futuro, se recomienda seguir ampliando la batería de pruebas y complementar estos resultados con métricas de cobertura para asegurar que nuevos cambios en el código mantengan el mismo nivel de confiabilidad