

Bootcamp de Full Stack

Bienvenidos a la clase N°64

- Pruebas de Software
- Documentación de API - Swagger
- Cierre de cursada

Fundamentos

Pruebas de Software

¿Qué son las Pruebas de Software?

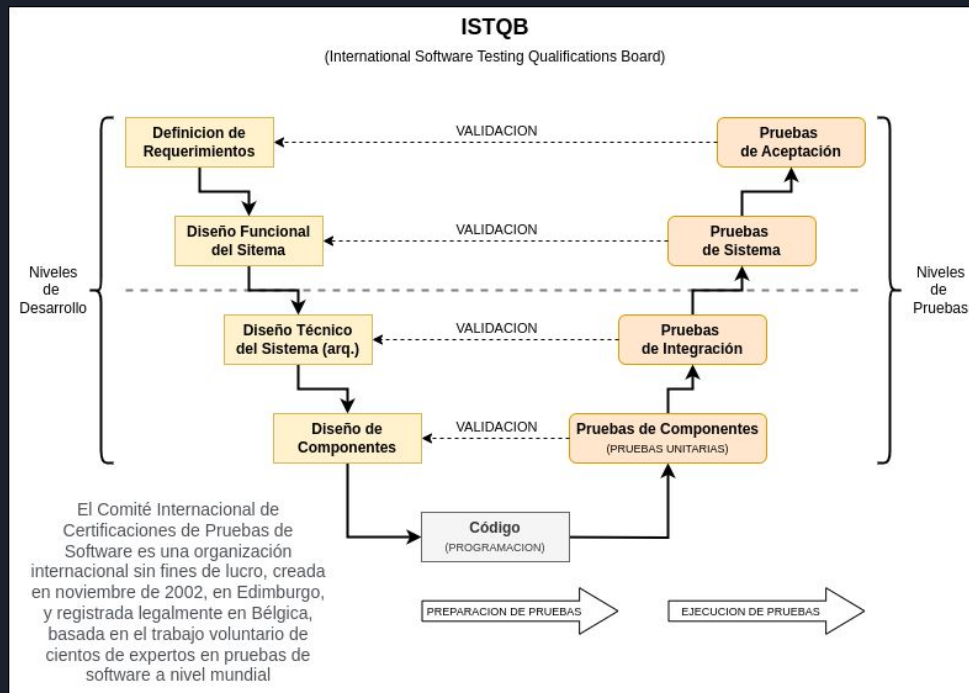
Son un conjunto de actividades realizadas para evaluar y validar un programa de computadora con el objetivo de garantizar que cumpla con los requisitos especificados y funcione correctamente. Estas pruebas se llevan a cabo en diferentes etapas del ciclo de vida del desarrollo de software y se centran en detectar errores, defectos o fallos en el software antes de que sea puesto en producción.

Principales tipos de pruebas:

- Pruebas unitarias: Se centran en probar cada componente de forma individual
- Pruebas de integración: Verifican la interacción entre diferentes componentes
- Pruebas de sistema: Se realizan sobre el sistema completo, comprobando su comportamiento en un entorno similar al de producción
- Pruebas de aceptación: Estas pruebas son llevadas a cabo por los usuarios finales o representantes del cliente. El objetivo es asegurarse de que el software cumple con las expectativas y necesidades del usuario final.

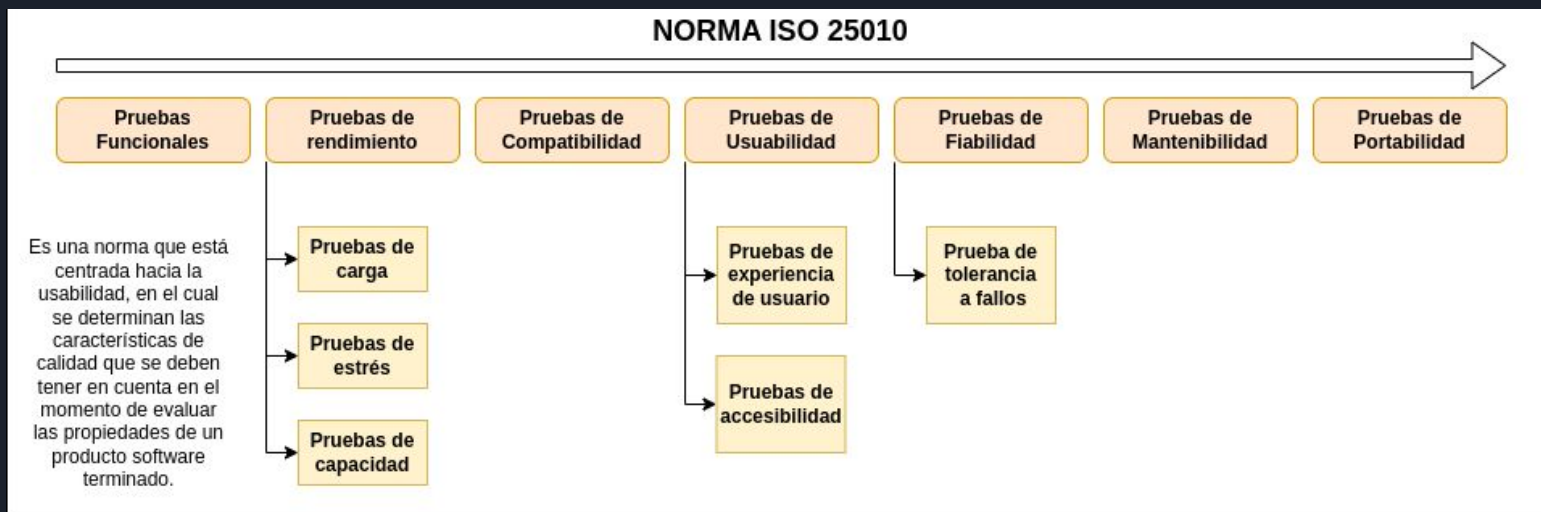
Fundamentos

Pruebas de Software



Fundamentos

Pruebas de Software



Fundamentos

Pruebas de Software

¿Qué es Jest?

Es un marco de prueba (testing framework) popular para JavaScript, específicamente diseñado para probar aplicaciones y bibliotecas escritas en React, pero también es compatible con otros frameworks y bibliotecas de JavaScript. Jest fue desarrollado por Facebook y se caracteriza por ser fácil de configurar y usar, ofreciendo una amplia gama de funcionalidades para realizar:

- Pruebas unitarias
- Pruebas de integración
- Pruebas de componentes de React

Documentación oficial: <https://jestjs.io/docs/getting-started>

Actividad

Pruebas de Software

Consigna:

- Crear un servidor HTTP local
- Instalar JEST: `npm install jest --save-dev`
- Definir pruebas unitarias para una calculadora

Fundamentos

Documentación de API

¿Qué significa documentar una API?

Significa crear una guía detallada que describe cómo usar y acceder a una interfaz de programación de aplicaciones (API). Esta documentación proporciona información esencial sobre los endpoints de la API, los métodos admitidos, los parámetros requeridos y opcionales, los tipos de datos que se pueden enviar y recibir, así como ejemplos de uso y casos de uso típicos.

¿Qué es Swagger?

Es una herramienta de código abierto para documentar y diseñar APIs. Proporciona una forma de describir la funcionalidad de una API de una manera clara y legible tanto para humanos como para máquinas. Con Swagger, puedes crear documentación interactiva para tu API, lo que facilita a los desarrolladores comprender cómo usarla y probarla.

¿Qué es JSDoc?

Es una herramienta de documentación para JavaScript que permite generar documentación a partir de comentarios especiales en el código fuente. Estos comentarios están escritos en un formato específico que sigue ciertas convenciones y pueden incluir información sobre los tipos de datos, parámetros, valores de retorno y descripciones de funciones, métodos, clases y otros elementos del código.

Fundamentos

Documentación de API

¿Para qué sirven las dependencias `swagger-jsdoc` y `swagger-ui-express`?

Estas dependencias se utilizan típicamente en proyectos de Node JS. Proporcionan una forma eficiente para documentar, compartir y consumir APIs RESTful.

- Dependencia **`swagger-jsdoc`**: Permite escribir documentación Swagger en forma de comentarios en el código fuente. Específicamente, permite definir la documentación de tus rutas utilizando anotaciones especiales en tus comentarios, puedes describir los endpoints, sus parámetros, respuestas esperadas, etc.
- Dependencia **`swagger-ui-express`**: Es un middleware que sirve la interfaz de usuario de Swagger UI. Una vez que se ha generado la documentación de la API con `swagger-jsdoc` y se ha obtenido el documento Swagger en formato JSON, se puede utilizar `swagger-ui-express` para montar una ruta en la aplicación que sirva la interfaz de Swagger UI.

En resumen, **`swagger-jsdoc`** te ayuda a escribir la documentación de tu API en tu código fuente, mientras que **`swagger-ui-express`** te permite visualizar esta documentación de una manera amigable para los usuarios finales.

BREAK

Descansemos 15 minutos



Cierre de Cursada

Consigna:

- Jugar a sopa de letras <https://puzzel.org/es/wordseeker/play?p=-Nv-S2IIL-vL3YVcVkXi>
- Realizar un balance sobre la cursada
- Temas de interés
- Compartir [PPT](#)

CIERRE DE CLASE

Continuaremos en la próxima clase

