



## UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

Departamento de Ciencias de la Computación

Pruebas Unitarias

Pruebas del Sistema

Sangolquí, Ecuador

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Objetivo de las Pruebas Unitarias</b>	<b>2</b>
<b>3. Alcance</b>	<b>2</b>
<b>4. Estrategia de Pruebas Unitarias</b>	<b>3</b>
<b>5. Paso a Paso para la Implementación de Pruebas Unitarias</b>	<b>3</b>
5.1. Paso 1: Identificación de Unidades . . . . .	3
5.2. Paso 2: Preparación del Entorno . . . . .	3
5.3. Paso 3: Diseño del Caso de Prueba . . . . .	3
5.4. Paso 4: Ejecución de las Pruebas . . . . .	4
5.5. Paso 5: Análisis de Resultados . . . . .	4
<b>6. Ejemplo de Caso de Prueba Unitaria</b>	<b>4</b>
<b>7. Métricas de Pruebas Unitarias</b>	<b>4</b>
<b>8. Conclusión</b>	<b>5</b>

## Índice de figuras

## Índice de cuadros

## 1 Introducción

El presente informe documenta el proceso de diseño, ejecución y análisis de las pruebas unitarias aplicadas al sistema **El Granito**, una plataforma financiera orientada a la gestión de créditos, pagos electrónicos y generación de reportes.

Las pruebas unitarias se basan en los flujos de actividades definidos para cada rol del sistema y tienen como objetivo validar el correcto funcionamiento de los componentes individuales antes de su integración con otros módulos.

## 2 Objetivo de las Pruebas Unitarias

El objetivo principal de las pruebas unitarias es:

- Verificar que cada función o método del sistema cumpla con los requisitos definidos.
- Detectar errores de forma temprana en la lógica del negocio.
- Asegurar la correcta validación de datos de entrada.
- Facilitar el mantenimiento y evolución del sistema.

## 3 Alcance

Las pruebas unitarias cubren los siguientes módulos del sistema:

- Autenticación y recuperación de contraseña
- Gestión de usuarios
- Gestión de créditos
- Procesamiento de pagos
- Generación de certificados y reportes

No se incluyen en este informe pruebas de integración, pruebas de rendimiento ni pruebas de aceptación.

## 4 Estrategia de Pruebas Unitarias

Las pruebas unitarias se ejecutaron de forma automatizada utilizando frameworks de pruebas según la tecnología de la aplicación (por ejemplo, Jest para frontend o JUnit para backend).

Cada prueba valida una única funcionalidad bajo el principio **AAA**:

- **Arrange:** Preparación de datos y contexto.
- **Act:** Ejecución del método a probar.
- **Assert:** Verificación del resultado esperado.

## 5 Paso a Paso para la Implementación de Pruebas Unitarias

### 5.1 Paso 1: Identificación de Unidades

Se identifican las funciones críticas del sistema a partir de los diagramas de flujo de actividades, por ejemplo:

- Validación de credenciales de usuario
- Cálculo de saldo pendiente de un crédito
- Registro de un pago

### 5.2 Paso 2: Preparación del Entorno

- Configurar el framework de pruebas.
- Crear una carpeta específica para pruebas (`/tests`).
- Aislar dependencias externas mediante mocks.

### 5.3 Paso 3: Diseño del Caso de Prueba

Cada caso de prueba debe incluir:

- Entrada esperada
- Resultado esperado
- Condición de validación

### Ejemplo: Validación de contraseña

- Entrada: contraseña con menos de 8 caracteres
- Resultado esperado: rechazo del sistema

## 5.4 Paso 4: Ejecución de las Pruebas

Las pruebas se ejecutan de forma automática mediante comandos del framework de pruebas. Los resultados se registran indicando pruebas exitosas y fallidas.

## 5.5 Paso 5: Análisis de Resultados

Las pruebas fallidas se analizan para identificar:

- Error lógico
- Validación incorrecta
- Falta de cobertura

Los defectos detectados se registran en la herramienta de gestión de incidencias.

## 6 Ejemplo de Caso de Prueba Unitaria

Elemento	Descripción
ID del Caso	UT-LOGIN-01
Módulo	Autenticación
Descripción	Validar longitud mínima de contraseña
Entrada	Contraseña de 4 caracteres
Resultado Esperado	El sistema rechaza la contraseña
Resultado Real	El sistema acepta la contraseña
Estado	Fallido
Acción	Registro de bug en el sistema de incidencias

## 7 Métricas de Pruebas Unitarias

- Casos unitarios planificados: 150
- Casos ejecutados: 145

- Casos exitosos: 140
- Casos fallidos: 5
- Cobertura de código: 82 %

## 8 Conclusión

Las pruebas unitarias permitieron identificar defectos críticos en etapas tempranas del desarrollo, especialmente en validaciones de seguridad y reglas de negocio.

El nivel de cobertura alcanzado cumple con los criterios establecidos, por lo que el sistema se considera estable a nivel unitario y apto para avanzar a pruebas de integración.