|  |  |
| --- | --- |
|  | **FACULTAD DE CIENCIAS NATURALES Y EXACTAS**  **DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**  **Trabajo de suficiencia**  **Ingeniería de Software**  **TERCER AÑO LICENCIATURA EN CIENCIA DE LA COMPUTACIÓN**  **Proyecto:** Recaudadora de impuestos  **Nombre**: Alejandro Fonseca Cuza  **Diciembre de 2024**  **Santiago de Cuba** |

**Situación Problémica**

El Estado cubano desea implementar un sistema para gestionar la información sobre la recaudación de un organismo administrador de impuestos. Para ello el administrador debe registrar las personas, las cuales se identifican por su cédula de identidad y de las cuales se registran su nombre y sus teléfonos. Los contribuyentes se identifican por el RUT (registro único tributario) y de ellos se conoce además la fecha de inicio de sus actividades. Éstos pueden ser personas naturales o jurídicas. Si es una persona natural se asocia a la persona correspondiente. Si es una empresa se registra su razón social y se registra que personas son titulares de la empresa. Una persona no puede ser titular de una empresa y estar registrado como persona natural a la vez. Una empresa debe tener siempre al menos un titular, pero puede tener más de uno y una persona puede ser titular de varias empresas. Para cada contribuyente se guarda un registro de cuáles son los impuestos que debe pagar. De los impuestos se recoge el número de formulario asociado (que lo identifica) y su descripción, por ejemplo: 1006, “IRAE”. Los contribuyentes abonan sus impuestos en entidades recaudadoras, las cuales se clasifican en locales de pago externos u oficinas del organismo. Los locales de pago se identifican por el nombre de empresa de cobranzas y el número de local, por ejemplo: hábitat nº10, y de cada local se conoce su dirección. Cada oficina del organismo pertenece a una dependencia. Las dependencias se identifican por departamento, existe una sola por cada departamento y de cada una de ellas se registra la cantidad de empleados de la misma. Las oficinas se identifican por número, pero este número se puede repetir entre distintas dependencias. Los económicos necesitan registrar los pagos de impuestos realizados por los contribuyentes en las distintas entidades recaudadoras en distintas fechas. De cada pago se registra el monto y la fecha de pago. Un contribuyente puede realizar varios pagos en distintas entidades recaudadoras para distintos impuestos en distintas fechas. Sin embargo, no existen dos pagos para el mismo contribuyente, el mismo impuesto en la misma entidad recaudadora para la misma fecha. Se desea un sistema de gestión de los datos anteriores, y que responda además a las consultas siguientes:

a) Listado de los contribuyentes (RUT, CI y Nombre) de las personas físicas que deben pagar todos los impuestos.

b) Listado de los contribuyentes de empresa que incluya el RUT y el total de personas titulares de la estas.

c) Listado de los contribuyentes (RUT, CI y Nombre) de las personas físicas que sólo han realizado pagos en los locales de pago externos.

d) Listado de los contribuyentes de empresa que incluya el RUT y el monto total de los pagos realizados en un período dado.

e) Los 3 primeros impuestos que más han recaudado en un período dado.

**Capítulo 1 “Marco Teórico Referencial”**

**Introducción del capítulo**

El presente documento se enfocará en el diseño de un sistema de gestión de información tributaria, considerando aspectos fundamentales como la identificación de los actores involucrados, la estructura necesaria para su registro y la interacción entre los diferentes elementos del sistema. Para ello, se emplearán principios de ingeniería de software y metodologías de desarrollo que garanticen la robustez, flexibilidad y escalabilidad del sistema, asegurando así su adecuado funcionamiento.

**Modelado de negocio**

**Actores del negocio**

* Contribuyentes

Justificación Contribuyentes: Son gestionados por el negocio y se lleva un registro con los impuestos que deben pagar.

**Trabajadores del negocio**

* Administrador
* Económico

Justificación Administrador: Es el encargado de gestionar el negocio.

Justificación Económico: Registra todos los pagos realizados por los contribuyentes.

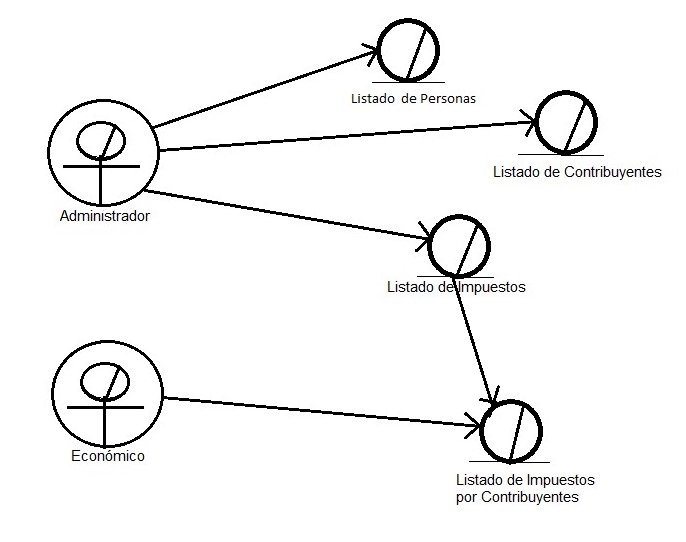
**Casos de uso del negocio**

* Registrar Personas
* Gestionar Personas
* Gestionar Contribuyentes
* Pagar Impuestos

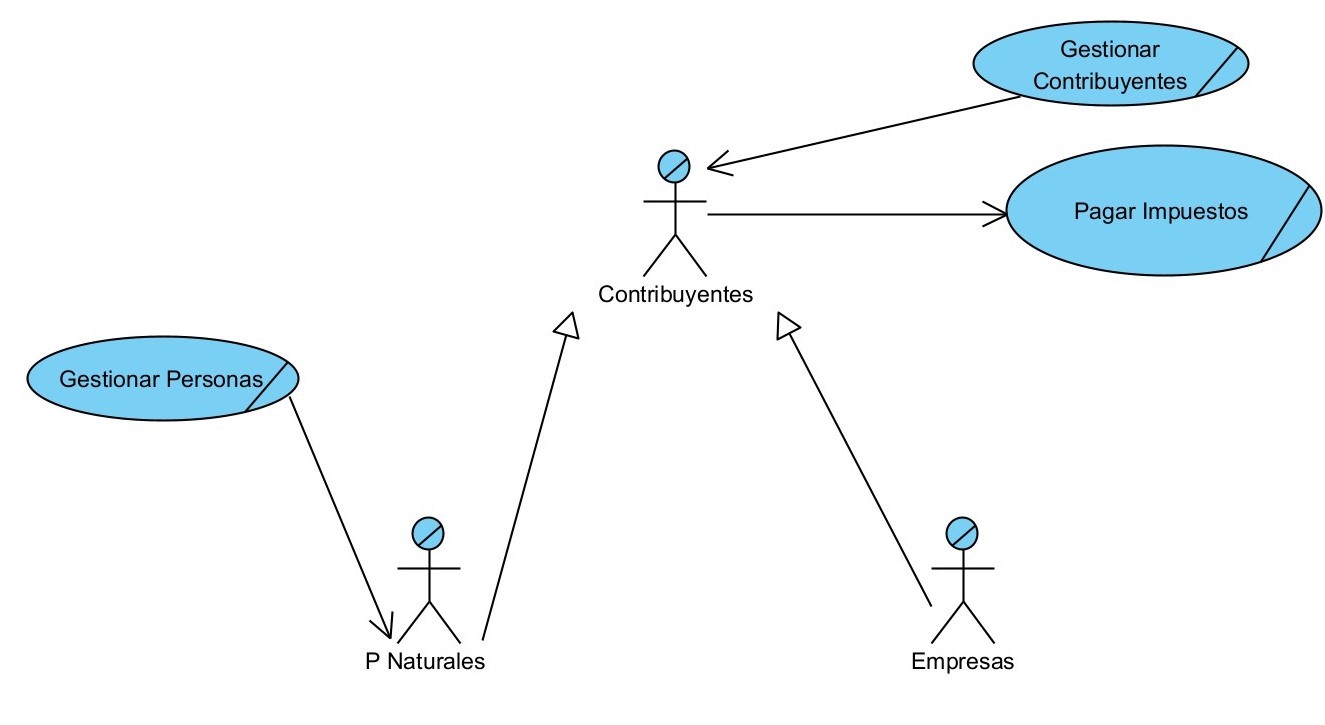
**Entidades del negocio**

* Listado de personas
* Listado de contribuyentes
* Listado de impuestos
* Listado de impuestos por contribuyentes

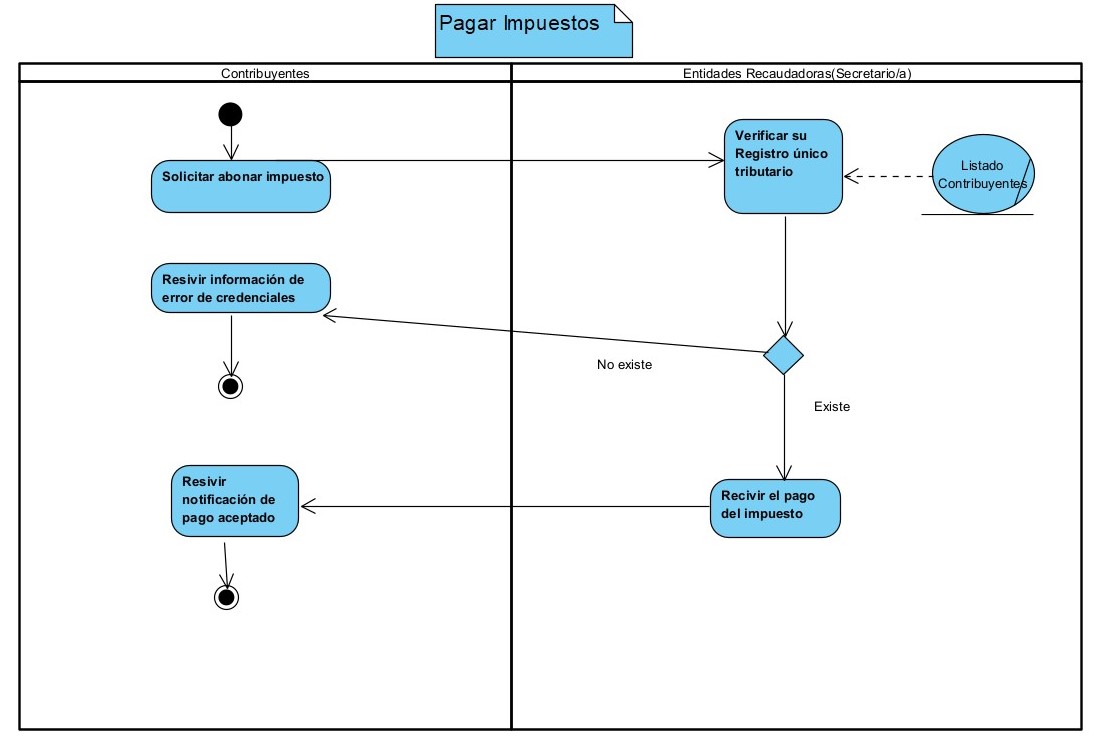
**Diagrama del modelo de objetos**



**Diagrama de casos de uso del negocio**



**Diagrama de actividades de un caso crítico del negocio**



**Herramientas utilizadas**

Vista:

1. Angular (Framework Javascript): Es un framework de desarrollo front-end que se utiliza para construir aplicaciones web de una sola página (SPA). Proporciona una estructura robusta para el desarrollo de interfaces de usuario dinámicas y responsivas.

2. Bootstrap (Framework Css): Es un framework de desarrollo front-end que proporciona una colección de herramientas y estilos CSS predefinidos para facilitar el diseño y la creación de interfaces de usuario modernas y receptivas.

3. HTML: Es el lenguaje estándar para crear páginas web. Se utiliza para definir la estructura y el contenido de una página web.

Servidor:

1. PHP: PHP es un lenguaje de programación de uso general especialmente adecuado para el desarrollo web. Se utiliza principalmente en el lado del servidor para crear aplicaciones web dinámicas e interactuar con bases de datos.

Base de datos:

1. MySQL: MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto. Es ampliamente utilizado en aplicaciones web para almacenar y administrar datos de manera eficiente y segura.

2. ER/Studio: ER/Studio es una herramienta de modelado de datos que permite diseñar, visualizar y mantener modelos de datos relacionales y multidimensionales. Se utiliza para crear y gestionar diagramas de entidad-relación (ER) y otros modelos de datos.

Pruebas:

1. Jasmine: Es un marco de pruebas de código abierto para realizar pruebas unitarias en aplicaciones JavaScript. Proporciona una sintaxis clara y fácil de usar para escribir y ejecutar pruebas automatizadas.

Otros:

1. Visual Paradigm: Es una herramienta de modelado UML (Lenguaje de Modelado Unificado) que se utiliza para diseñar y visualizar sistemas de software. Permite crear diagramas UML para representar la estructura, el comportamiento y la interacción de un sistema.

2. Nice Mind: Es una herramienta de mapeo mental que se utiliza para visualizar ideas, conceptos y relaciones de manera gráfica. Ayuda a organizar y estructurar la información de manera clara y concisa.

**Metodología Aplicada**

Para este proyecto se utilizó la metodología de software tradicional Rup ya que es la que se ha venido estudiando en la asignatura de Ingeniería de Software y entre sus principales características de este se encuentra.

- Se siguió estrictamente un plan

- Se definió la arquitectura tempranamente en el proyecto

-Énfasis en la definición del proceso: roles (actores), actividades (casos de uso) y artefactos (objetos).

-Se espera que no ocurran cambios de gran impacto durante el proyecto.

Aunque el desarrollo del software se realizó en un sprint de pequeña duración.

**Conclusiones del capítulo**

El diseño y desarrollo de un sistema de gestión tributaria requiere la aplicación de principios y metodologías de ingeniería de software. Esto implica la identificación precisa de los requisitos del sistema, el modelado de datos adecuado y la implementación de funcionalidades que satisfagan las necesidades específicas de los usuarios finales.

**Capítulo 2 “Desarrollo del Prototipo del Sistema”**

**Introducción del capítulo**

En este capítulo, se adentra en el proceso detallado de concepción, diseño, implementación y evaluación de un prototipo funcional del sistema en desarrollo. Desde la definición de requisitos hasta las pruebas realizadas a una versión inicial del sistema. Ofrece una visión integral de las etapas cruciales en el ciclo de vida del desarrollo de software.

**Requisitos funcionales**

* Administrar Personas
* Administrar Contribuyentes
* Administrar Entidades Recaudadoras
* Gestionar Impuestos para cada contribuyente
* Verificar titulares de empresa
* Listar Personas Naturales que deben pagar todos los impuestos
* Listar RUT de las empresas y la cantidad de titulares que tienen
* Listar Contribuyentes (Personas Naturales) que sólo pagan en locales de pago
* Listar Monto total pagado por las empresas en un período dado
* Listar los 3 primeros impuestos que más han recaudado en un período

**Requisitos no funcionales**

* Al ser un sistema tributario se espera que la aplicación cumpla con los estándares de seguridad actuales
* El software debe de ser preciso cuando opera con los montos de los impuestos
* Es un sistema gubernamental así que la eficiencia y la rapidez es algo primordial debido a la gran concurrencia de clientes

**Actores del sistema**

* Administrador
* Económicos

Justificación Administrador: Es el encargado de registrar y administrar las personas,

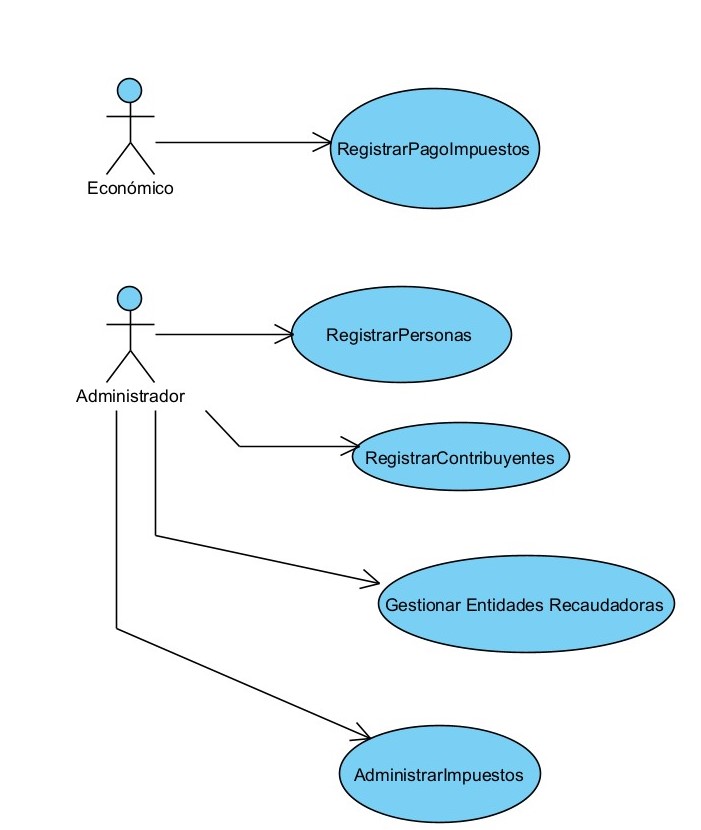
Contribuyentes, Entidades Recaudadoras.

Justificación Económicos: Se encarga de registrar los pagos de impuestos realizados por cada contribuyente en una Entidad Recaudadora

**Casos de uso del sistema y clasificación**

* Registrar Personas (crítico)
* Administrar Impuestos (crítico)
* Gestionar Entidades Recaudadoras (crítico)
* Registrar Contribuyentes (crítico)
* Registrar Pago Impuestos (crítico)

**Diagrama de casos de uso del sistema**

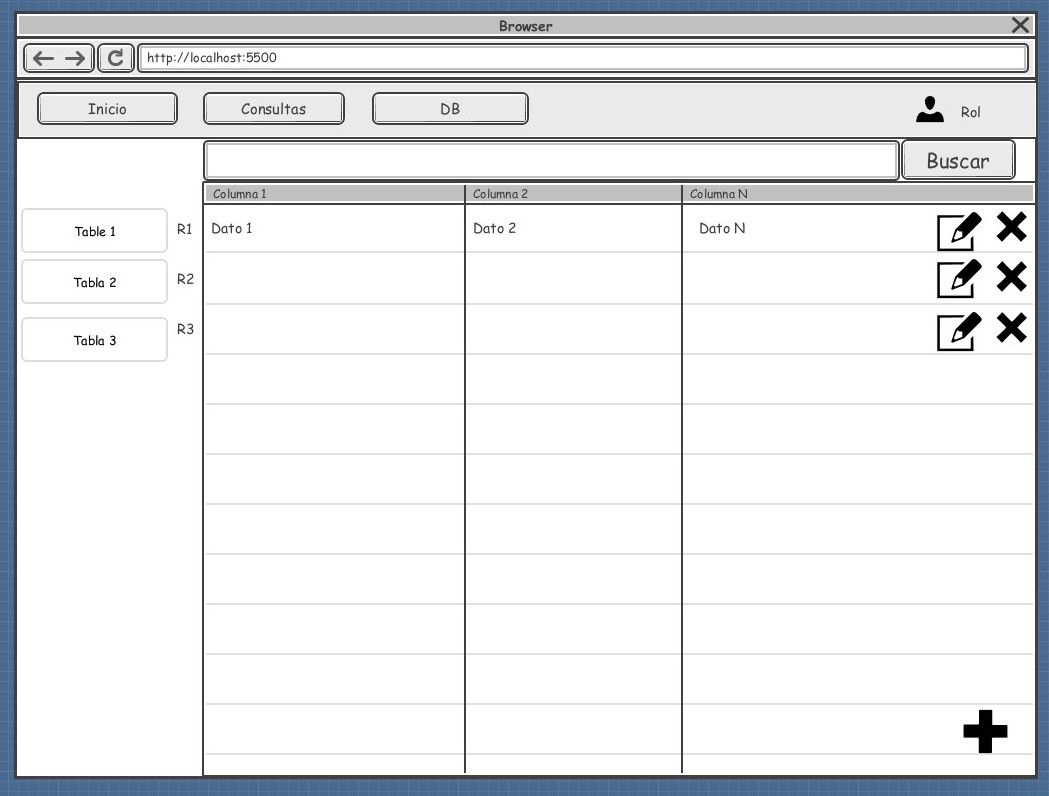


**Descripción en formato de alto nivel de un caso de uso crítico del sistema:**

|  |  |  |
| --- | --- | --- |
| **Caso de uso:** | **Registrar Contribuyentes** | |
| **Actores:** | **Administrador** | |
| **Descripción:**  El caso de uso comienza cuando el Administrador del sistema accede al menú Registrar contribuyentes. Introduce los datos del contribuyente, que incluyen RUT (registro único tributario) y de ellos se conoce además la fecha de inicio de sus actividades, y tipo de contribuyente si es persona natural o persona jurídica, si es una persona natural se le asocia sus credenciales con las personas registradas en el sistema y en caso de ser una empresa se registra su razón social y se registra que personas son titulares de la empresa. En caso de que los datos no sean válidos o no se puedan almacenar correctamente, se informa al contribuyente para que tome las medidas necesarias. | | |
| **Referencias:** | | - Requerimientos funcionales:  1. Registrar contribuyentes |
| **Precondiciones:** | | 1.El administrador se encuentra autenticado en el sistema. |
| **Pos condiciones:** | | Se almacena el contribuyente en la base de datos.  Si el contribuyente es una empresa se actualiza la tabla titular empresa con los nuevos datos. |
| **Requerimientos especiales** | |  |

**Prototipos de interfaz de usuario**

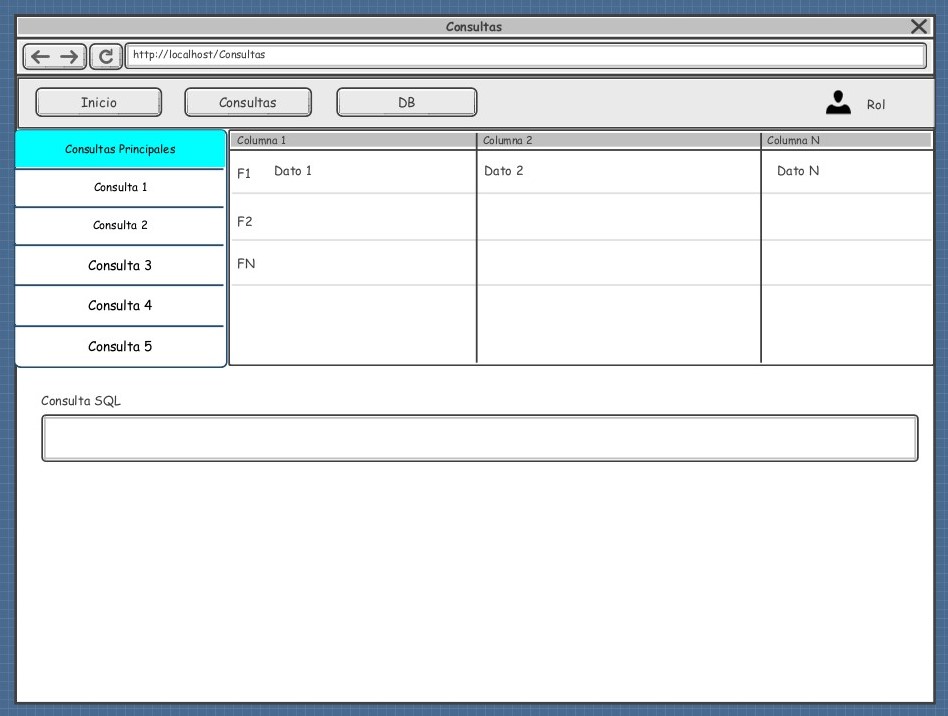
Vista ‘Inicio’



**Requisitos funcionales que se recogen en esta vista**

* Administrar Personas
* Administrar Contribuyentes
* Administrar Entidades Recaudadoras
* Gestionar Impuestos para cada contribuyente
* Verificar titulares de empresa

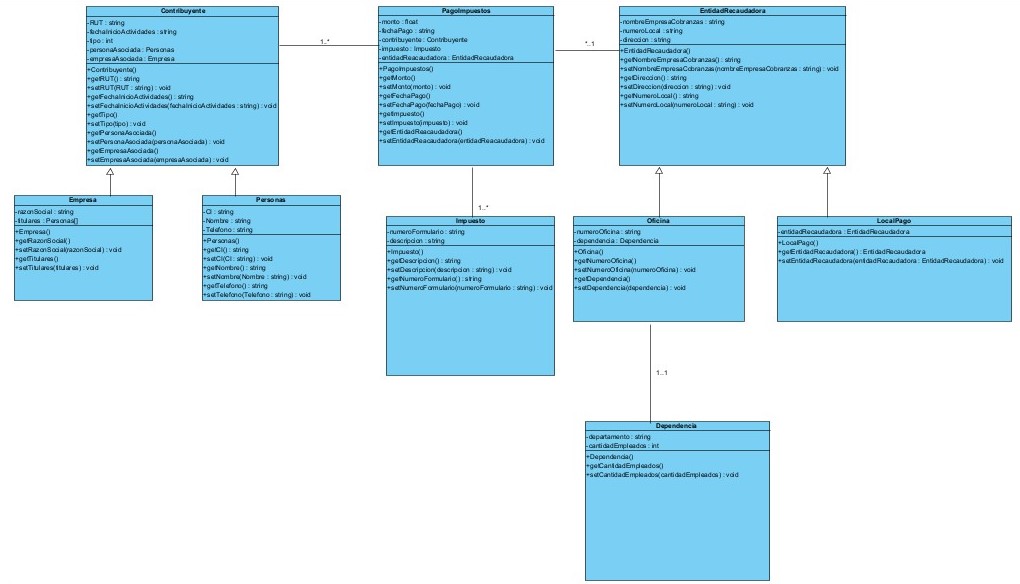
Vista ‘Consultas’



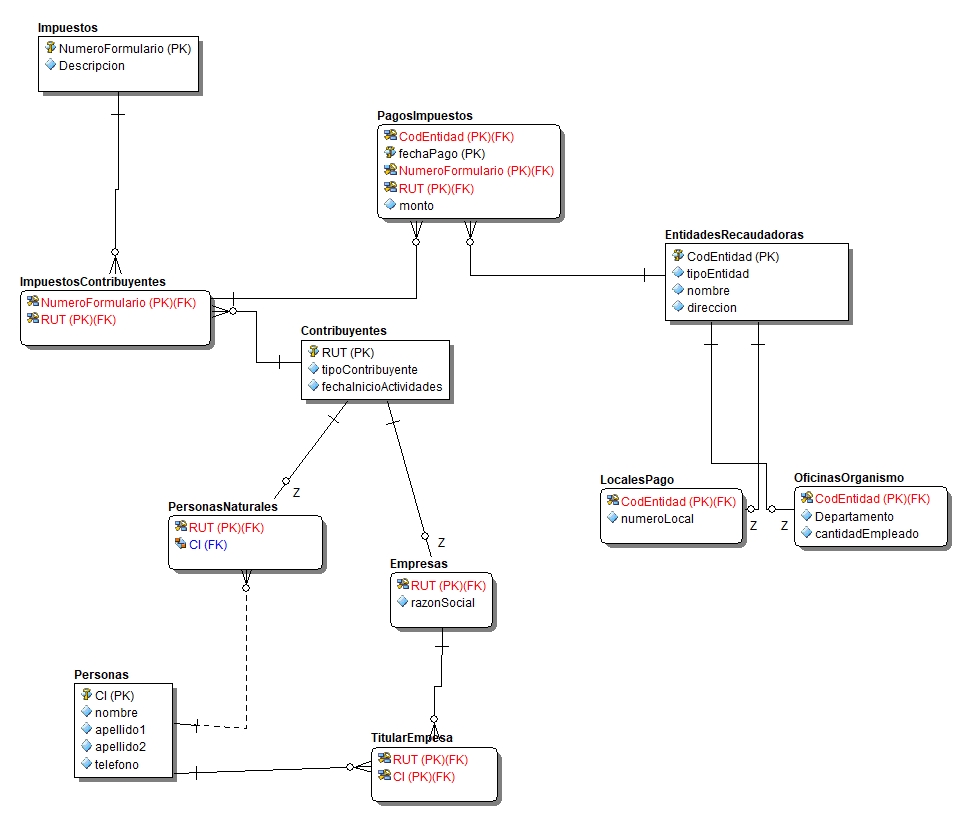
**Requisitos funcionales que se recogen en esta vista**

* Listar Personas Naturales que deben pagar todos los impuestos
* Listar RUT de las empresas y la cantidad de titulares que tienen
* Listar Contribuyentes (Personas Naturales) que sólo pagan en locales de pago
* Listar Monto total pagado por las empresas en un período dado
* Listar los 3 primeros impuestos que más han recaudado en un período

**Diagrama de clases**



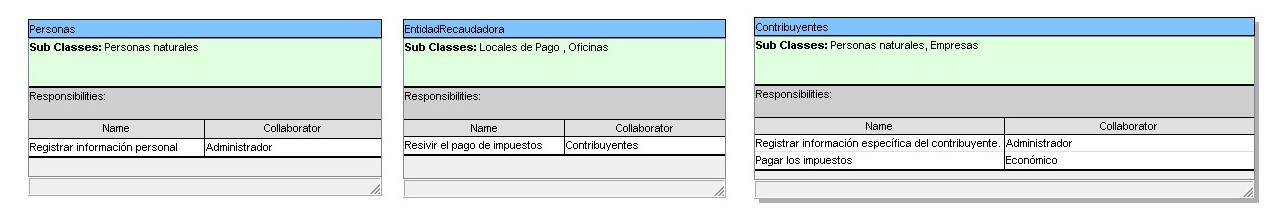
**Diagrama Entidad Relación**



**Tareas de ingeniería:**

1. Diseñar la base de datos para almacenar la información de personas, contribuyentes, empresas, impuestos, entidades recaudadoras, etc.
2. Desarrollar la funcionalidad de registro y gestión de personas, contribuyentes y empresas.
3. Implementar la lógica de asociación entre personas y contribuyentes, así como la gestión de titulares de empresas.
4. Desarrollar la funcionalidad de registro y gestión de impuestos y pagos.
5. Implementar la lógica de registro y gestión de entidades recaudadoras, locales de pago y oficinas del organismo.
6. Desarrollar la funcionalidad de registro y gestión de dependencias del organismo.
7. Crear interfaces de usuario para permitir a los administradores y económicos interactuar con el sistema.

**Tarjetas CRC**



**Historia de Usuario**

1. Como administrador, quiero poder registrar personas identificadas por su cédula de identidad, con nombre y teléfonos asociados.
2. Como administrador, quiero poder registrar contribuyentes identificados por su RUT, con fecha de inicio de actividades.
3. Como administrador, quiero distinguir entre personas naturales y jurídicas como contribuyentes.
4. Como administrador, quiero asociar personas naturales a contribuyentes correspondientes.
5. Como administrador, quiero registrar empresas con su razón social y titulares asociados.
6. Como administrador, quiero gestionar la asociación de personas como titulares de empresas.
7. Como administrador, quiero gestionar los impuestos que deben pagar los contribuyentes.
8. Como administrador, quiero registrar entidades recaudadoras como locales de pago externos u oficinas del organismo.
9. Como administrador, quiero registrar las dependencias del organismo por departamento con su cantidad de empleados.
10. Como administrador, quiero registrar oficinas identificadas por número y pertenecientes a una dependencia.
11. Como económico, quiero registrar los pagos de impuestos realizados por contribuyentes en distintas entidades recaudadoras.
12. Como económico, quiero registrar el monto y fecha de pago de cada transacción.

**Pruebas**

Durante la implementación del sistema se llevaron a cabo diversas pruebas para garantizar la calidad y funcionamiento del software.

**Pruebas unitarias**

Aprovechando el modelo basado en componentes de Angular se implementaron pruebas unitarias utilizando la herramienta Jasmine que permite escribir casos de prueba para verificar el comportamiento esperado. Por ejemplo, se verificó que al registrar un conjunto nuevas personas, se almacenaran correctamente en la base de datos y se mostrara en la lista de personas registradas.

**Pruebas Integración**

Posteriormente, se realizaron pruebas de integración para verificar la interacción entre los distintos componentes. Estas pruebas se centraron en asegurar que los módulos se comunicaran correctamente entre sí y que los datos se transmitieran de manera adecuada a lo largo de todo el flujo de trabajo Se simularon escenarios de uso real para garantizar que todas las partes del sistema funcionaran correctamente en conjunto. Por ejemplo, se probó el proceso completo de registro de un contribuyente, desde la creación de la entidad hasta el pago de impuestos en una entidad recaudadora.

**Pruebas de Validación**

Finalmente, se llevaron a cabo pruebas de validación para verificar que el sistema cumplía con los requisitos y expectativas del usuario final. Estas pruebas se realizaron utilizando casos de prueba específicos que reflejaban los principales flujos de trabajo del sistema.

**Pruebas de Aceptación**

Aun no se han realizado pruebas de aceptación con algún usuario real, pero queda como recomendación para obtener retroalimentación sobre la usabilidad y funcionalidad del sistema.

**Conclusiones del capítulo**

Este capítulo sirve como base sólida para las fases subsiguientes del proyecto, proporcionando una estructura coherente y funcional sobre la cual construir y mejorar el sistema en desarrollo. El prototipo desarrollado no solo representa un logro en sí mismo, sino también un punto de partida para futuras iteraciones, para alcanzar un producto final robusto y satisfactorio para sus usuarios.

**Conclusiones Generales**

Se puede concluir que el sistema gestor de recaudación de impuestos ha sido exhaustivamente abordado desde diversas perspectivas. El análisis inicial ha permitido identificar los procesos clave y los actores involucrados. El desarrollo del prototipo del sistema ha sido guiado por una metodología bien definida, que ha incluido la especificación detallada de requisitos funcionales y no funcionales. La implementación de historias de usuario y tarjetas CRC ha facilitado la asignación de tareas de ingeniería. En conjunto, este enfoque ha permitido diseñar un sistema que cumple con los requisitos del cliente de manera efectiva y que está preparado para ser desarrollado y desplegado en un entorno operativo real, contribuyendo así a mejorar la eficiencia y transparencia en la gestión de impuestos.

**Bibliografía**

1. Pressman, R. S., & Troya, J. M. (1988). *Ingeniería del software*. McGraw-Hill Educación
2. Pressman, R. S . (2010). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education
3. Sommerville, I. (2011). *Ingeniería del software*. Pearson educación.