

Refactorización:

La refactorización es el proceso de reestructurar el código existente sin cambiar su comportamiento externo. Se realiza para mejorar la legibilidad, mantenibilidad y eficiencia del código sin alterar su funcionalidad.

Mockito:

Mockito es un marco de simulación (mocking framework) para pruebas unitarias en Java. Permite crear objetos simulados (mocks) para simular el comportamiento de objetos reales en pruebas unitarias.

Mocks:

Los mocks son objetos simulados que se utilizan en pruebas unitarias para simular el comportamiento de objetos reales. Se configuran para devolver resultados específicos o simular determinadas interacciones durante las pruebas.

Código limpio:

El código limpio se refiere a un estilo de codificación que enfatiza la legibilidad, la simplicidad y la mantenibilidad del código. Un código limpio es fácil de entender, modificar y depurar.

Código smell:

Un código smell, o "olor a código" en español, es un indicio o una señal de que un código puede tener problemas de diseño o calidad. Los code smells no son errores directos, pero pueden indicar áreas donde el código necesita refactorización.

Refactoring:

Refactoring es el proceso de mejorar el diseño interno del código sin cambiar su comportamiento externo. Se realiza para eliminar code smells, mejorar la estructura del código y hacerlo más fácil de entender y mantener.

Deuda técnica:

La deuda técnica se refiere a compromisos técnicos a corto plazo que se toman para acelerar el desarrollo de software pero que pueden resultar en costos adicionales en el futuro. Es similar a la deuda financiera: se contrae con la intención de pagarla más tarde, pero si no se aborda, puede acumularse y causar problemas en el futuro.

SOLID:

SOLID es un acrónimo que representa cinco principios de diseño de software:

S (Single Responsibility Principle): Cada clase debe tener una única responsabilidad.

O (Open/Closed Principle): Las clases deben estar abiertas para la extensión pero cerradas para la modificación.

L (Liskov Substitution Principle): Los objetos de un programa deben ser reemplazables por instancias de sus subtipos sin alterar la corrección del programa.

I (Interface Segregation Principle): Los clientes no deben depender de interfaces que no usen.

D (Dependency Inversion Principle): Las clases deben depender de abstracciones en lugar de implementaciones concretas.

Tipos de pruebas:

Hay varios tipos de pruebas en el desarrollo de software, incluyendo pero no limitado a:

Pruebas unitarias

Pruebas de integración

Pruebas de aceptación

Pruebas de regresión

Pruebas de rendimiento

Caja negra:

La prueba de caja negra se centra en la funcionalidad externa de un programa sin conocer su estructura interna. Se prueban las entradas y se observan las salidas correspondientes sin tener en cuenta cómo se procesan dentro del sistema.

Caja blanca:

La prueba de caja blanca se centra en la estructura interna del software y en cómo se implementan las funciones. Se prueban los caminos de ejecución dentro del código para garantizar que se ejecute correctamente y que se cubran todas las posibles condiciones.

JUNIT:

JUnit es un marco de pruebas unitarias para Java. Permite escribir y ejecutar pruebas automatizadas para garantizar que las unidades individuales de código funcionen como se espera.

JDK:

JDK significa Java Development Kit. Es un conjunto de herramientas de desarrollo para crear programas Java. Incluye el compilador Java, bibliotecas Java y otras herramientas necesarias para desarrollar aplicaciones Java.

SDK:

SDK significa "Software Development Kit". Es un conjunto de herramientas de desarrollo de software que permite a los programadores crear aplicaciones para una plataforma específica. Por lo general, incluye un conjunto de bibliotecas, herramientas de compilación, documentación y ejemplos de código.

Tests:

Los tests, o pruebas en español, son procedimientos automáticos que se utilizan para verificar el correcto funcionamiento de una parte específica del software. Se utilizan para detectar errores y garantizar que el software cumpla con los requisitos especificados.

Anotaciones:

Las anotaciones son metadatos que se pueden agregar al código fuente de un programa en Java. Proporcionan información adicional sobre cómo debe tratarse el código. En Java, las anotaciones se prefijan con el símbolo "@" y se utilizan para proporcionar información a los compiladores, herramientas de análisis estático o marcos de trabajo.

Proyectos:

Un proyecto en el contexto del desarrollo de software es un esfuerzo organizado para lograr un objetivo específico, como desarrollar un software, una aplicación o una función específica. Un proyecto suele tener un alcance definido, un equipo asignado, un presupuesto y un plazo.

MAVEN:

Maven es una herramienta de gestión de proyectos de software basada en Java. Se utiliza para gestionar la construcción, dependencias y documentación de proyectos Java. Maven utiliza un archivo XML llamado "pom.xml" para definir la configuración del proyecto y las dependencias.

GRADLE:

Gradle es otra herramienta de gestión de proyectos de software que se utiliza principalmente en el ecosistema de desarrollo de Java. Ofrece flexibilidad y potencia en la configuración y construcción de proyectos. Se utiliza para automatizar tareas de construcción, gestión de dependencias y otros aspectos del desarrollo de software.

Javadocs:

Javadoc es una herramienta utilizada en Java para generar documentación a partir de comentarios en el código fuente. Los comentarios específicamente formateados con la sintaxis de Javadoc pueden ser procesados por la herramienta para producir documentación legible y estructurada.

Archivos JAR:

JAR (Java ARchive) es un formato de archivo utilizado para distribuir bibliotecas, componentes y aplicaciones Java. Un archivo JAR puede contener código compilado, recursos, metadatos y otros archivos necesarios para la ejecución de una aplicación Java.

Dockerfile:

Un Dockerfile es un archivo de texto que contiene instrucciones para construir una imagen Docker. Define los pasos necesarios para configurar el entorno de ejecución y ejecutar una aplicación dentro de un contenedor Docker. Los Dockerfiles son utilizados en el proceso de creación y despliegue de aplicaciones utilizando Docker.

DOCKER:

Docker es una plataforma de software que utiliza contenedores para facilitar la creación, implementación y ejecución de aplicaciones. Los contenedores Docker encapsulan una aplicación y todas sus dependencias en un entorno autónomo que se puede ejecutar en cualquier sistema compatible con Docker.

Volúmenes:

En Docker, los volúmenes son un mecanismo para persistir datos generados por contenedores o para compartir datos entre contenedores. Los volúmenes Docker son directorios montados fuera del sistema de archivos raíz del contenedor, lo que permite que los datos persistan incluso después de que el contenedor se detenga o se elimine.

Código fuente:

El código fuente es el conjunto de instrucciones escritas en un lenguaje de programación específico que constituyen un programa de computadora. Es el formato legible por humanos en el que los programadores escriben y modifican el software.

Código objeto:

El código objeto es el resultado de la fase de compilación de un programa de computadora. Es el código de máquina generado por el compilador a partir del código fuente, pero aún no es ejecutable.

Código ejecutable:

El código ejecutable es el programa de computadora que puede ser ejecutado por el hardware de una computadora. Es el resultado final del proceso de compilación, en el que el código fuente se transforma en código objeto y luego se vincula para formar un programa completo y funcional.

Compilador:

Un compilador es un programa informático que traduce código fuente escrito en un lenguaje de programación específico a código objeto ejecutable por una computadora. Los compiladores son herramientas fundamentales en el proceso de desarrollo de software.

Lenguajes interpretados:

Los lenguajes interpretados son aquellos en los que el código fuente es ejecutado directamente por un intérprete, en lugar de ser previamente compilado a código objeto. Los lenguajes interpretados suelen ser más lentos que los compilados, pero ofrecen flexibilidad y portabilidad.

Las fases de desarrollo de programas:

Las fases de desarrollo de programas son los pasos o etapas que se siguen en el proceso de desarrollo de software, que pueden incluir: análisis de requisitos, diseño, implementación, pruebas, despliegue y mantenimiento.

Desarrollo en cascada, V, espiral:

Estos son modelos de desarrollo de software que describen diferentes enfoques para organizar las fases de desarrollo de un proyecto. El desarrollo en cascada sigue un flujo lineal de trabajo, el modelo V organiza las fases en forma de V, y el desarrollo en espiral incorpora iteraciones y retroalimentación en el proceso.

React:

React es una biblioteca de JavaScript utilizada para construir interfaces de usuario interactivas y dinámicas para aplicaciones web. Desarrollado por Facebook, React se centra en la creación de componentes reutilizables que gestionan su propio estado y se actualizan de manera eficiente cuando cambian los datos.

POO (Programación Orientada a Objetos):

La programación orientada a objetos es un paradigma de programación que se basa en el concepto de "objetos", que pueden contener datos en forma de campos y procedimientos en forma de métodos. La POO enfatiza la modularidad y el reuso del código, así como la encapsulación, herencia y polimorfismo.

Bootstrap:

Bootstrap es un framework de desarrollo front-end de código abierto que facilita el diseño y la creación de sitios web y aplicaciones móviles. Proporciona un conjunto de herramientas y componentes predefinidos, así como estilos CSS y JavaScript, para ayudar a los desarrolladores a construir interfaces de usuario responsivas y atractivas de manera rápida y sencilla.

Framework:

Un framework es un conjunto de herramientas, bibliotecas y convenciones que proporciona una estructura para el desarrollo de software. Los frameworks facilitan la creación de aplicaciones al ofrecer funcionalidades comunes y abstracciones que permiten a los desarrolladores concentrarse en aspectos específicos del problema.