

Pruebas de Software

Una prueba de software se define como el proceso de validación de un producto o programa para verificar si hace lo que debería de hacer, y en caso de no pasar la prueba, definir los errores, disminuir costos o mejorar el rendimiento que no permite su buen funcionamiento.

- **Pruebas locales (UNITARIAS)**
 - Son aquellas hechas dentro del dispositivo local(computadora, máquina virtual) para analizar el funcionamiento de cada función elemental del proyecto.
- **Pruebas de integración**
 - Son aquellas que permiten examinar la funcionalidad de cómo los elementos de un proyecto se relacionan entre sí. Aquí se examinan qué elementos unitarios son redundantes y si la funcionalidad es la correcta o esperada.
- **Pruebas alfa**
 - También conocidas como pruebas de campo, son utilizadas cuando el software funciona en distintos entornos de usuario, e incluso para explorar nuevos entornos que aún no ofrece el software, recomendado para un sistema pensado para un público general.
 - Las pruebas alfa son utilizadas con frecuencia para software de distribución masiva como una forma de pruebas de aceptación internas.
- **Pruebas de regresión:**
 - Es un tipo de prueba funcional donde se prueba si una aplicación sigue cumpliendo con su funcionamiento después de haber sido actualizada o modificada.
- **Pruebas dinámicas de validación**
 - Es un método de prueba que se realiza en una configuración ejecutada por código, con el fin de monitorear y verificar su funcionamiento de acuerdo a los requerimientos establecidos por el usuario y comprobar su desempeño en el “mundo real”. De tipo correctiva
- **Pruebas bajo condiciones frontera**
 - Para este tipo de pruebas generalmente se simulan sistemas grandes con el fin de modelar una pequeña parte que se encuentre “fuera de su frontera”, en otras palabras, es utilizada para modelar/simular espacios para probar el sistema/software fuera de sus fronteras habituales.

Herramientas de pruebas automatizadas

Tipo de prueba	Nombre	Licencia	Plataforma/ lenguaje	Particularidades
Unitarias	Jest	Open source	Javascript Se distribuye como un	Es el framework de test de Facebook, con el que se hacen los test de React, pero también se puede usar con otros frameworks como Vue o Angular, incluso

			paquete NPM	<p>con VanillaJs.</p> <p>Este marco de pruebas está construido sobre Jasmine y es mantenido por Meta.</p>
Integración	Test Container	Comercial	Biblioteca de java, que permite usar instancias en Docker	<p>En algunas partes del código no es posible hacer test de integración debido a que suelen tener dependencia con una base de datos.</p> <p>Soporta JUnit tests, proporcionando instancias ligeras y desechables de BD comunes, navegadores web selenium o cualquier otro que pueda ejecutarse en un contenedor Docker.</p>
Alfa	test complete	Comercial	JavaScript, VBScript, Python, o C++Script para escribir scripts de testeo.	<p>Se pueden hacer pruebas en aplicaciones web, móviles y de escritorio.</p> <p>Tiene además, inserción de puntos de control en los pasos del test para verificar los resultados.</p>
Regresión	RanorexSpy	Comercial	Ranorex Studio	<p>Los desarrolladores pueden agregar elementos IU en un repositorio y organizar la automatización de pruebas en una interfaz de arrastrar y soltar, en aplicaciones de escritorio, web o móviles.</p> <p>La plataforma es compatible con diversos sistemas y lenguajes, como Delphi, Java, C#, C++, ActiveX, Adobe Air, CefSharp, Apache Tapestry, Bootstrap y más. Las organizaciones también pueden agregar métodos de codificación a una biblioteca y analizar los resultados y fallas de las pruebas mediante informes basados en XML.</p>
Dinámicas de validación	Concordio	Open source	Framework de Java	<p>Las especificaciones (pruebas) se escriben normalmente en archivos HTML, usando tablas y todos los elementos comunes para darle formato. De esta manera se logran especificaciones muy fáciles de leer y que todos pueden comprender (desde analistas de negocio hasta desarrolladores).</p> <p>A partir del requerimiento en HTML se realizan asociaciones entre el texto y las pruebas (instrumentación del HTML), extrayendo la información valiosa para la prueba automatizada.</p> <p>Las pruebas en Concordion son pruebas</p>

				JUnit.
Bajo condiciones frontera	Lambda Test	Comercial	Selenium basado en la nube	El usuario puede realizar pruebas automatizadas entre navegadores en más de 2000 navegadores y garantizar que sus usuarios obtengan una experiencia perfecta en todos los navegadores, versiones y sistemas operativos. LambdaTest se integra con todas las herramientas líderes de gestión de proyectos como JIRA, Asana, Github, Trello, Slack, etc.

Aplicación de prueba unitaria:
conversions.test.js

```
import ChangeDate from "../changeDate";
import ChangeName from "../changeName";
import GradeToLetter from "../gradeToLetter";

describe('individual functions', () => {
  test('grade conversion', () => {
    expect(GradeToLetter(99)).toBe('A')
    expect(GradeToLetter(92)).toBe('A-')
    expect(GradeToLetter(88)).toBe('B+')
    expect(GradeToLetter(85)).toBe('B')
    expect(GradeToLetter(80)).toBe('B-')
    expect(GradeToLetter(77)).toBe('C+')
    expect(GradeToLetter(75)).toBe('C')
    expect(GradeToLetter(70)).toBe('C-')
    expect(GradeToLetter(65)).toBe('D')
    expect(GradeToLetter(50)).toBe('E')
  })

  test('change date format', () => {
    expect(ChangeDate('24/08/2022')).toBe('08/24/2022')
    expect(ChangeDate('13/02/2002')).toBe('02/13/2002')
    expect(ChangeDate('25/08/2022')).toBe('08/25/2022')
  })

  test('change name format', () => {
    expect(ChangeName('Diego Mellado Oliveros')).toBe('Diego Mellado')
    expect(ChangeName('Iwalani Amador Piaga')).toBe('Iwalani Amador')
    expect(ChangeName('Tonatiuh Reyes Huerta')).toBe('Tonatiuh Reyes')
    expect(ChangeName('Manuel López Portillo')).toBe('Manuel López')
  })
})
```

app.test.js

Resultados:

```
import { render, screen, cleanup, fireEvent } from "@testing-library/react";
// Importing the jest testing library
import '@testing-library/jest-dom'
import App from './App';

describe('App component', () =>{
  test('App rendering', () =>{
    render(<App/>)
    const table1 = screen.getByTestId("initial-table")
    const table2 = screen.getByTestId("changed-table")

    expect(table1).toBeInTheDocument();
    expect(table2).toBeInTheDocument();
  })
})
```

PASS

src/conversions.test.js

PASS

src/App.test.js

Test Suites: 2 passed, 2 total

Tests: 4 passed, 4 total

Snapshots: 0 total

Time: 3.305 s

Ran all test suites.

Watch Usage: Press w to show more.

Referencias:

- Junior, A. (2018, April 4). *¿Qué son Pruebas de regresión?: Definiciones ADICTEC*. Adictec. Retrieved August 24, 2022, from <https://adictec.com/que-son-pruebas-de-regresion/>
- Mundo Testing. (2020, February 20). *Tipos y etapas de las pruebas de software (Etapas Del Testing)*. Mundo Testing. Retrieved August 24, 2022, from https://mundotesting.com/etapas-de-pruebas-de-software-etapas-del-testing/#Pruebas_de_campo_tambien_llamadas_pruebas_alfa_o_beta
- Pruebas dinámicas: ¿Qué son las pruebas dinámicas? el: Tipos y técnicas.* ¿Qué son las pruebas dinámicas? El | Tipos y técnicas. (n.d.). Retrieved August 24, 2022, from <https://es.education-wiki.com/4220987-dynamic-testing>
- Salarios en software testing 2020 (méxico y latinoamérica)*. Cool Testers. (n.d.). Retrieved August 24, 2022, from <https://www.cooltesters.com/blog/salarios-en-software-testing-2020-m%C3%A9xico-y-latinoam%C3%A9rica>
- ¿Qué son las pruebas de software y Cómo Funcionan?* IBM. (n.d.). Retrieved August 24, 2022, from <https://www.ibm.com/ar-es/topics/software-testing>
- I. (2019, 18 febrero). Ranorex Studio. GetApp. Recuperado 25 de agosto de 2022, de <https://www.getapp.com.co/software/2043291/ranorex-studio#:~:text=Ranorex%20Studio%20es%20una%20soluci%C3%B3n%20de%20pruebas%20automatizadas,pruebas%20para%20aplicaciones%20m%C3%B3viles%2C%20web%20y%20de%20escritorio.>
- Testcontainers*. (s. f.). Testcontainer. Recuperado 25 de agosto de 2022, de <https://www.testcontainers.org/>
- P. (2020, 7 junio). *Pruebas de integración con Testcontainers, ejemplo de JPA con la base de datos PostgreSQL*. Blog Bitix. Recuperado 25 de agosto de 2022, de <https://picodotdev.github.io/blog-bitix/2020/06/pruebas-de-integracion-con-testcontainers-ejemplo-de-jpa-con-la-base-de-datos-postgresql/>
- Romero, J. F. (2020, 13 julio). *Testcontainers - Dockeriza tus tests de integración en Java*. Adictos al trabajo. Recuperado 25 de agosto de 2022, de <https://www.adictosaltrabajo.com/2020/07/13/testcontainers-dockeriza-tus-tests-de-integracion-en-java/>
- Concordion - Dos Ideas*. (s. f.). Dos Ideas. Recuperado 25 de agosto de 2022, de <https://dosideas.com/wiki/Concordion>

Global Business IT. (s. f.). *Top 10 de herramientas para Pruebas Automatizadas* - Global Business IT. Recuperado 25 de agosto de 2022, de <https://gbitcorp.com/blog/posts/top-10-herramientas-para-pruebas-automatizadas/>

K, M. (2022, 10 febrero). *Beginners Guide to Automation Testing*. AALPHA. Recuperado 25 de agosto de 2022, de <https://www.aalpha.net/blog/beginners-guide-to-automation-testing/>

Kim, M. (2015, 27 mayo). *Una introducción a Appium*. Code Envato Tuts+. Recuperado 25 de agosto de 2022, de <https://code.tutsplus.com/es/tutorials/an-introduction-to-appium--cms-23861>

Pérez, B. (2021, 14 julio). *Test unitarios en Javascript: una pequeña introducción*. Paradigma Digital. Recuperado 25 de agosto de 2022, de <https://www.paradigmadigital.com/dev/test-unitarios-javascript-introduccion/>

Suárez, I. (2021, 11 junio). *Herramientas de automatización de pruebas*. Apiumhub. Recuperado 25 de agosto de 2022, de <https://apiumhub.com/es/tech-blog-barcelona/herramientas-de-automatizacion-de-pruebas/>