

Claude Code

aplicado al

Testing

(ES/EN)

Descripción general

Overview

Este workshop de 4 sesiones en vivo enseña a integrar Claude Code en cada fase del ciclo de testing: desde la exploración manual asistida por IA hasta la automatización completa con pipelines CI/CD y flujos agénticos con MCP servers.

This 4-session live workshop teaches how to integrate Claude Code into every phase of the testing cycle: from AI-assisted manual exploration to full automation with CI/CD pipelines and agentic flows with MCP servers.

#	Fecha / Date	Hora / Time	Tema / Topic
01	29 mayo / May 29, 2026	22:00 h (España/Spain)	Fundamentos + Testing Manual con IA
02	5 junio / Jun 5, 2026	22:00 h (España/Spain)	Unit Tests, TDD e Integración
03	12 junio / Jun 12, 2026	22:00 h (España/Spain)	E2E, API Testing y CI/CD
04	19 junio / Jun 19, 2026	22:00 h (España/Spain)	MCP Servers, Flujos Oficiales y Demo Day

■ Primera sesión / First session	Viernes 29 mayo 2026	Friday, May 29, 2026
■ Horario / Schedule	22:00 h hora de España	Fridays 10:00 pm Spain time
■ Modalidad / Format	En vivo por Zoom	Live on Zoom
■ Duración / Duration	4 viernes · 2–3 horas c/u	4 Fridays · 2–3 hours each
■ Inversión / Investment	199 € (≈50 € por sesión)	€199 (≈€50 per session)
■ Certificado / Certificate	Sí, con QR de verificación	Yes, with verification QR

Fundamentos y exploración asistida por IA

Fundamentals & AI-assisted exploration

ESPAÑOL	ENGLISH
<ul style="list-style-type: none">• Instalación de Claude Code CLI y configuración del entorno• Modelo mental: Claude como pair tester, no como sustituto• Prompt engineering para QA: plantillas y patrones• Generación de test charters desde user stories o tickets• Casos de prueba en formato Gherkin (Given/When/Then)• Bug reports estructurados generados automáticamente• Priorización de casos por riesgo/impacto con Claude• Integración con TestRail, Xray o Notion• Flujo de revisión: cómo auditar lo que genera la IA	<ul style="list-style-type: none">• Claude Code CLI installation and environment setup• Mental model: Claude as pair tester, not replacement• Prompt engineering for QA: templates and patterns• Test charter generation from user stories or tickets• Gherkin test cases (Given/When/Then format)• Automatically generated structured bug reports• Risk/impact prioritization of cases with Claude• Integration with TestRail, Xray or Notion• Review workflow: how to audit what AI generates

ENTREGABLE / DELIVERABLE	Suite de 30+ casos de prueba generados con Claude Code, bug report plantilla y test charter documentado.	30+ test cases generated with Claude Code, a bug report template and a documented test charter.
---------------------------------	--	---

Unit tests, TDD e integración

Unit tests, TDD & integration

ESPAÑOL

- Generar unit tests desde código: Jest (JS/TS), pytest (Python), JUnit (Java)
- Cobertura inteligente: ramas no cubiertas y edge cases
- Mocking y stubbing generados automáticamente
- Ciclo TDD Red-Green-Refactor acelerado con Claude
- Pair programming: Claude sugiere, el QA valida y decide
- Detección de over-mocking y anti-patterns en tests
- Integration tests para servicios REST (supertest, httpx)
- Contract testing con Pact: contratos desde documentación
- Mutation testing con Stryker/mutmut asistido por IA

ENGLISH

- Generate unit tests from code: Jest (JS/TS), pytest (Python), JUnit (Java)
- Intelligent coverage: uncovered branches and edge cases
- Automatically generated mocks and stubs
- Accelerated TDD Red-Green-Refactor cycle with Claude
- Pair programming: Claude suggests, QA validates and decides
- Detection of over-mocking and test anti-patterns
- Integration tests for REST services (supertest, httpx)
- Contract testing with Pact: contracts from documentation
- Mutation testing with Stryker/mutmut assisted by AI

ENTREGABLE / DELIVERABLE

Suite de unit + integration tests con ≥80% de cobertura de las funciones core. Un ciclo TDD documentado.

Unit + integration test suite with ≥80% coverage of core functions. A documented TDD cycle.

E2E, API testing y pipelines CI/CD

E2E, API testing & CI/CD pipelines

ESPAÑOL

- Generar page objects y tests E2E desde capturas o texto
- Playwright: autenticación, formularios, navegación
- Claude Code como debugger: diagnóstico de selectores rotos
- Visual regression testing con screenshots de Playwright
- Colecciones Postman/Bruno desde OpenAPI spec con Claude
- Performance testing: scripts k6 generados por Claude
- Security testing de APIs: OWASP top 10 asistido
- GitHub Actions: workflow YAML completo generado por Claude
- Análisis de fallos de CI y diagnóstico de flaky tests

ENGLISH

- Generate page objects and E2E tests from screenshots or text
- Playwright: authentication, forms, multi-page navigation
- Claude Code as debugger: broken selector diagnosis
- Visual regression testing with Playwright screenshots
- Postman/Bruno collections from OpenAPI spec with Claude
- Performance testing: k6 scripts generated by Claude
- API security testing: OWASP top 10 assisted
- GitHub Actions: full YAML workflow generated by Claude
- CI failure analysis and flaky test diagnosis

ENTREGABLE / DELIVERABLE

Pipeline CI/CD funcional en GitHub con smoke tests E2E, suite de API tests y reportes automáticos.

Functional CI/CD pipeline in GitHub Actions with E2E smoke tests, API test suite and automatic reports.

MCP servers, flujos oficiales y Demo Day

MCP servers, official workflows & Demo Day

ESPAÑOL

- CLAUDE.md: el archivo de contexto para proyectos QA
- Modo agente: Claude ejecuta, testea y corrige sin intervención
- Slash commands personalizados para flujos QA repetitivos
- MCP servers: qué son y por qué transforman el testing
- MCP + Jira/Linear: bugs automáticos desde fallos de test
- MCP + GitHub: PRs, comentarios e issues automatizados
- MCP + bases de datos: validación de datos de test en DB
- Seguridad: qué no enviarle nunca a Claude (datos de prod)
- Demo Day: presentación del proyecto final (10 min c/u)

ENGLISH

- CLAUDE.md: the context file for QA projects
- Agent mode: Claude runs, tests and fixes without intervention
- Custom slash commands for repetitive QA flows
- MCP servers: what they are and why they transform testing
- MCP + Jira/Linear: automatic bugs from test failures
- MCP + GitHub: automated PRs, comments and issues
- MCP + databases: test data validation directly in DB
- Security: what to never send to Claude (production data)
- Demo Day: final project presentation (10 min each)

ENTREGABLE / DELIVERABLE

Suite de testing completa: CLAUDE.md documentado, al menos un flujo MCP funcionando y pipeline CI/CD activo.

Complete testing suite: documented CLAUDE.md, at least one working MCP flow and active CI/CD pipeline.

¿Para quién es este workshop?

Who is this workshop for?

✓	Conocimientos básicos de testing de software	Basic software testing knowledge
✓	Programación básica en Python, JS o Java	Basic programming in Python, JS or Java
✓	Experiencia básica con Git / GitHub	Basic experience with Git / GitHub
✓	Laptop con 8 GB de RAM o más	Laptop with 8 GB RAM or more
✓	Cuenta de Claude Pro o API key de Anthropic	Claude Pro account or Anthropic API key