

UNIVERSIDAD DEL VALLE DE GUATEMALA

Ingeniería de Software – Sección 21

Ing. Pablo Barreno



Tarea 5 – Pruebas automatizadas

Osman Emanuel de León García – 23428

Milton Giovanni Polanco Serrano - 23471

Gadiel Amir Ocaña Véliz - 231270

Guatemala, 22 de agosto de 2025

Herramientas o frameworks existentes

En el desarrollo de nuestro proyecto trabajamos principalmente con React en el frontend y Node.js con Supabase en el backend. Investigamos diversas herramientas de pruebas automatizadas aplicables a nuestro entorno:

- **Jest:** Framework de testing en JavaScript mantenido por Meta, con soporte para pruebas unitarias, integración y regresión. Muy utilizado por la comunidad.
- **React Testing Library (RTL):** Librería que complementa Jest, enfocada en probar componentes de React desde la perspectiva del usuario.
- **Supertest:** Permite automatizar pruebas de integración sobre APIs en Node.js, simulando peticiones HTTP sin necesidad de levantar un servidor completo.
- **Cypress:** Framework especializado en pruebas end-to-end y de integración, muy útil para validar el flujo completo de la aplicación en el navegador.
- **Postman/Newman:** Usados comúnmente para pruebas de endpoints en backend, pero con menor integración con CI/CD que Cypress o Jest.

Herramienta seleccionada

Como grupo decidimos utilizar Jest + Supertest para las pruebas de integración en el backend y Cypress para el frontend.

Las razones de esta elección fueron:

- En el backend, Jest es robusto y ampliamente documentado, mientras que Supertest facilita el envío de peticiones a nuestras rutas sin levantar toda la aplicación. Esto nos permitió validar endpoints, middleware, seguridad y rendimiento de manera eficiente.
- En el frontend, Cypress nos ofreció un enfoque práctico para validar la aplicación en el navegador, incluyendo navegación, formularios, responsive design y manejo de errores de red. Además, su sintaxis es clara y su comunidad muy activa.

Con esta combinación logramos cubrir pruebas unitarias, de integración, funcionales y end-to-end de forma integrada en nuestro flujo de desarrollo.

Conclusiones

Como grupo concluimos que la combinación de Cypress en el frontend y Jest + Supertest en el backend nos permitió cubrir de forma completa las pruebas de integración y funcionales en el proyecto MoneyFlow. Cypress resultó muy útil para validar la aplicación desde la perspectiva del usuario, incluyendo navegación, formularios, diseño responsivo y

manejo de errores, mientras que Jest y Supertest nos dieron confianza en la solidez de la API al comprobar rutas, validaciones, middleware, seguridad y rendimiento. Aunque la configuración inicial de Cypress y la definición de pruebas en Supertest nos tomaron tiempo, luego el desarrollo de pruebas se volvió más ágil y nos permitió detectar errores con rapidez. Los principales retos fueron aprender a usar interceptores en Cypress para simular fallos de red y configurar correctamente la validación en el backend, pero logramos resolverlos con documentación y pruebas iterativas. En general, el esfuerzo invertido valió la pena, ya que ahora contamos con un conjunto robusto de pruebas automatizadas que aseguran la estabilidad y calidad tanto del frontend como del backend.