

Informe — Pruebas de integración

Proyecto: TutorVirtual (backend + frontend)

Fecha: 2025-11-23

Resumen

- Se ejecutaron y estabilizaron las pruebas de integración UI→API (fullstack) y las suites de API.
- Cobertura backend (API) en ejecución dedicada: Statements 99.17%, Lines 100% (branches pendientes en report/pdf).
- El test de integración principal src/integration/fullstack.int.test.js valida el flujo login demo + POST /api/history desde la UI simulada.
- Recomendación para una métrica formal de “interacciones >70%”: añadir 1–2 E2E (Cypress/Playwright) que recorran UI→API→DB en un navegador real o tests de contrato para APIs externas.

Evidencia numérica (último run backend)

- Suites: 13/13 pasadas (30 tests).
- Cobertura API: Statements 99.17%, Branches 66.07%, Functions 95.45%, Lines 100%. Ramas sin cubrir: report/pdf/route.js (líneas 15,20,32,42-44,50,105-114) y ramas de entorno en analyze/route.js (líneas 5,36).
- Fullstack integration: fullstack.int.test.js pasada (flujo UI→API en modo demo con mocks de IA/Mongo).

Casos y edge cases cubiertos en integración/API

- Auth demo: credenciales incorrectas → 401; token válido en /me; token expirado.
- History: modo demo y Mongo (GET/POST correctos, validaciones).
- Analyze: falta de GEMINI_API_KEY → 503; modo demo con fallback; error de IA mockeado devolviendo fallacias.
- Admin: /api/admin/history y /api/admin/users con clave válida (200) y sin clave (401); paginación y resolución por email.
- Reportes: /api/report (TXT attachment) y /api/report/pdf (PDF con texto, falacias, preguntas y respuestas).
- Fullstack bridge: login demo y POST /api/history desde componente QuestionsDisplay.

Mocking y aislamiento en integraciones

- MongoDB: mocks de getDb en admin; mongodb-memory-server en tests de Mongo (auth/history).
- IA: mocks de detectLogicalFallacies para simular éxito, falta de clave y error.
- Reportes: generación local sin servicios externos.
- Fullstack: fetch bridge que enruta a handlers Next sin levantar servidor real.

Recomendaciones para “10 filas verdes más” / mayor cobertura de integración

- Cubrir ramas faltantes en report/pdf/route.js con tests adicionales (inputs vacíos, fallacies con explicación, answers sin options, timestamp ausente).
- Añadir 1–2 pruebas E2E (Cypress/Playwright) sobre flujos críticos:
 1. Registro/login → análisis → guardar historial → descargar PDF.
 2. Ruta de admin (GET history/users) con ADMIN_KEY en entorno real.
- Opcional: tests de contrato o E2E que midan interacciones UI→API→DB y reporten esa métrica.
- Silenciar console.error esperado en tests de IA si quieres una salida más limpia (se genera al mockear error en /api/analyze).

Anexo

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	99.17	66.07	95.45	100	
admin/history	100	100	100	100	
route.js	100	100	100	100	
admin/users	100	100	100	100	
route.js	100	100	100	100	
analyze	100	84.61	100	100	
route.js	100	84.61	100	100	5,36
auth/login	100	100	100	100	
route.js	100	100	100	100	
auth/logout	100	100	100	100	
route.js	100	100	100	100	
auth/me	100	100	100	100	
route.js	100	100	100	100	
auth/signup	100	100	100	100	
route.js	100	100	100	100	
history	100	100	100	100	
route.js	100	100	100	100	
report	100	100	100	100	
route.js	100	100	100	100	
report/pdf	98.61	60.46	91.66	100	
route.js	98.61	60.46	91.66	100	15,20,32,42-44,50,105-114
Test Suites: 13 passed, 13 total					
Tests: 34 passed, 34 total					
Snapshots: 0 total					
Time: 8.556 s, estimated 18 s					
Ran all test suites matching /src\\app\\api/i.					