

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Santa Fe



**Tecnológico
de Monterrey**

Pruebas de software

Iván Díaz Lara	A01365801
Andrea Yela González	A01025250
Emilio Sibaja Villarreal	A01025139

25 de agosto del 2022

1.-

Pruebas locales:

BrowserStack:

Ponen a prueba servidores privados, internos y archivos locales (CSS, HTML y JS) que siguen en producción y solo son accesibles desde una red local o la computadora del dueño.

Para poder realizar este tipo de prueba, se crea una conexión segura entre la computadora y “BrowserStack” donde la información se transfiere. Usando las máquinas virtuales de “BrowserStack” para hacer testing, uno puede evitar el tener que establecer y mantener “testing”, de misma manera ayuda mucho a ahorrar dinero y tiempo.

Pruebas de integración

Checan las diferentes interfaces que existan en cierto programa y sus interacciones con el sistema. Hay varios enfoques en este tipo de pruebas pero estos son los principales

Big Bang

“Todos los componentes o módulos se integran simultáneamente, después de lo cual todo se prueba como un todo.” (Trans-TI, 2020).

Descendente

“Las pruebas se llevan a cabo de arriba a abajo, siguiendo el flujo de control o la estructura arquitectónica (por ejemplo, comenzando desde la GUI o el menú principal). Los componentes o sistemas se sustituyen por stubs.” (Trans-TI, 2020).

Ascendente

“Las pruebas se llevan a cabo desde la parte inferior del flujo de control hacia arriba. Los componentes o sistemas se sustituyen por controladores.” (Trans-TI, 2020).

Incremental

“Otro enfoque es que todos los programadores se integran uno por uno y se realiza una prueba después de cada paso.” (Trans-TI, 2020).

Sandwich

“Las pruebas de integración de sándwich son una combinación de enfoques de arriba hacia abajo y de abajo hacia arriba. También se denomina prueba de integración híbrida o prueba de integración mixta. En las pruebas de integración sándwich, se considera que el sistema está compuesto por tres capas. Una capa en el medio que será

el objetivo de las pruebas, otra sobre la capa de destino y una capa debajo de la capa de destino. Las pruebas comienzan desde la capa exterior y convergen en la capa intermedia.” (Trans-TI, 2020).

Una de las herramientas que se puede usar para esta prueba es Gatling, la cuál detecta fácilmente errores y evita bloqueos, además es de código abierto. (Khatri, 2022)

Pruebas alfa

Son pruebas realizadas por “Testers” que son empleados de la organización, también se necesita de un ambiente dedicado al testeo de dicho software.

A diferencia del “Beta Testing”, en las pruebas Alfa no se pone a prueba la Seguridad, Confiabilidad y robustez de la aplicación, sin embargo los bugs y problemas se arreglan inmediatamente.

En conclusión, el testeo alfa ayuda a simular el “user environment” antes de que el software se libere para el testeo beta, y ayuda a moldear a dicho software para que sea elegible para avanzar el testeo Beta.

Pruebas de regresión

Su objetivo es comprobar que los cambios sobre un componente de un sistema no tenga un comportamiento no deseado o errores adicionales en otros componentes no modificados.

Appsurify:

Una herramienta de aprendizaje automático plug and play para realizar pruebas, encontrar defectos y acelerar los tiempos de ciclo. Puntos clave de esta aplicación son:

- Acorta el tiempo de finalización de la prueba de regresión
- Funciona con las prácticas de prueba existentes
- Evita que las pruebas inestables rompan la compilación
- Maximiza la cobertura de la prueba

Pruebas dinámicas de validación

Este tipo de pruebas se realizan al momento de la ejecución del programa. Se concentra en los requerimientos establecidos y que se hayan realizado correctamente. A diferencia de las pruebas estáticas, esta se realiza al final del programa y por lo mismo es más costosa.

Testim es una herramienta que se puede usar en este tipo de pruebas, siendo una herramienta comercial. Usa anotaciones para encontrar inconsistencias y errores en el sistema. Los errores que se registran se pueden reproducir automáticamente simplemente haciendo clic en la prueba nuevamente. (Khatri, 2022)

Pruebas bajo condiciones frontera

Estas pruebas se enfocan en los valores mínimos y máximos que se aceptan en el software. Se testean tanto los valores dentro del rango permitido, como aquellos que están fuera y se comprueba que se obtengan resultados válidos o incorrectos según el valor dado.

Aplicación de pruebas

Index	Name	ID	Graduation	Grade
1	Loreto Sepúlveda Torres	A12345678	24/06/2021	83
2	Yessica Castillo Noguera	A12346541	24/06/2021	93
3	Valentín Caro García	A12345696	18/12/2021	75
4	Virginia Rodríguez Guerra	A12345683	18/12/2021	86
5	Inocencio Arreola Soto	A15833310	19/12/2021	84
6	Benito Ramos Félix	A20693321	22/06/2022	71
7	Bernardina González Fontana	A20210921	23/06/2022	99
8	Gloria Mata Vázquez	A27780010	23/06/2022	75
9	Amaya Obando Sosa	A12433321	15/12/2022	84
10	Maricruz Muñoz Aquino	A44813137	15/12/2022	91

Index	Name	ID	Graduation	Grade
1	Loreto Sepúlveda	A12345678@tec.mx	06/24/2021	B
2	Yessica Castillo	A12346541@tec.mx	06/24/2021	A
3	Valentín Caro	A12345696@tec.mx	12/18/2021	C
4	Virginia Rodríguez	A12345683@tec.mx	12/18/2021	B
5	Inocencio Arreola	A15833310@tec.mx	12/19/2021	B
6	Benito Ramos	A20693321@tec.mx	06/22/2022	C-
7	Bernardina González	A20210921@tec.mx	06/23/2022	A
8	Gloria Mata	A27780010@tec.mx	06/23/2022	C
9	Amaya Obando	A12433321@tec.mx	12/15/2022	B
10	Maricruz Muñoz	A44813137@tec.mx	12/15/2022	A-

```

import "@testing-library/jest-dom";

import {
  grade_to_letter,
  date_to_american,
  remove_surname,
  format_line,
} from "../App";

describe("Individual functions", () => {
  test("grade conversion", () => {
    expect(grade_to_letter(99)).toBe("A");
    expect(grade_to_letter(94)).toBe("A");
    expect(grade_to_letter(92)).toBe("A-");
    expect(grade_to_letter(88)).toBe("B+");
    expect(grade_to_letter(80)).toBe("B-");
    expect(grade_to_letter(74)).toBe("C");

    expect(grade_to_letter(30)).toBe("E");
  });

  test("date conversion", () => {
    expect(date_to_american("17/08/2022")).toBe("08/17/2022");
    expect(date_to_american("31/12/2022")).toBe("12/31/2022");
    expect(date_to_american("01/01/2022")).toBe("01/01/2022");
  });

  test("remove surname", () => {
    expect(remove_surname("One Two Three")).toBe("One Two");
    expect(remove_surname("Gilberto Echeverria Furio")).toBe(
      "Gilberto Echeverria"
    );
    expect(remove_surname("Paco Gomez Juarez")).toBe("Paco Gomez");
  });
});

describe("Complete line", () => {
  test("convert line", () => {
    expect(format_line(0, "1")).toBe("1");
    expect(format_line(1, "One Two Three")).toBe("One Two");
    expect(format_line(2, "A01234567")).toBe("A01234567@tec.mx");
    expect(format_line(3, "20/08/2020")).toBe("08/20/2020");
    expect(format_line(4, "70")).toBe("C-");
  });
});

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PASS src/func.test.js
  Individual functions
    ✓ grade conversion (2 ms)
    ✓ date conversion
    ✓ remove surname (1 ms)
  Complete line
    ✓ convert line (1 ms)

Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        3.021 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.█
```

Para nuestra aplicación usamos pruebas **unitarias locales**, donde “testemos” de manera exitosa las siguientes funciones de manera individual y grupal:

Funciones Individuales:

- garde_to_letter();
- date_to_american();
- remove_surname();

Grupo de Funciones:

- format_line();

Referencias

BrowserStack. (2022) *What is Local Testing and why should I use it?*

<https://www.browserstack.com/question/39563#:~:text=Local%20Testing%20is%20the%20ability,servers%2C%20or%20work%20in%20progress.>

Castillo, E. (s.f.). *Planeación de sistemas de software.*

https://docs.google.com/presentation/d/1MOB-bNlbC_f6hPCgm4N0gC5iGpkvMBPcrvFdktzGD4k/edit#slide=id.p1

ESTRATEGIAS DE PRUEBA DE SOFTWARE. (s.f.).

http://artemisa.unicauca.edu.co/~cardila/CS_06_Estrategias_de_Prueba_SW.pdf

Gomez, C (2021) *Pruebas dinámicas vs pruebas estáticas.*

<https://www.diariodeqa.com/post/pruebas-din%C3%A1micas-vs-pruebas-est%C3%A1ticas>

Guru99 (2020) *Alpha Testing Vs Beta Testing: What's the Difference?*

<https://www.guru99.com/alpha-beta-testing-demystified.html>

Khatri, V. (2022) *Las 19 mejores herramientas de prueba de software que debe conocer como evaluador.* <https://geekflare.com/es/software-testing-tools/>

Trans-Ti (2020), *¿Qué Son Las Pruebas De Integración En El Software Testing?*

<https://trans-ti.com/2020/12/14/que-son-las-pruebas-de-integracion-en-el-software-testing/>

Boundary Value Analysis. (s.f.).

<https://www.javatpoint.com/boundary-value-analysis-in-black-box-testing>