

# HERRAMIENTA

Jest para aplicaciones

web JavaScript

Grupo 8

# ¿Qué es Jest?

- Framework de pruebas unitarias diseñado para aplicaciones JavaScript.
- Creado y mantenido por Facebook.
- Optimizado para trabajar con bibliotecas como React, pero funciona con cualquier proyecto JS.
- Enfoque: simplicidad y rendimiento en la ejecución de pruebas automatizadas.



# Características principales

- Fácil configuración
- Se integra rápidamente en la mayoría de los proyectos.
- Ejecución de pruebas en paralelo
- Aumenta la velocidad en grandes conjuntos de pruebas.
- Snapshot Testing
- Ideal para asegurarse de que las interfaces o salidas no cambien inesperadamente.
- Cobertura de código
- Proporciona un informe detallado sobre qué líneas han sido cubiertas por las pruebas.
- Soporte para mocks
- Simula funciones y módulos para crear entornos de pruebas más predecibles.



# Funcionalidades adicionales



3

## Coverage Thresholds:

*Configura umbrales mínimos de cobertura de código para asegurar la calidad del código.*

## Global Setup & Teardown:

*Ejecuta scripts globales antes y después de las pruebas para preparar entornos o datos.*

## Test Suites y Test Cases:

*Agrupar pruebas lógicas en "suites" usando `describe()`, lo que permite organizar pruebas de manera más clara*

## Test Hooks (Before/After):

*Hooks como `beforeEach()`, `afterEach()`, `beforeAll()`, `afterAll()` permiten preparar o limpiar el estado antes/después de las pruebas.*

# Tipos de pruebas soportadas

4

**Pruebas Unitarias:** Validan el funcionamiento de funciones o componentes individuales.

**Pruebas de integración:** Verifican cómo interactúan diferentes partes de la aplicación.

**Pruebas de Instantáneas (Snapshot Testing):** Comparan la salida actual con una versión previamente guardada.

**Pruebas de Asincronía:** Soporte para promesas y `async/await`, asegurando que las funciones asíncronas se comporten correctamente.



# Integraciones de Jest

**React:** Herramienta ideal para pruebas de componentes React.

**Node.js:** Soporta tanto aplicaciones back-end como front-end.

**Babel y Webpack:** Integración directa para proyectos que requieren transpiladores y empaquetadores.

**TypeScript:** Soporte para proyectos con TypeScript mediante configuración mínima.



# Comparativa con otras herramientas

## Jest vs Mocha

**Configuración:** Jest no requiere configuraciones adicionales. Mocha depende de bibliotecas como Chai y Sinon.

**Ejecución paralela:** Jest ejecuta pruebas en paralelo automáticamente, mientras que Mocha requiere configuraciones manuales.

**Snapshot Testing:** Funcionalidad nativa en Jest, no disponible en Mocha.



VS



# Conclusión

**Jest es un framework completo y versátil para realizar pruebas unitarias, de integración y de instantáneas.**

**Su simplicidad y velocidad lo convierten en la herramienta preferida en el desarrollo de aplicaciones JavaScript.**

**Altamente compatible con bibliotecas y tecnologías modernas como React, Node.js, Babel, y TypeScript.**





# Integrantes

- 1.Tomás Armando Campos Lopez -CI231461
- 2.Alvaro Ricardo Salazar Hernandez -SH191788
- 3.Asly Eduvina Hernandez Melara -HM240071
- 4.Omarvis Innaun Mendoza Portillo -MP192089

G01T

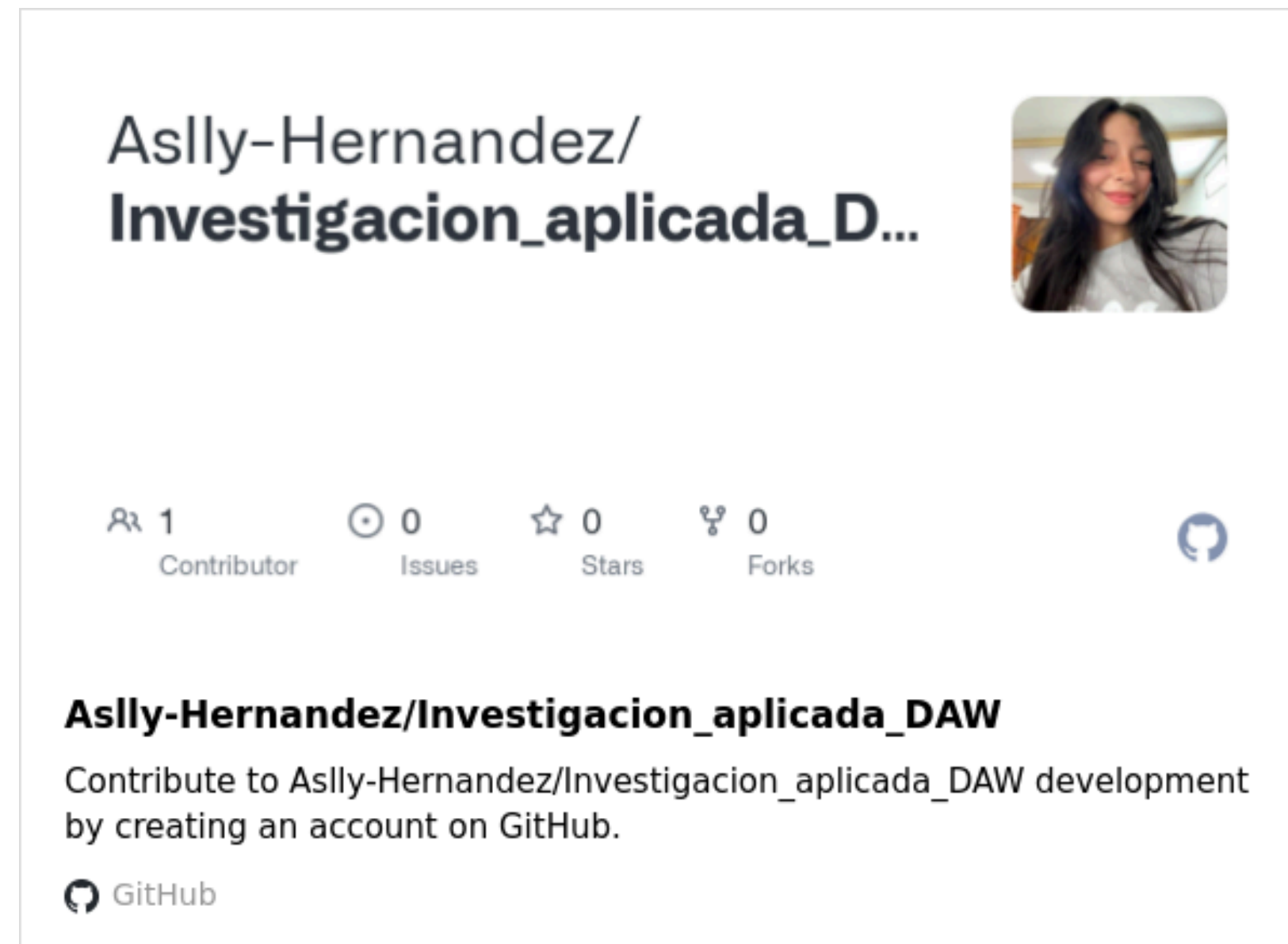
# Bibliografias

[1]	L. Hoo, "Testing React Applications with Jest," Journal of Web Development, vol. 15, no. 3, pp. 45-56," 2022. [Online].
[2]	J. Doe, "Comparative Analysis of JavaScript Testing Frameworks," International Journal of Software Engineering, vol. 12, no. 4, pp. 25-40," 2023. [Online].
[3]	R. Johnson, "An Introduction to Unit Testing with Jest," Software Testing Insights, vol. 5, no. 2, pp. 12-18," 2023. [Online].

# Reposito Github



9



[https://github.com/Aslly-Hernandez/Investigacion\\_Aplicada](https://github.com/Aslly-Hernandez/Investigacion_Aplicada)



***GRACIAS***

