

Ámbito del Software PGC “Ingeniería del software un enfoque practico – Roger S.  
Pressman 7ma edición”

Julian Esteban Ballesteros Ortiz

[jestebanballesteros@ucundinamarca.edu.co](mailto:jestebanballesteros@ucundinamarca.edu.co)

Juan Diego Walteros Cortes

[jdiego@ucundinamarca.edu.co](mailto:jdiego@ucundinamarca.edu.co)

Ingeniera de Sistemas y Computación

Universidad de Cundinamarca

Seccional Ubaté

El ámbito de software del proyecto de gestión y emisión de documentación y facturación electrónica para el Régimen Simple de Tributación en Colombia incluye los siguientes aspectos técnicos y metodológicos:

### **Tecnologías Fundamentales:**

- 1. XML:** El sistema se basa en la generación de facturas electrónicas en formato XML, cumpliendo con los requisitos de la DIAN.
- 2. Sello digital:** Se utiliza un sello digital para la firma de las facturas electrónicas, garantizando la autenticidad de los documentos.
- 3. Infraestructura de clave pública (PKI):** Para la verificación de las firmas digitales, el sistema hace uso de una infraestructura de clave pública.
- 4. Almacenamiento en la nube:** Las facturas y demás documentos se almacenan en la nube, asegurando su disponibilidad, redundancia y seguridad.

### **Metodología de Desarrollo:**

El proyecto sigue un enfoque de desarrollo ágil utilizando Scrum, con iteraciones llamadas "Sprints" de dos semanas de duración. En cada Sprint, se desarrollan incrementos funcionales del sistema, permitiendo una retroalimentación temprana por parte de los usuarios y la adaptación a requisitos cambiantes.

### **Componentes del Sistema:**

- 1. Frontend:** Desarrollo de la interfaz de usuario (UI) usando HTML, CSS y JavaScript, diseñada en base a principios de usabilidad. Se trabaja con Visual Studio Code para el diseño del frontend.
- 2. Backend:** Implementado en paralelo con el frontend, se encarga de la lógica de negocio, validaciones y la generación de facturas. Además, garantiza la seguridad y la integridad de la información mediante la validación de datos como el formato correcto del NIT.

**3. Base de Datos:** Se ha creado un prototipo de base de datos en MySQL Workbench, que contiene tablas para la gestión de clientes, productos, facturas y ítems de factura. Esta estructura permite la organización y manejo eficiente de la información financiera del sistema.

### **Seguridad:**

El sistema está diseñado para ser seguro, protegiendo los datos sensibles a través de técnicas como la firma digital y el uso de la nube para almacenamiento seguro. También se implementan validaciones estrictas en los datos de entrada para evitar errores y garantizar la conformidad con los estándares de la DIAN.

### **Resultados esperados:**

El sistema mejorará la eficiencia y productividad de los empresarios colombianos, permitiéndoles generar facturas de manera sencilla, rápida y segura, cumpliendo con los estándares regulatorios exigidos por la DIAN

Pruebas de software Capítulo 17 “Ingeniería del software un enfoque practico –  
Roger S. Pressman 7ma edición”

Julian Esteban Ballesteros Ortiz  
jestebanballesteros@ucundinamarca.edu.co  
Juan Diego Walteros Cortes  
jdiegowalteros@ucundinamarca.edu.co

Universidad de Cundinamarca  
Seccional Ubaté

Semestre VI

Para evaluar qué pruebas de software son aplicables al proyecto de desarrollo de facturación electrónica y gestión documental bajo el Régimen Simple de Tributación, revisaremos las pruebas descritas en el capítulo 17 del libro "Ingeniería del Software: Un Enfoque Práctico".

### 1. Pruebas de Unidad:

- **Aplicabilidad:** Las pruebas de unidad son altamente aplicables. Este tipo de pruebas verifica que cada componente individual (como funciones, métodos o clases) funcione correctamente de manera aislada. Dado que el proyecto involucra operaciones críticas como cálculos financieros y manejo de documentos, es esencial asegurarse de que cada componente funcione correctamente antes de integrarlo.
- **Razón:** Ayuda a identificar errores en el nivel más bajo de código, lo que permite una detección temprana y corrección de errores.
- **Por qué no aplica:** No hay una razón clara para no aplicarla, ya que las pruebas de unidad son fundamentales para garantizar la calidad del software.

### 2. Pruebas de Integración:

- **Aplicabilidad:** Estas pruebas también son aplicables al proyecto. Verifican que los diferentes módulos del sistema interactúen correctamente. Dado que el proyecto involucra múltiples módulos como gestión de usuarios, emisión de facturas y almacenamiento de documentos, es crucial asegurarse de que todos estos módulos funcionen juntos sin problemas.
- **Razón:** Permite detectar problemas en las interacciones entre componentes, lo cual es crucial en un sistema que depende de la correcta comunicación entre varias partes.
- **Por qué no aplica:** No hay razones claras para no aplicar este tipo de prueba en el proyecto.

### 3. Pruebas de Sistema:

- **Aplicabilidad:** Las pruebas de sistema son completamente aplicables. Estas pruebas evalúan el sistema completo y su comportamiento en un entorno de producción simulado. Es esencial para verificar que el sistema cumpla con todos los requisitos especificados y funcione correctamente en condiciones reales de uso.

- **Razón:** Asegura que el sistema en su totalidad, con todas sus interacciones y dependencias, funcione según lo esperado.
- **Por qué no aplica:** Sería inapropiado no aplicarla, ya que es la forma de asegurarse de que el producto final cumpla con los requisitos.

#### 4. Pruebas de Regresión:

- **Aplicabilidad:** Las pruebas de regresión son aplicables. Estas pruebas verifican que los cambios recientes en el código no han introducido nuevos errores en partes ya probadas del sistema. En un proyecto en evolución como este, donde se esperan frecuentes actualizaciones y cambios, es vital realizar pruebas de regresión para mantener la estabilidad.
- **Razón:** Garantiza que nuevas actualizaciones no rompan las funcionalidades existentes.
- **Por qué no aplica:** No aplicarlas sería un riesgo, ya que podría comprometer la estabilidad del sistema.

#### 5. Pruebas de Validación:

- **Aplicabilidad:** Las pruebas de validación son aplicables. Verifican que el software cumpla con las necesidades y expectativas del cliente. En este proyecto, es crucial validar que el sistema de facturación y gestión documental funcione según los requisitos específicos del Régimen Simple de Tributación en Colombia.
- **Razón:** Asegura que el producto final realmente cumple con los requerimientos y expectativas del usuario.
- **Por qué no aplica:** No existen razones para no aplicar estas pruebas.

#### 6. Pruebas de Aceptación:

- **Aplicabilidad:** Son aplicables y necesarias. Estas pruebas son realizadas por el cliente para determinar si acepta el producto final. Para este proyecto, es crucial que los usuarios finales validen que el sistema cumple con sus necesidades de facturación y gestión documental antes de su implementación.
- **Razón:** Proveen una confirmación final de que el sistema está listo para ser usado en un entorno de producción.

- **Por qué no aplica:** No se justifica omitirlas, ya que son la última etapa antes de la implementación.

## **7. Pruebas de Rendimiento:**

- **Aplicabilidad:** Estas pruebas son parcialmente aplicables. Aunque no se menciona explícitamente en el proyecto, evaluar el rendimiento es crucial si se espera que el sistema maneje un gran volumen de datos o solicitudes.
- **Razón:** Es útil para asegurarse de que el sistema funcione bien bajo cargas esperadas de uso.
- **Por qué no aplica:** Si el sistema no requiere un rendimiento elevado o no se espera que maneje grandes volúmenes de datos, se podrían omitir estas pruebas.

## **8. Pruebas de Seguridad:**

- **Aplicabilidad:** Son muy aplicables. Este proyecto maneja datos sensibles como información financiera y documentos legales, por lo que la seguridad es una preocupación clave.
- **Razón:** Asegura que el sistema protege adecuadamente los datos y resiste ataques malintencionados.
- **Por qué no aplica:** No aplicarlas sería un error grave, dado el tipo de datos que se manejan

Plan de calidad del PGC “Ingeniería del software un enfoque practico – Roger S.  
Pressman 7ma edición”

Julian Esteban Ballesteros Ortiz  
jestebanballesteros@ucundinamarca.edu.co  
Juan Diego Walteros Cortes  
jdiegowalteros@ucundinamarca.edu.co

Universidad de Cundinamarca  
Seccional Ubaté

Semestre VI



## **Plan de Calidad**

Este plan de calidad tiene como objetivo garantizar que el desarrollo del sistema de facturación electrónica cumpla con los requisitos normativos establecidos por la DIAN bajo el Régimen Simple de Tributación en Colombia. El plan describe las estrategias, métricas y controles necesarios para asegurar la calidad del producto y los servicios relacionados, con énfasis en la eficiencia, seguridad y cumplimiento normativo.

## **Alcance**

El plan de calidad se aplica a todas las etapas del desarrollo del sistema, desde el diseño hasta la implementación final y la capacitación a los usuarios. Se incluyen los procesos de desarrollo de software, la creación de interfaces gráficas, la seguridad de la información y la emisión de documentación ante la DIAN.

## **Objetivos**

- Garantizar el cumplimiento de las normativas vigentes en materia de facturación electrónica bajo el Régimen Simple de Tributación.
- Desarrollar un software seguro y eficiente que asegure la correcta emisión de documentos electrónicos.
- Asegurar que el sistema se mantenga dentro de los estándares de calidad esperados, tanto en rendimiento como en funcionalidad.

## **Estrategias**

- Pruebas unitarias y de integración: Se realizarán pruebas continuas a cada módulo del sistema para identificar errores tempranamente.
- Revisión de código: El equipo de desarrollo revisará periódicamente el código para garantizar la calidad técnica.
- Auditorías de seguridad: Se realizarán evaluaciones periódicas para asegurar que los datos de las empresas y de la DIAN estén protegidos.
- Capacitación a usuarios: Se brindarán capacitaciones detalladas sobre el uso del sistema para garantizar su correcta implementación.

## **Métricas de Calidad**

- Tasa de errores detectados en producción.
- Tiempo de respuesta del sistema ante consultas y operaciones de facturación.

- Número de incidentes de seguridad reportados.
- Nivel de satisfacción de los usuarios.

### **Plan de Mejoramiento de Riesgo**

Tiene como objetivo identificar, evaluar y mitigar los riesgos que puedan afectar el éxito del proyecto de implementación del sistema de facturación electrónica bajo el Régimen Simple de Tributación. Se busca garantizar la seguridad, la continuidad operativa y el cumplimiento normativo.

### **Identificación de Riesgos**

Los riesgos identificados en el proyecto incluyen:

- Riesgos técnicos: Fallos en la infraestructura tecnológica que impidan el correcto funcionamiento del sistema.
- Riesgos de seguridad: Amenazas relacionadas con la filtración de datos o accesos no autorizados a la plataforma.
- Riesgos de cumplimiento normativo: Cambios en las regulaciones de la DIAN que puedan afectar el desarrollo o la operación del sistema.
- Riesgos operativos: Fallos durante la emisión de facturas o en el proceso de contabilidad que perjudiquen la experiencia de los usuarios.

### **Análisis de Riesgos**

Cada riesgo será evaluado en términos de probabilidad e impacto. Se dará prioridad a aquellos riesgos con mayor potencial de afectar la operatividad y la seguridad del sistema.

### **Estrategias de Mitigación**

- Riesgos técnicos: Se establecerá un protocolo de mantenimiento continuo y actualizaciones periódicas del software.
- Riesgos de seguridad: Implementación de un sistema de encriptación robusto y autenticación de dos factores para los usuarios.
- Riesgos de cumplimiento normativo: Monitoreo continuo de las actualizaciones legales y normativas en materia de facturación electrónica.
- Riesgos operativos: Capacitación a los usuarios y soporte técnico continuo para resolver cualquier eventualidad en el uso del sistema.

## **Plan de Acción**

- Desarrollar un sistema de copias de seguridad automáticas para prevenir la pérdida de datos.
- Realizar simulaciones periódicas de ataques cibernéticos para probar la resiliencia del sistema.
- Mantener un equipo de respuesta rápida para solucionar fallos en la emisión de documentos.

## **Monitoreo y Control**

Se establecerán controles y revisiones trimestrales del estado de los riesgos y la efectividad de las medidas implementadas, con la participación de todos los actores involucrados en el desarrollo del proyecto

Análisis de factibilidad y plan de pruebas de software “Ingeniería del software un enfoque practico – Roger S. Pressman 7ma edición”

Julian Esteban Ballesteros Ortiz  
jestebanballesteros@ucundinamarca.edu.co  
Juan Diego Walteros Cortes  
jdiegowalteros@ucundinamarca.edu.co

Universidad de Cundinamarca  
Seccional Ubaté

Semestre VI

## 1. Viabilidad Técnica

El proyecto de facturación electrónica, contabilidad, y emisión de documentación ante la DIAN para el Régimen Simple de Tributación (RST) en Colombia presenta una viabilidad técnica sólida. La implementación del sistema utiliza tecnologías modernas como bases de datos MySQL y un enfoque de desarrollo ágil mediante la metodología Scrum. La estructura del proyecto en GitHub muestra un manejo organizado del código fuente, diagramas, y demás recursos del proyecto, lo cual facilita el control de versiones y la colaboración entre los miembros del equipo.

El sistema ha sido diseñado para ser intuitivo y accesible, incluso para empresarios sin conocimientos técnicos especializados, cumpliendo con los requisitos legales establecidos por la DIAN, lo que asegura su aplicabilidad en el contexto colombiano. Además, el desarrollo de un prototipo de base de datos demuestra la capacidad del sistema para gestionar eficientemente datos críticos como facturas, ítems de facturas, y datos de clientes.

## 2. Factores Económicos

El proyecto promete mejorar la eficiencia y productividad de pequeños empresarios y emprendedores, reduciendo costos y tiempo asociados al cumplimiento de los requisitos técnicos y legales para la facturación electrónica. La simplificación de estos procesos podría traducirse en ahorros significativos para los usuarios del sistema, especialmente en sectores donde el cumplimiento normativo es crucial para la operación del negocio.

## 3. Evaluación de Riesgos

Existen riesgos asociados a la implementación de este sistema, principalmente en cuanto a la seguridad de la información. Dado que el sistema manejará datos sensibles como información financiera y personal, la implementación de medidas de seguridad robustas es esencial. Cualquier brecha en la seguridad podría no solo comprometer la información de los usuarios, sino también la credibilidad del sistema.

Otro riesgo potencial es la resistencia al cambio por parte de los usuarios que no estén familiarizados con la tecnología. Este riesgo puede mitigarse a

través de capacitación y soporte continuo para los usuarios del sistema, asegurando una transición suave hacia el uso del sistema propuesto.

#### 4. Consideraciones Legales

El sistema está diseñado para cumplir con todos los requisitos técnicos y legales establecidos por la DIAN, lo cual es fundamental para su viabilidad. La integración con los formularios de la DIAN y la generación de facturas electrónicas en formatos aprobados son aspectos críticos que han sido abordados en el diseño del sistema.

## Plan de pruebas de Software.

### 1. Introducción

El propósito de este plan de pruebas es garantizar que el sistema de facturación electrónica, contabilidad y emisión de documentación bajo el Régimen Simple de Tributación en Colombia cumpla con los requisitos funcionales y no funcionales. Se verificará la estabilidad, funcionalidad, rendimiento, seguridad y usabilidad del sistema.

### 2. Alcance

Este plan cubre las pruebas para el desarrollo del sistema mencionado, incluyendo las siguientes funcionalidades clave:

- Generación y emisión de facturas electrónicas.
- Validación de firmas digitales y XML.
- Gestión de documentación contable.
- Cumplimiento de los estándares de la DIAN.
- Almacenamiento y consulta en la nube.

### 3. Estrategia de Pruebas

#### 3.1. Tipos de Pruebas

- Pruebas Funcionales: Verifican si cada funcionalidad del sistema cumple con los requisitos definidos.
- Casos de uso: Generación de facturas, validación de firma digital, emisión de reportes contables.
- Tecnologías: XML, Firma Digital, Cloud Storage.
- Pruebas de Seguridad: Se asegurarán de que la información de las facturas y la documentación contable esté protegida.
- Casos de uso: Prueba de inyecciones SQL, pruebas de acceso no autorizado.
- Tecnologías: Cifrado de datos, autenticación de usuario.
- Pruebas de Rendimiento: Evalúan el rendimiento del sistema bajo diferentes cargas de trabajo.
- Casos de uso: Emisión simultánea de 100 facturas.
- Indicadores: Tiempo de respuesta, uso de CPU y memoria.
- Pruebas de Usabilidad: Se evaluará la facilidad de uso de

la interfaz del sistema por usuarios sin conocimientos técnicos.

- Casos de uso: Generación de factura por usuario sin experiencia.
- Métrica: Satisfacción del usuario, tasa de errores.
- Pruebas de Integración: Verifican que los módulos de facturación, contabilidad y gestión documental trabajen correctamente en conjunto.
- Casos de uso: Integración entre generación de facturas y actualización de contabilidad.
- Pruebas de Compatibilidad: Aseguran que el sistema funcione en múltiples navegadores y dispositivos.
- Casos de uso: Prueba en Chrome, Firefox y dispositivos móviles.

#### 4. Casos de Prueba

##### 4.1. Caso de Prueba 001: Generación de Factura Electrónica

- Descripción: Validar que el sistema permite generar una factura con la información completa.
- Precondiciones: Usuario autenticado.
- Entradas: Datos del cliente, productos y cantidades.
- Resultado Esperado: Generación de una factura válida en formato XML.

##### 4.2. Caso de Prueba 002: Validación de Firma Digital

- Descripción: Asegurar que las facturas estén firmadas digitalmente.
- Precondiciones: Factura generada.
- Entradas: Factura sin firmar.
- Resultado Esperado: Factura firmada digitalmente conforme a los estándares DIAN.

##### 4.3. Caso de Prueba 003: Prueba de Seguridad - Inyección SQL

- Descripción: Validar que el sistema es resistente a ataques de inyección SQL.
- Precondiciones: Usuario autenticado.
- Entradas: Consulta con parámetros SQL.



- Resultado Esperado: El sistema no debe ejecutar comandos maliciosos.

#### 4.4. Caso de Prueba 004: Prueba de Usabilidad

- Descripción: Evaluar si un usuario sin experiencia puede generar una factura sin errores.
- Precondiciones: Usuario registrado y autenticado.
- Entradas: Datos del cliente y productos.
- Resultado Esperado: Factura generada correctamente sin asistencia adicional.

#### 5. Herramientas de Prueba

- JMeter: Para pruebas de rendimiento.
- OWASP ZAP: Para pruebas de seguridad.
- Selenium: Para pruebas de interfaz.
- JUnit: Para pruebas unitarias.

#### 6. Criterios de Aceptación

- Al menos el 95% de los casos de prueba deben ser exitosos.
- No deben presentarse vulnerabilidades críticas de seguridad.
- El sistema debe manejar adecuadamente 100 usuarios concurrentes.

#### 7. Cronograma

- Preparación del entorno: 1 semana.
- Ejecución de pruebas funcionales: 2 semanas.
- Ejecución de pruebas de rendimiento: 1 semana.
- Pruebas de seguridad: 1 semana.
- Pruebas de integración y usabilidad: 1 semana.

#### 8. Conclusión

Este plan de pruebas proporciona un enfoque exhaustivo y riguroso para asegurar que el sistema de facturación electrónica bajo el Régimen Simple de Tributación sea funcional, seguro y eficiente para su uso en un entorno de producción. Este plan se ajustará a medida que el sistema evolucione y se reciban nuevas especificaciones o requerimientos