

REALIZA PLAN PRUEBAS DE SOFTWARE

Presentado por: David Fernando León Barreiro

Instructor (a): Frank Jairo Castillo Padilla

Ficha: **2721423**

Servicio Nacional de Aprendizaje **SENA**

Tecnología en Análisis y Desarrollo de Software **ADSO**

CENTRO DE MANUFACTURA EN TEXTIL Y CUERO

Distrito Capital.

GA9-220501096-AA1-EV02

Realiza plan pruebas de software

Contenido

Introducción	3
Tipos de pruebas:	4
Herramientas:	4
Criterios de entrada y salida	4
Casos de Prueba	4
Planificación y Cronograma	5
Recursos Necesarios.....	5
Gestión de Defectos	5
Riesgos y Contingencias	5
Reportes de Pruebas	5
Aprobaciones.....	5

Introducción

El proceso de pruebas de software es una etapa fundamental en el ciclo de vida del desarrollo de aplicaciones, ya que garantiza la calidad, funcionalidad y desempeño del producto antes de su entrega o implementación en entornos productivos. Las pruebas no solo validan que el software cumpla con los requerimientos especificados, sino que también ayudan a identificar errores, riesgos y áreas de mejora, asegurando una experiencia de usuario satisfactoria y la estabilidad del sistema.

El presente plan de pruebas tiene como objetivo proporcionar una guía detallada y estructurada para evaluar el software de manera eficiente, siguiendo un enfoque sistemático y basado en buenas prácticas. A través de este plan, se busca minimizar el impacto de defectos en etapas avanzadas del proyecto, reducir costos asociados con correcciones tardías y garantizar que el producto final cumpla con los estándares de calidad esperados.

En este documento, se detallan los pasos necesarios para analizar los requerimientos del software, identificar las funcionalidades a evaluar, establecer estrategias de prueba, definir los entornos necesarios y diseñar los artefactos que permitan un registro completo y ordenado de las pruebas realizadas. Asimismo, se presentan las herramientas sugeridas para implementar las pruebas y los riesgos potenciales que podrían afectar el proceso, junto con sus respectivas medidas de mitigación.

Con esta introducción, se establece el marco conceptual para la planificación, ejecución y documentación de las pruebas de software, subrayando la importancia de un enfoque metódico para garantizar la calidad en cada etapa del desarrollo.

Los planes de pruebas constan de diferentes pasos los cuales describiremos aquí:

Tipos de pruebas:

- **Pruebas unitarias:** Verificar cada componente de manera aislada.
- **Pruebas de integración:** Comprobar que los módulos funcionan correctamente juntos.
- **Pruebas funcionales:** Evaluar que el sistema cumple con los requisitos funcionales.
- **Pruebas de regresión:** Asegurar que nuevas modificaciones no afecten funcionalidades existentes.
- **Pruebas de carga o rendimiento:** Evaluar el comportamiento bajo diferentes niveles de carga.
- **Pruebas de seguridad:** Verificar que la aplicación protege datos sensibles.

Herramientas:

Menciona las herramientas que usarás (ej., Selenium, JUnit, Postman, etc.).

Criterios de entrada y salida

Define los requisitos para iniciar y finalizar el proceso de pruebas.

- **Criterios de entrada:**
 - Requisitos del sistema claramente definidos.
 - Ambiente de pruebas configurado y estable.
 - Casos de prueba aprobados.
- **Criterios de salida:**
 - Todas las pruebas críticas pasadas.
 - Todos los defectos críticos solucionados.
 - Informes de pruebas completados.

Casos de Prueba

Describe los casos de prueba que serán ejecutados.

- **Identificación del caso de prueba:** Número único o identificador.
- **Descripción:** Qué funcionalidad será probada.
- **Entradas:** Datos necesarios para ejecutar la prueba.
- **Resultados esperados:** Comportamiento esperado del sistema.
- **Resultados reales:** Registro del comportamiento observado.

Planificación y Cronograma

Especifica el tiempo necesario para las pruebas y los responsables.

- **Cronograma:** Fechas clave para cada etapa de las pruebas.
- **Responsabilidades:** Quién realizará cada tipo de prueba.

Recursos Necesarios

Lista los recursos técnicos y humanos requeridos.

- **Hardware y software:** Servidores, entornos de prueba, bases de datos.
- **Personal:** Testers, desarrolladores, administradores de sistemas.

Gestión de Defectos

Explica cómo se registrarán, rastrearán y solucionarán los defectos.

- **Sistema de gestión:** Ejemplo: Jira, Bugzilla, Trello.
- **Proceso:** Definición de prioridades, asignación de responsables, cierre de defectos.

Riesgos y Contingencias

Identifica posibles riesgos y cómo serán mitigados.

- **Ejemplos de riesgos:**
 - Entorno de pruebas no está listo a tiempo.
 - Recursos humanos insuficientes.
 - Requisitos cambiantes.
- **Mitigaciones:** Alternativas para enfrentar cada riesgo.

Reportes de Pruebas

Detalla cómo se presentarán los resultados.

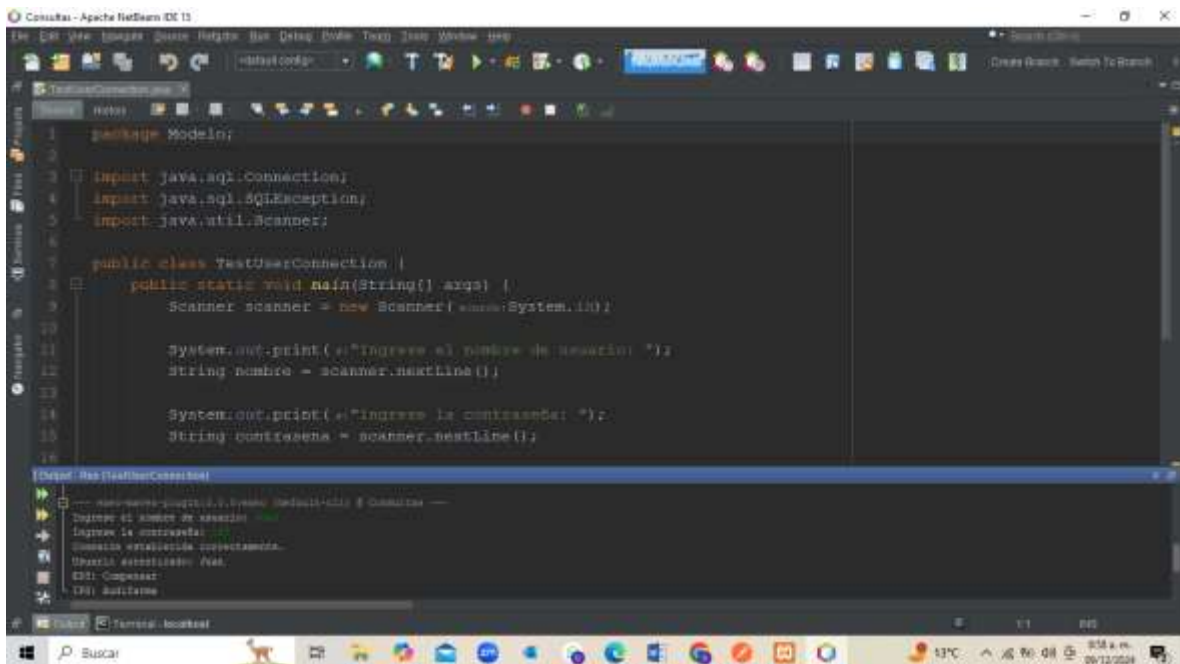
- **Formato:** Gráficos, tablas, resúmenes de defectos.
- **Frecuencia:** Diario, semanal, al finalizar las pruebas.

Aprobaciones

Incluye una sección para que los responsables revisen y aprueben el plan.

- **Firmas:** Equipo de pruebas, desarrolladores, gerencia.

Otra opción consiste en escribir nuestro propio código de prueba personalizado, adaptándolo a los requisitos específicos del sistema y asegurando que cubra los escenarios necesarios para validar el correcto funcionamiento del software.

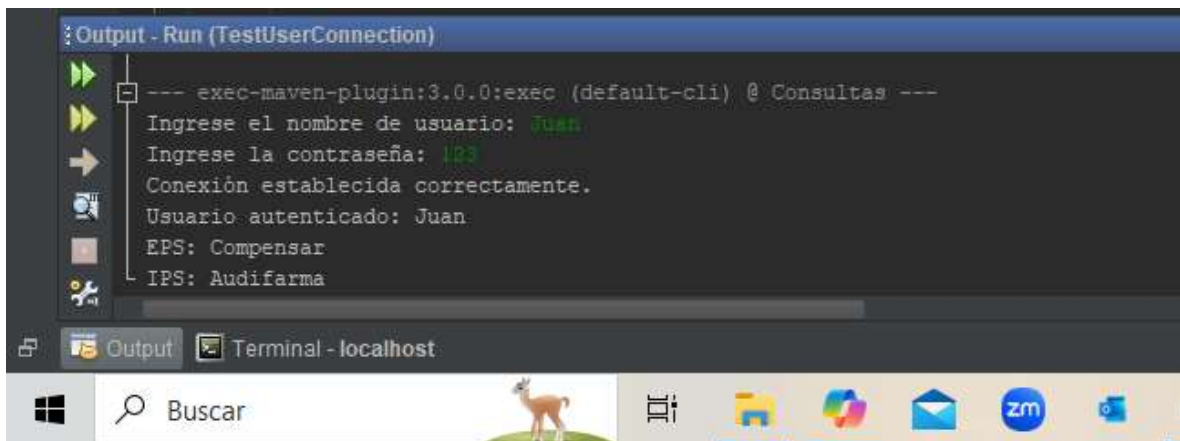


The screenshot shows the Apache NetBeans IDE interface. The main editor displays the `TestUserConnection.java` file with the following code:

```
1 package Modelo;
2
3 import java.sql.Connection;
4 import java.sql.SQLException;
5 import java.util.Scanner;
6
7 public class TestUserConnection {
8     public static void main(String[] args) {
9         Scanner scanner = new Scanner(System.in);
10
11         System.out.print("Ingrese el nombre de usuario: ");
12         String nombre = scanner.nextLine();
13
14         System.out.print("Ingrese la contraseña: ");
15         String contraseña = scanner.nextLine();
16     }
17 }
```

The Output window at the bottom shows the execution results:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Consultas ---
Ingrese el nombre de usuario: Juan
Ingrese la contraseña: 123
Conexión establecida correctamente.
Usuario autenticado: Juan
EPS: Compensar
IPS: Audifarma
```



The screenshot shows the Output window titled "Output - Run (TestUserConnection)". It displays the execution results of the test, including the user input and the system output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Consultas ---
Ingrese el nombre de usuario: Juan
Ingrese la contraseña: 123
Conexión establecida correctamente.
Usuario autenticado: Juan
EPS: Compensar
IPS: Audifarma
```