



EDUARDO GALLIFA

Ciudad Madero, Tamaulipas, México · +52 833 121 0691

eduardogallifao@gmail.com

<http://www.github.com/eduardogallifaochoa>

<http://www.linkedin.com/in/eduardogallifaochoa/>

PROFILE

QA Engineer & AI Automation Consultant. I build practical, production-style test systems: Playwright + Pytest for UI, FastAPI + TestClient for API, and OpenAPI contract & fuzz testing with Schemathesis—wired into CI/CD (GitHub Actions + Docker) with quality gates, HTML reports, and coverage. Comfortable with both Automation and Manual QA (exploratory, test design, bug triage). I openly use AI to speed up test creation, code, and reporting—clear outcomes for non-technical stakeholders.

CORE SKILLS

- **UI:** Playwright (Chromium/Firefox/WebKit), selectors, fixtures, parallel runs, HTML reports
- **API:** FastAPI, Pydantic, HTTPX, Starlette TestClient, OpenAPI 3.1, contract & fuzz (Schemathesis/Hypothesis)
- **CI/CD & DevOps:** GitHub Actions, Docker & Docker Compose, Uvicorn, Nginx, Codecov
- **Quality & Security:** Pytest, coverage, Ruff (lint/format), Bandit (SAST), pip-audit (dependency vulns)
- **Scripting & Data:** Python, JavaScript, Bash; SQL (QA queries)
- **Manual QA:** exploratory testing, boundary/edge cases, test cases, bug reporting (Jira/TestRail)
- **Practices:** STLC/SDLC, Agile (Scrum/Kanban), requirements-to-tests traceability

- **Languages:** Spanish (Native), English (B2–C1 Cambridge)

HIGHLIGHT PROJECT — “QA Automation Site” (2025)

github.com/eduardogallifaochoa/qa-automation-site

- **Stack:** Static Nginx frontend (index/login/contact) + FastAPI backend (login/contact) with OpenAPI 3.1 at /openapi.json, CORS & security headers.
- **UI Automation (Playwright + Pytest):** positive/negative/edge cases for login & contact; reusable fixtures; seeded runs; HTML report.
- **API Tests (Pytest TestClient):** unit/integration for /api/* with input validation & error paths.
- **Contract & Fuzz:** Schemathesis CLI against OpenAPI 3.1; reproducible seeds; CLI wrapper via Pytest.
- **CI/CD:**
 - test.yml: Docker Compose (frontend + backend), readiness checks, UI + API tests, upload report.html, push coverage to Codecov.
 - fuzz.yml: boot FastAPI (Uvicorn), run Schemathesis (--checks all, --max-examples), then Pytest wrapper.
- **Quality Gates:** pipeline blocks merges if UI/API/fuzz tests fail or if lint (Ruff), security (Bandit/pip-audit), or coverage thresholds are not met.
- **Tooling:** browser caching in CI, deterministic fuzzing, clean fixtures, and business-friendly reports.

EXPERIENCE

QA Engineer / AI Automation Consultant — Freelance (2024–Present)

- Designed UI/API test suites and CI/CD pipelines with quality gates for safer releases.
- Introduced AI workflows (structured prompting) to generate/expand cases, scripts, and analyze failures.
- Reduced flakiness (stability controls), curated test data, and led bug triage/prioritization with product.

EDUCATION

B.Eng. Industrial Engineering — Universidad Latinoamericana (2020–2024)

Certifications: Cambridge English (B2–C1)

SELECTED TOOLS

Playwright, Pytest, FastAPI, HTTPX, Pydantic, Schemathesis/Hypothesis, Docker, GitHub Actions, Uvicorn, Nginx, Ruff, Bandit, pip-audit, Codecov, Jira, TestRail

AVAILABILITY

Open to Automation QA, Manual QA, and short-term consulting (quality gates, test architecture, release safety with AI).