

Taller sobre codificación de módulos de software

Yuliet Faizuli Pachon Caro

Jorge Miller Gutierrez Ospina

Analisis y desarrollo de software

Instructor

Ivan Leonardo Medina Gomez

SENA

Septiembre 2024

Introducción

Este taller se enfoca en las pruebas de software aplicadas a la aplicación de escritorio en Java para la gestión del consultorio odontológico “Maria Jose Velez Gutierrez”. En el desarrollo de sistema, es esencial garantizar que los módulos dedicados a la gestión de pacientes, odontólogos, citas, y demás elementos del sistema funcionen correctamente. A través de este taller, se evaluarán diferentes tipos de pruebas de software.

1. Tipos de Pruebas de Software

Para asegurar que un sistema funcione correctamente, se utilizan diversos tipos de pruebas de software. A continuación, se describen las más relevantes para este proyecto:

1.1 Pruebas Unitarias:

- **Características:** Verifican el correcto funcionamiento de los módulos o funciones individuales de la aplicación. En el caso de una aplicación de escritorio en Java, se realizan sobre clases como `Ctrl_Usuario`, `Ctrl_Pacientes`, `Ctrl_Odontologos`, y los métodos relacionados con la gestión de datos.
- **Beneficios:** Facilitan la identificación temprana de errores en la lógica interna de cada componente sin depender de otros módulos.

1.2 Pruebas de Integración:

- **Características:** Se encargan de probar la interacción entre los diferentes módulos de la aplicación, como la integración entre la gestión de pacientes y la agenda de citas.
- **Beneficios:** Aseguran que los módulos se comuniquen correctamente y que los datos fluyan entre los diferentes componentes.

1.3 Pruebas Funcionales:

- **Características:** Verifican que el sistema cumpla con los requisitos funcionales, como el registro de pacientes o la consulta de la agenda de citas de un odontólogo.
- **Beneficios:** Se enfocan en asegurar que el sistema entregue los resultados esperados con base en los requisitos del usuario.

1.4 Pruebas de Regresión:

- **Características:** Se ejecutan para verificar que después de realizar modificaciones o actualizaciones en el código, el sistema siga funcionando como antes sin introducir errores.
- **Beneficios:** Garantizan la estabilidad del sistema tras cada actualización.

1.5 Pruebas de Aceptación:

- **Características:** Realizadas por el cliente final o usuario del sistema (en este caso, el consultorio odontológico) para validar que el software cumpla con los requisitos del negocio.
- **Beneficios:** Confirman que el sistema es apto para su uso en el entorno real.

1.6 Pruebas de Rendimiento:

- **Características:** Evaluar cómo la aplicación responde bajo condiciones de carga, por ejemplo, cuando se registran varios pacientes simultáneamente o se consultan múltiples citas a la vez.
- **Beneficios:** Identifican cuellos de botella en la eficiencia y velocidad del sistema.

2. Pruebas Adecuadas para el Proyecto de Software

En el contexto de esta aplicación para la gestión de un consultorio odontológico, las pruebas que mejor se adaptan a este tipo de solución son:

- **Pruebas Unitarias:** Dado que la aplicación maneja varias operaciones en clases específicas como `Ctrl_Usuario`, `Ctrl_Pacientes`, y `Ctrl_Citas`, es crucial verificar que cada módulo funcione correctamente en aislamiento.

- **Pruebas de Integración:** Debido a que la aplicación involucra la interacción entre la gestión de pacientes, odontólogos y citas, las pruebas de integración son fundamentales para asegurar que los diferentes módulos se comuniquen y funcionen como un sistema cohesivo.

- **Pruebas Funcionales:** Estas son importantes para asegurarse de que las operaciones clave del sistema (como registrar pacientes, gestionar citas y consultar la agenda de los odontólogos) funcionen conforme a los requisitos del consultorio.

- **Pruebas de Aceptación:** Son vitales para validar que el software es útil y cumple con las expectativas del consultorio odontológico, ya que los usuarios finales deben sentirse cómodos y confiados al usar el sistema.

3. Herramienta de Pruebas de Software: JUnit

Para la realización de pruebas unitarias en la aplicación, se ha seleccionado “JUnit” como herramienta de pruebas de software. JUnit es ampliamente utilizado para probar aplicaciones en Java y se integra fácilmente con el IDE que se está utilizando (Eclipse, NetBeans, IntelliJ IDEA).

Pasos para la instalación y configuración de JUnit:

1. **Descargar JUnit:** Si se utiliza un IDE como IntelliJ IDEA o Eclipse, es probable que ya esté integrado. Si no, se puede descargar desde [JUnit](https://junit.org/junit5/).

2. **Agregar JUnit al proyecto:** En el caso de no estar integrado, se agrega a través del administrador de dependencias del proyecto (Maven, Gradle) o manualmente al **classpath**.

3. **Configurar una clase de prueba:** Se crea una clase de prueba que utiliza los métodos `@Test` de JUnit para verificar la funcionalidad de cada módulo.

4. Pruebas Básicas Realizadas

Se realizaron pruebas unitarias sobre los módulos principales del sistema:

- **Prueba 1:** Registro de Pacientes

- Se probó el método `guardarPaciente` en la clase `Ctrl_Pacientes`, verificando que los datos del paciente se guardan correctamente en la base de datos.

- Prueba 2: Consultar Citas

- Se probó la funcionalidad de `consultarCitas` en la clase `Ctrl_Citas` para validar que el sistema devuelve correctamente las citas programadas de un odontólogo.

5. Resumen de Pruebas Realizadas

Se realizaron “5 pruebas unitarias” que abarcaron la funcionalidad de las clases `Ctrl_Usuario`, `Ctrl_Pacientes`, y `Ctrl_Citas`. Los resultados de las pruebas fueron exitosos, lo que indica que los métodos clave en estas clases funcionan correctamente. Además, se ejecutaron ****pruebas de integración**** para verificar la interacción entre los módulos de pacientes y citas, con resultados positivos.

Las pruebas realizadas cubrieron las siguientes áreas:

1. Registro y consulta de pacientes.
2. Gestión de citas.
3. Validación de acceso de usuarios.

Estas pruebas garantizaron que el sistema maneja los datos correctamente y permite una gestión fluida del consultorio odontológico.

Conclusiones

Las pruebas de software son esenciales para garantizar la calidad y funcionalidad de cualquier aplicación, y en el contexto de un la aplicación “sistema de información consultorio odontológico Maria Jose Velez Gutierrez”, aseguran que el sistema funcione sin problemas en un entorno crítico. Las pruebas unitarias y de integración han sido claves para verificar que cada módulo de la aplicación interactúe correctamente. Herramientas como JUnit permiten una validación eficiente de la lógica del sistema, asegurando su estabilidad antes de ser implementado en un entorno de producción.