

La estrategia de automatización primero

- Automatiza checks estables, repetibles y de alto valor
- Mantenga pruebas manuales para descubrimiento y juicio de UX
- Trate la automatización como un código de producto de ingeniería

Referencias: <https://martinfowler.com/articles/practical-test-pyramid.html>

Pirámide de pruebas en la práctica

- Comprobaciones de unidad/servicio más rápidas, menos UI E2E
- Las pruebas de UI cubren únicamente viajes críticos para el usuario
- Equilibre la confianza con el costo de ejecución

Referencias: <https://martinfowler.com/articles/practical-test-pyramid.html>

Mitigación de pruebas inestables

- Ponga en cuarentena pruebas no deterministas rápidamente
- Eliminar el estado compartido y las dependencias de tiempo
- Realice un seguimiento de las tendencias de confiabilidad a lo largo del tiempo

Referencias: <https://testing.googleblog.com/2016/05/flaky-tests-at-google-and-how-we.html>

Arquitectura de prueba mantenible

- Usa objetos de página/guión cuando corresponda
- Centralice fixtures y pruebe los creadores de datos
- Mantenga las assertions explícitas y legibles

Referencias: <https://playwright.dev/docs/intro> |
<https://docs.pytest.org/en/stable/>

BDD: cuándo y cuándo no

- Usa Gherkin para lograr un entendimiento compartido
- Evite convertir archivos de funciones en scripts de bajo nivel
- Prefiere escenarios legibles para el negocio

Referencias: <https://cucumber.io/docs/gherkin/reference/>

