# PLANTEAMIENTO DEL PROYECTO

Estudiante

SNEIDER MONROY QUIROGA

STEPHAN VARGAS QUIROGA

JHOAN SEBASTIAN NARANJO QUIROGA

JUAN DAVID ROJAS PIINEDA

CORPORACION UNIVERSITARIA DEL HUILA

PROGRAMA INGENIERÍA DE SISTEMAS

INGENIERO: JESUS ARIEL GONZALEZ BONILLA

NEIVA – HUILA

02 DE SEPTIEMBRE DEL AÑO 2023

# PLANTEAMIENTO DEL PROBLEMA

En el contexto de una empresa de servicios profesionales, se ha detectado una problemática relacionada con el proceso de facturación manual. Este procedimiento manual consume una cantidad significativa de tiempo y ha ocasionado errores frecuentes en las facturas, lo que a su vez ha generado retrasos en los pagos y una experiencia insatisfactoria para los clientes. Con el propósito de solucionar este problema y mejorar la eficiencia en el proceso de facturación, se plantea la necesidad de desarrollar un sistema de facturación automatizado.

**Instrumento de recolección de información**

Para obtener datos relevantes sobre las necesidades y requerimientos del sistema de facturación automatizado, se llevarán a cabo entrevistas con varios grupos de interés.

* **Personal de facturación:** Se realizarán entrevistas semiestructuradas para comprender en profundidad los desafíos específicos que enfrentan en el proceso actual de facturación y sus requerimientos en términos de automatización (Apellido, Inicial del Nombre, Fecha de la entrevista).
* **Clientes:** Se emplearán cuestionarios estructurados para recopilar información sobre su percepción de las facturas actuales, su claridad y precisión, y cómo una mejora en el proceso de facturación podría beneficiarlos (Apellido, Inicial del Nombre, Fecha de la encuesta).
* **Contadores o departamentos financieros:** Se realizarán entrevistas semiestructuradas para comprender las regulaciones fiscales y contables que deben cumplirse en las facturas y cómo el sistema puede cumplir con estas regulaciones (Apellido, Inicial del Nombre, Fecha de la entrevista).

**Las preguntas del sistema son las siguientes**

1.¿Preferiría recibir sus facturas por correo electrónico o en formato impreso?

Opciones: Correo electrónico/Impreso

2. ¿Le gustaría recibir notificaciones automáticas cuando se emita una nueva factura? (Opciones: Sí/No)

3. ¿Le gustaría poder personalizar el formato de sus facturas según sus preferencias? (Opciones: Sí/No)

**Requerimientos funcionales**

Con base en la información recopilada, se proponen los siguientes requerimientos funcionales para el sistema de facturación automatizado

1. **Generación de facturas electrónicas:** El sistema debe tener la capacidad de generar automáticamente facturas electrónicas a partir de los datos del servicio prestado, incluyendo detalles del cliente, descripción de los servicios, tarifas y fechas.
2. **Integración con bases de datos de clientes:** Debe permitir la integración con una base de datos de clientes para recuperar información relevante, como direcciones y datos de contacto.
3. **Cálculo automático de impuestos:** El sistema debe realizar automáticamente el cálculo de los impuestos aplicables de acuerdo con las regulaciones fiscales vigentes y agregarlos a la factura.
4. **Gestión de pagos:** Deberá llevar un registro de los pagos recibidos y permitir la reconciliación de pagos con facturas pendientes.
5. **Envío automático de facturas:** El sistema debe contar con la capacidad de enviar automáticamente las facturas por correo electrónico a los clientes y almacenar copias en un archivo digital.
6. **Generación de informes financieros**: Debe ofrecer la capacidad de generar informes financieros, como resúmenes de ingresos, cuentas por cobrar y cuentas por pagar.
7. **Seguridad de datos:** Es fundamental garantizar la seguridad de los datos del cliente y la confidencialidad de la información financiera.
8. **Facilidad de uso:** Se debe diseñar una interfaz intuitiva y amigable para el usuario, de modo que el personal de facturación pueda utilizarlo sin problemas.

**Atributos del sistema**

## **Seguridad:**

1. Confidencialidad: El sistema debe garantizar que los datos confidenciales, como la información financiera del cliente, estén protegidos y solo sean accesibles por usuarios autorizados.
2. Integridad: Debe asegurar que los datos no se corrompan ni se alteren de manera no autorizada.
3. Autenticación y Autorización: Debe verificar la identidad de los usuarios y otorgarles acceso según sus roles y permisos.

## **Rendimiento:**

1. Tiempo de Respuesta: El sistema debe ser rápido y eficiente al realizar operaciones, como la generación de facturas y la búsqueda de registros.
2. Escalabilidad: Debe ser capaz de manejar un aumento en el volumen de datos y usuarios sin degradación significativa del rendimiento.

## **Disponibilidad:**

1. El sistema debe estar disponible durante las horas de operación previstas y debe minimizar el tiempo de inactividad no planificado.

**Confiabilidad:**

1. El sistema debe ser confiable y funcionar correctamente sin errores críticos que puedan afectar la operación normal.

**Mantenibilidad:**

1. Debe ser fácil de mantener y actualizar, lo que implica una estructura de código modular y documentación clara.

**Facilidad de Uso:**

1. La interfaz de usuario debe ser intuitiva y amigable, lo que facilita la capacitación de nuevos usuarios y minimiza los errores de entrada de datos.

**Escalabilidad:**

1. Debe ser capaz de crecer y adaptarse a las cambiantes necesidades del negocio, lo que incluye la capacidad de agregar nuevos módulos o funcionalidades según sea necesario.

**Interoperabilidad:**

1. Debe ser compatible con otros sistemas o aplicaciones que la empresa utilice, como sistemas contables o de gestión de clientes.

**Cumplimiento Legal:**

1. Debe cumplir con las regulaciones fiscales y contables locales, asegurando que las facturas generadas cumplan con las leyes vigentes.

**Aseguramiento de la Calidad**:

1. Debe incluir pruebas rigurosas para garantizar que el software funcione según lo previsto y sea libre de errores significativos.

**Adaptabilidad:**

1. Debe ser capaz de adaptarse a cambios en las regulaciones fiscales o contables sin interrupciones importantes en la operación.

**Cronograma**

**Fase 1: requisitos y análisis**

1. Semanas 1-2: Definición de requisitos

- Entrevistar al personal de facturación, clientes y contadores para obtener y documentar los requisitos del sistema.

2. Semanas 3-4: Análisis de necesidades

- Analizar y validar los requisitos obtenidos.

- Documentar casos de uso y requisitos funcionales y no funcionales.

3. Semanas 5-6: revisión y aprobación de requisitos

- Revisar y obtener la aprobación de los requisitos por parte de las partes interesadas.

**Fase 2: Diseño**

1. Semanas 7-8: Diseño de Arquitectura

- Diseñar la arquitectura del sistema e identificar módulos y componentes clave.

2. Semanas 9-10: Diseño detallado

- Realizar diseño detallado de diversos componentes y módulos del sistema.

- Crear diagramas UML y documentación técnica. 3. Semanas 11-12: Revisión y aprobación del diseño

- Revisar y obtener la aprobación de los diseños por parte de las partes interesadas.

**Fase 3: Implementación**

1. Semanas 13-16: Codificación

- Implementar el sistema según el diseño aprobado.

2. Semana 17-18: pruebas unitarias y corrección de errores

- Realizar pruebas unitarias y corregir errores en el código.

3. Semanas 19-20: Integración y prueba del sistema

- Integrar componentes y realizar pruebas del sistema.

**Fase 4: Pruebas**

1. Semanas 21-22: Pruebas de Validación

- Realizar pruebas de validación para garantizar que el sistema cumpla con los requisitos del usuario.

2. Semana 23-24: Pruebas de usuario

- Invitar a usuarios clave a realizar pruebas y recopilar comentarios.

**Fase 5: Implementación y Mantenimiento**

1. Semana 25-26: Implementación del sistema

- Implementar el sistema en un entorno de producción.

2. Semanas 27-28: Capacitación del personal

- Capacitar al personal de facturación y otros usuarios clave.

3. Semanas 29-30: Evaluación final y mejora

- Evaluar el desempeño del sistema en producción.

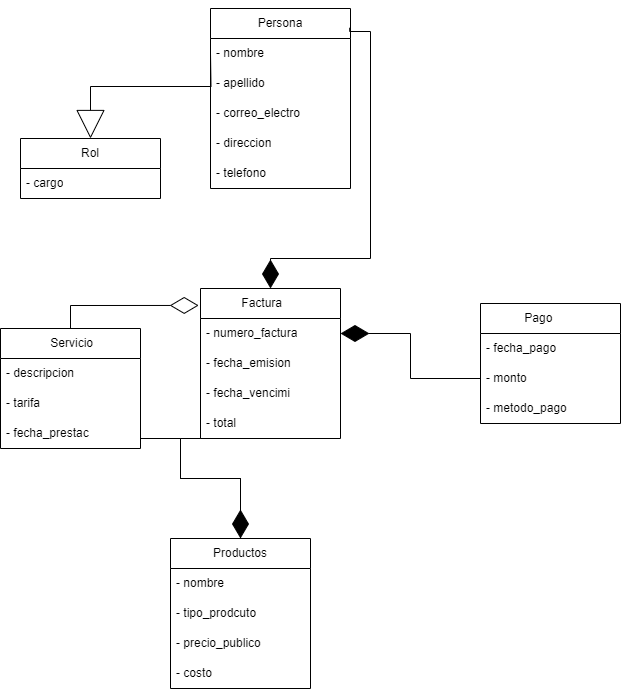
- Mejoras y correcciones según sea necesario.

4. Semanas 31-32: Documentación final

- Finalizar la documentación del sistema.

**Relaciones Entre Las Clases En El Sistema De Facturación:**

1. Persona - Rol:  
   -       Tipo de Relación: Herencia  
   -       Descripción: La clase "Rol" hereda  
   los atributos de la clase "Persona". Esta relación se expresa  
   mediante la relación de herencia, indicando que la clase "Rol" hereda  
   los atributos y comportamientos de la clase "Persona".
2. Factura - Cliente:  
   -       Tipo de Relación: Composición (1:1)  
   -       Descripción: La clase "Factura"  
   tiene una relación de composición con la clase "Cliente". Esto  
   significa que un objeto de la clase "Factura" está compuesto por un  
   objeto de la clase "Cliente". La existencia de una factura depende de  
   la existencia de un cliente, y si la factura se elimina, también se eliminará  
   el cliente asociado.
3. Factura - Servicio:  
   -       Tipo de Relación: Agregación (1:N)  
   -       Descripción: La clase "Factura"  
   tiene una relación de agregación con la clase "Servicio". Esto indica  
   que una factura puede contener varios servicios, pero la existencia de un  
   servicio no depende de la existencia de la factura. Puede haber servicios sin  
   estar asociados a una factura específica.
4. Factura - Pago:  
   -       Tipo de Relación: Composición (1:1)  
   -       Descripción: La clase "Factura"  
   tiene una relación de composición con la clase "Pago". Esto significa  
   que una factura está compuesta por un pago. La existencia de una factura  
   depende de la existencia de un pago, y si la factura se elimina, también se  
   eliminará el pago asociado.
5. Servicio - Factura:  
   -       Tipo de Relación: Asociación (1:1)  
   -       Descripción: La clase "Servicio"  
   tiene una relación de asociación con la clase "Factura". Esto indica  
   que un servicio está asociado a una factura específica, pero la existencia de  
   un servicio no depende de la existencia de la factura.
6. Pago - Factura:  
   -       Tipo de Relación: Composición (1:1)  
   -       Descripción: La clase "Pago" tiene  
   una relación de composición con la clase "Factura". Esto significa  
   que un pago está compuesto por una factura. La existencia de un pago depende de  
   la existencia de una factura, y si el pago se elimina, también se eliminará la  
   factura asociada



**Diagrama de Casos de Uso - Sistema de Facturación**

1. Actor: Usuario

- Casos de Uso:

- Consultar Factura

- Generar Factura

- Realizar Pago

- Gestionar Clientes

- Gestionar Servicios

1. Casos de Uso Detallados:

- Consultar Factura:

- Descripción: Permite al usuario consultar detalles de una factura existente.

- Actores: Usuario, Factura

- Flujo Principal:

\* El usuario selecciona la opción de "Consultar Factura".

\* El sistema solicita el número de factura.

\* El usuario ingresa el número de factura.

\* El sistema muestra los detalles de la factura.

- Generar Factura:

- Descripción: Permite al usuario generar una nueva factura.

- Actores: Usuario, Factura, Cliente, Servicio

- Flujo Principal:

\* El usuario selecciona la opción de "Generar Factura".

\* El sistema solicita la información del cliente y los servicios a incluir.

\* El usuario proporciona la información requerida.

\* El sistema crea la factura asociada al cliente y servicios.

- Realizar Pago:

- Descripción: Permite al usuario registrar un pago para una factura.

- Actores: Usuario, Pago, Factura

- Flujo Principal:

\* El usuario selecciona la opción de "Realizar Pago".

\* El sistema solicita el número de factura y la información del pago.

\* El usuario ingresa la información requerida.

\* El sistema registra el pago y actualiza el estado de la factura.

- Gestionar Clientes:

- Descripción: Permite al usuario agregar, modificar o eliminar clientes.

- Actores: Usuario, Cliente

- Flujo Principal:

\* El usuario selecciona la opción de "Gestionar Clientes".

\* El sistema muestra las opciones de agregar, modificar o eliminar clientes.

\* El usuario elige una opción y realiza las acciones correspondientes.

- Gestionar Servicios:

- Descripción: Permite al usuario agregar, modificar o eliminar servicios.

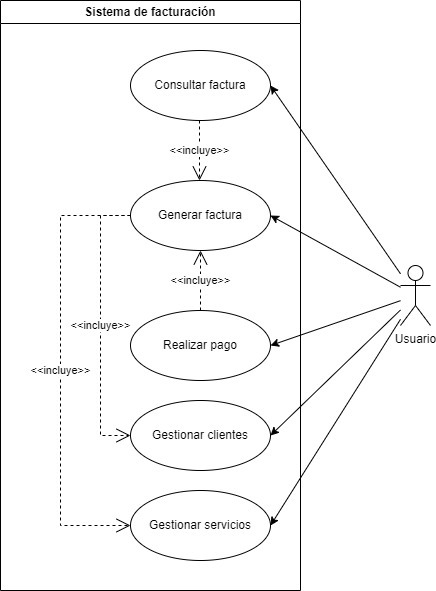
- Actores: Usuario, Servicio

- Flujo Principal:

\* El usuario selecciona la opción de "Gestionar Servicios".

\* El sistema muestra las opciones de agregar, modificar o eliminar servicios.

\* El usuario elige una opción y realiza las acciones correspondientes.

****