**PROPUESTA PROYECTO INTEGRADOR**

**MAYRON SANTIAGO SERRANO HERNANDEZ**

**JAVIER ANTONIO BALLESTEROS HOYOS**

**ESTEBAN DAVID LOZANO RODRÍGUEZ**

**INGENIERÍA DEL SOFTWARE I**

**BASE DE DATOS II**

**PROGRAMACIÓN II**

**JONATHAN JAIR MONSALVE SALAZAR**

**ALEXANDRA SORAYA BELTRAN CASTRO**

**SILVIA JULIANA MORENO ROA**

**4L**

**2024**

Tabla de contenido

* 1.Introducción
  + 1.1 Descripción del problema
  + 1.2 Objetivos
  + 1.3 Justificación
* 2 Propuesta del plan de proyecto
  + 2.1 Análisis de requerimiento del software
  + 2.2 Requisitos y requerimientos
* 3.Diseño UML
  + 3.1 Diagrama de casos de uso
  + 3.2 Diagrama de clases
* 4 Modelamiento de la base de datos
  + 4.1 Diagrama entidad relación
  + 4.2 Diagrama relacional
  + 4.3 Diccionario de datos
* 5 Interfaz de usuario

**Software de facturación y reportes**

**1.Introducción**

Este documento tiene como objetivo desarrollar una plataforma de software para Obras Vanguard, una compañía constructora fundada en 1999, que ha sido reconocida por su alto valor en el sector durante los últimos años, enfocada en la gestión de pagos,venta de apartamentos y la administración eficiente de proyectos de vivienda. Sin embargo presenta algunas dificultades con su sistema actual. Como objetivo principal se desea crear una solución centralizada, la plataforma permitirá a los administradores gestionar proyectos, torre y apartamentos, mientras que a los asesores podrán registrar ventas y pagos, con un enfoque en generar reportes financieros detallados y optimizar la toma de decisiones.

El software incluirá una base de datos estructurada para relacionar la información de clientes, asesores, proyectos y pagos, permitiendo el seguimiento claro de estas operaciones. También se incluirá un módulo de reportes que genere análisis detallado, se tendrá un sistema de roles que restringe el acceso según los permisos de los usuarios encargados, asegurando que solo estos usuarios encargados accedan a la información clave.

**2.Descripción del problema**

**Imprecisión en la gestión de los pagos:**La gestión manual de los pagos ha llevado a errores frecuentes en los registros financieros, lo que genera inconsistencias dentro de la compañía y dificulta el control de ingresos y egresos. Estos errores pueden afectar la relación con los clientes, complicar el seguimiento de las deudas y abonos pendientes.

**Ineficiencia en la creación y administración de proyectos:**La falta de un software eficiente y centralizado, hace que estos procesos sean más complejos a la hora de administrarlos, desde la creación de nuevos proyectos, torres y apartamentos. Esto hace que se tengan datos no actualizados por lo que deberán actualizarlos de manera manual y perder mucho tiempo además aumentando la probabilidad de errores al momento de actualizar estos datos.

**Dificultades en el seguimiento de los pagos:**Como se tiene ausencia de un método organizado para registrar y supervisar los pagos esto lleva a una confusión sobre los estados de los clientes, generando conflictos por pagos que están duplicados o retrasados. Generando que sea un problema para la compañía que podría cortar el flujo de compradores.

**Impedimento para generar reportes:La** generación de reportes es un proceso que se hace lento y propenso a errores, ya que depende de los datos registrados manualmente. Esta situación hace que se creen limitaciones por parte de los usuarios que controlan esto y dificultando la toma de decisiones que favorezcan a la compañía.

**3.Objetivos**

**General:**Desarrollar una solución centralizada de software que optimice los procesos claves como la gestión financiera de la compañía constructora, mejorando el control de las actividades de venta, cobro, facturación y seguimiento de los proyectos.

**Específico:**Diseñar y construir una base de datos estructurada que almacene y relacione la información de administradores, proyectos, clientes y asesores, en esto se incluye los datos necesarios para controlar las operaciones de ventas, pagos y el estado de cada proyecto.

**Específico:**Implementar un módulo de generación de reportes que pueda ofrecer informes precisos y detallados de la finanzas, las ventas, el desempeño de los asesores. Este módulo debe facilitar el acceso a datos claves en tiempo real para ayudar a los usuarios a tomar decisiones importantes de manera informada.

**Específicos:**Implementar un sistema de seguridad basado en roles que límite y gestione los permisos de cada usuario, diferenciando las capacidades de administradores y asesores, el sistema debe permitir que los usuarios autorizados puedan acceder a la información, realizar modificaciones y generar reportes según sea necesario.

**4.Justificación**

La creación de este software especializado permitirá centralizar y automatizar los procesos críticos actuales de la compañía en la gestión de proyectos, que incluye una falta de control centralizado de información, errores en el seguimiento de ventas y pagos además de la dificultad para generar reportes detallados y confiables. Al centralizar estos problemas, el sistema permitirá a los usuarios encargados visualizar y controlar cada aspecto de los proyectos de manera organizada y eficiente, generando de que se minimicen los errores manuales y facilitando la toma de las decisiones.

**5.Propuesta del plan de proyecto**

**Proyecto:** Software de Gestión de Proyectos de Vivienda

**Objetivo:** Crear un software que permita gestionar proyectos de vivienda, cubriendo la administración de torres, apartamentos, ventas, pagos y reportes, con roles diferenciados y una interfaz intuitiva.

**Enfoque Metodológico:**

El proyecto se desarrollará utilizando el Modelo en cascada, avanzando de forma secuencial desde la fase de análisis hasta la implementación. Este enfoque asegura que cada etapa esté completa antes de pasar a la siguiente, lo que facilita la planificación y el control de los entregables.

**Resumen del Alcance**

1. Gestión de Proyectos y Apartamentos: Control total sobre la creación, modificación y eliminación de proyectos, torres y apartamentos.

2. Clientes y Ventas: Registro de clientes y ventas, con detalles de precios, cuotas, intereses, y cliente asociado.

3. Pagos y Dashboard: Control de pagos realizados, con un tablero que muestra cuotas próximas a vencer y vencidas.

4. Reportes y Seguridad: Generación de reportes filtrables y sistema de autenticación con roles para proteger la información.

**Fases del Proyecto (Modelo en Cascada)**

**1. Análisis de Requisitos:** Documentación de especificaciones funcionales y no funcionales.

**2. Diseño del Software:** Modelado de la base de datos y diseño de interfaz.

**3. Desarrollo del Software:** Creación de módulos de gestión, ventas, pagos y dashboard.

**4. Pruebas:** Validación funcional y de seguridad.

**5. Implementación:** Capacitación, documentación y entrega final.

**Criterios de Éxito:** El software debe cumplir todos los requisitos funcionales, con una interfaz accesible, manejo seguro de datos, y un correcto funcionamiento del sistema de roles y permisos.

**Análisis de Requerimientos del software:**

**Requerimientos**

Requerimiento 1:

El software tiene que permitir crear, modificar y borrar proyectos de vivienda. Cada proyecto tendría varias torres, y cada torre tendría varios apartamentos.

Requerimiento 2:

Necesitamos que se pueda ver y manejar la información de cada proyecto, torre y apartamento, incluyendo detalles como el número de apartamentos por torre, el precio de cada apartamento, el tipo de unidad (apartamento, local, garaje), el área y otros datos importantes.

Requerimiento 3:

Hay que tener un registro completo de los clientes, incluyendo su identificación, contacto, información sobre subsidios y otros detalles para gestionar la venta de apartamentos.

Requerimiento 4:

Los asesores deben poder registrar las ventas de apartamentos, anotando el precio total, el número de cuotas, intereses y el cliente relacionado.

Requerimiento 5:

También necesitamos un control de los pagos que realizan los clientes, con el monto, la fecha, el cliente que paga y el asesor que recibe el pago.

Requerimiento 6:

Queremos un tablero que muestre las cuotas próximas a vencer y las vencidas, con los nombres de los clientes y toda la info relacionada.

Requerimiento 7:

Hay que generar reportes de ventas y pagos, con opciones para filtrar por proyectos, torres, apartamentos, clientes y demás.

Requerimiento 8:

El software tiene que tener roles: los administradores que se encargan de todo lo del proyecto y los asesores que gestionan las ventas y pagos. Cada rol debería tener acceso a lo que le corresponde.

Requerimiento 9:

Queremos que la info esté segura. Tiene que haber autenticación y cumplir con las leyes de protección de datos para que la información de los clientes, ventas y pagos esté protegida.

Requerimiento 10:

La interfaz debería ser intuitiva y sencilla, para que los administradores y asesores hagan sus tareas sin tener que recibir mucho entrenamiento.

**Requisito:** El software debe permitir la **Gestión de Proyectos**, permitiendo al administrador crear, modificar y eliminar proyectos, especificando el nombre y el número de torres.  
**Identificador:** RF001  
**Nombre:** Gestión de Proyectos  
**Tipo:** Funcional  
**Prioridad:** Clave  
**Importancia:** 10/10  
**Quién aprueba:** Administrador del Software  
**Estado de revisión:** Aprobado  
**Nivel de riesgo:** Medio  
**Descripción:** Este requisito es esencial para gestionar los diferentes proyectos de vivienda de la empresa. Cada proyecto puede tener múltiples torres que contienen apartamentos. El administrador debe tener control total sobre la creación y actualización de estos proyectos para mantener la información actualizada.

**Requisito:**El software debe permitir la **Gestión de Torres**, asignando cada torre a un proyecto, con el número de torre y cantidad de apartamentos que contiene.  
**Identificador:** RF002  
**Nombre:** Gestión de Torres  
**Tipo:** Funcional  
**Prioridad:** Clave  
**Importancia:** 8/10  
**Quién aprueba:** Administrador del Software  
**Estado de revisión:** Aprobado  
**Nivel de riesgo:** Medio  
**Descripción:** Este requisito asegura que las torres dentro de cada proyecto estén correctamente organizadas y asociadas. El administrador necesita esta funcionalidad para gestionar la infraestructura de cada proyecto de vivienda.

**Requisito:**El software debe permitir la **Gestión de Apartamentos**, registrando el número, valor, tipo (apartamento, local, garaje), área, matrícula y fecha de escritura.  
**Identificador:** RF003  
**Nombre:** Gestión de Apartamentos  
**Tipo:** Funcional  
**Prioridad:** Clave  
**Importancia:** 9/10  
**Quién aprueba:** Administrador del Software  
**Estado de revisión:** Aprobado  
**Nivel de riesgo:** Medio  
**Descripción:** Este requisito es clave para manejar los detalles específicos de cada unidad habitacional dentro de las torres. Proporciona la información necesaria sobre los apartamentos que están en venta, lo cual es fundamental para las futuras transacciones.

**Requisito:**El software debe permitir la **Gestión de Ventas de Apartamentos**, lo que incluye el registro de la información de venta, como el precio total de venta, el número de cuotas, los intereses y la información del cliente asociado.  
**Identificador:** RF004  
**Nombre:** Gestión de Ventas de Apartamentos  
**Tipo:** Funcional  
**Prioridad:** Clave  
**Importancia:** 10/10  
**Quién aprueba:** Asesor del Software  
**Estado de revisión:** Aprobado  
**Nivel de riesgo:** Alto  
**Descripción:** Este requisito es fundamental para la gestión efectiva de las ventas de apartamentos. Permite a los asesores registrar todas las transacciones de venta, manteniendo un registro detallado y preciso de cada operación, lo cual es clave para el seguimiento financiero y para asegurar la correcta relación entre clientes y el proceso de compra.

**Requisito:**El software debe permitir el **Registro de Clientes**, incluyendo la captura de información como cédula, nombre, SISBEN, subsidio ministerial, dirección, teléfono y correo electrónico.  
**Identificador:** RF005  
**Nombre:** Registro de Clientes  
**Tipo:** Funcional  
**Prioridad:** Clave  
**Importancia:** 8/10  
**Quién aprueba:** Administrador del Software  
**Estado de revisión:** Aprobado  
**Nivel de riesgo:** Medio  
**Descripción:** Este requisito es crucial para gestionar la información de los clientes de manera eficiente. Permite a los asesores y administradores ingresar y almacenar datos detallados sobre los clientes, lo cual es necesario para la correcta gestión de ventas, pagos y seguimiento de clientes en los proyectos de vivienda.

**Requisito:**El software debe permitir la **Gestión de Pagos**, facilitando el registro de pagos realizados por los clientes, incluyendo el valor del pago, la fecha y el cliente asociado.  
**Identificador:** RF006  
**Nombre:** Gestión de Pagos  
**Tipo:** Funcional  
**Prioridad:** Clave  
**Importancia:** 10/10  
**Quién aprueba:** Administrador del Software  
**Estado de revisión:** Aprobado  
**Nivel de riesgo:** Alto  
**Descripción:** Este requisito es esencial para el manejo financiero del software. Permite a los asesores registrar cada pago realizado por los clientes, asegurando que los datos de pagos estén actualizados y correctamente asociados a las ventas correspondientes. La gestión precisa de los pagos es crucial para mantener el control financiero y para la generación de reportes sobre el estado de cuentas de los clientes.

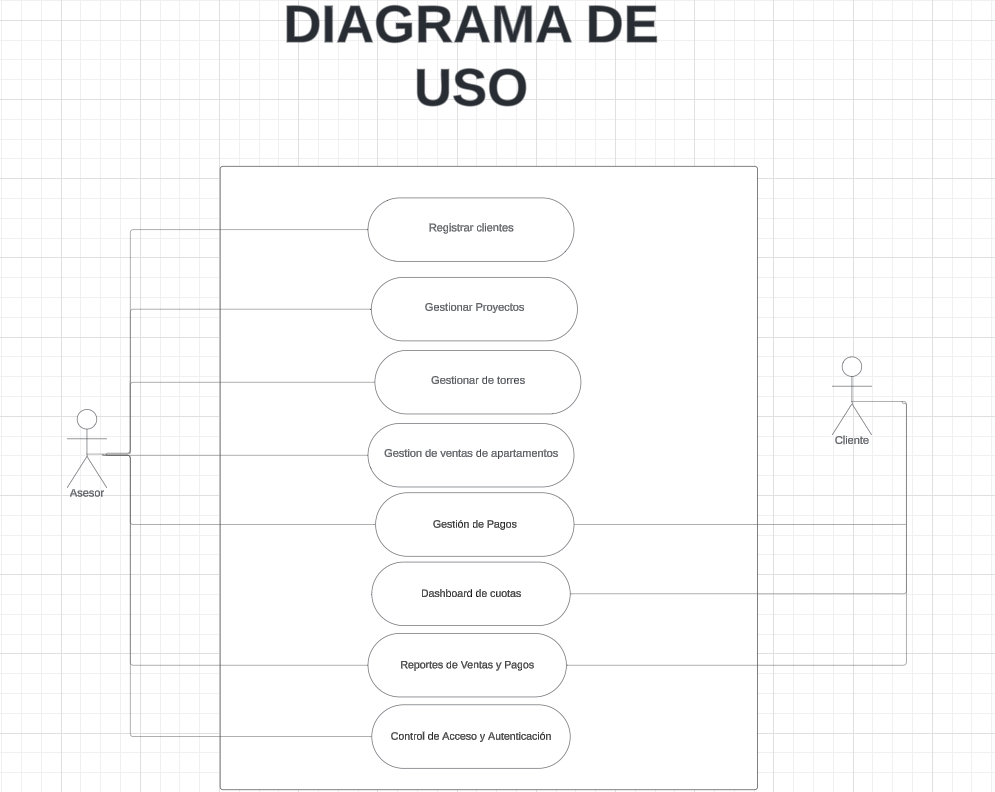
**Requisito:**El software debe proporcionar un **Dashboard de Cuotas** que muestre las cuotas próximas a vencer y las cuotas ya vencidas, con los nombres de los clientes correspondientes.  
**Identificador:** RF007  
**Nombre:** Dashboard de Cuotas  
**Tipo:** Funcional  
**Prioridad:** Media  
**Importancia:** 7/10  
**Quién aprueba:** Administrador del Software  
**Estado de revisión:** Aprobado  
**Nivel de riesgo:** Medio  
**Descripción:** Este requisito proporciona una vista consolidada de las cuotas pendientes y vencidas, permitiendo a los administradores y asesores monitorear fácilmente el estado de los pagos. El dashboard debe mostrar información clara y accesible para facilitar el seguimiento y la gestión de las cuentas de los clientes.

**Requisito:**El software debe generar **Reportes de Ventas y Pagos**, permitiendo la creación de informes detallados sobre las ventas realizadas y los pagos recibidos, con opciones de filtrado por cliente, fecha y proyecto.  
**Identificador:** RF008  
**Nombre:** Reportes de Ventas y Pagos  
**Tipo:** Funcional  
**Prioridad:** Media  
**Importancia:** 7/10  
**Quién aprueba:** Administrador del Software  
**Estado de revisión:** Aprobado  
**Nivel de riesgo:** Bajo  
**Descripción:** Este requisito es importante para la evaluación y análisis de las transacciones financieras. Los reportes deben proporcionar una visión clara y detallada de las ventas y pagos, facilitando la toma de decisiones y el seguimiento del rendimiento financiero del software. Los filtros permiten ajustar la información según diferentes criterios para obtener informes específicos y útiles.

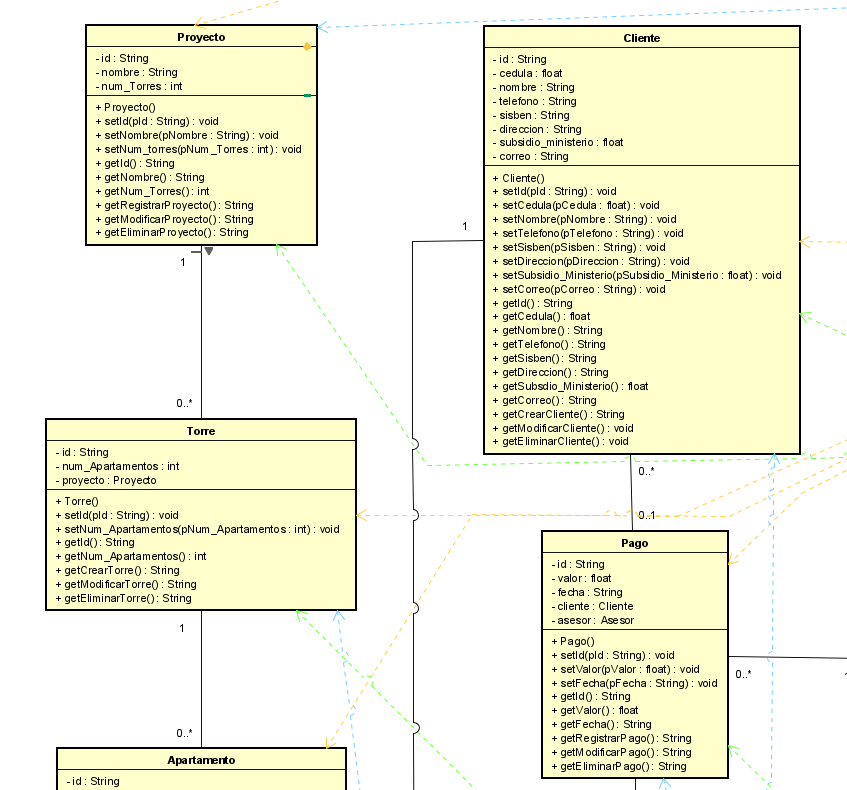
**Requisito:**El software debe implementar **Control de Acceso y Autenticación**, asegurando que solo los usuarios autorizados puedan acceder a las funcionalidades según sus roles definidos.  
**Identificador:** RNF009  
**Nombre:** Control de Acceso y Autenticación  
**Tipo:** No Funcional  
**Prioridad:** Clave  
**Importancia:** 9/10  
**Quién aprueba:** Administrador del Software  
**Estado de revisión:** Aprobado  
**Nivel de riesgo:** Alto  
**Descripción:** Este requisito es crucial para la seguridad del software. Debe garantizar que el acceso a diferentes secciones del software, como la gestión de proyectos, ventas y pagos, esté restringido según el rol del usuario (administrador o asesor). La autenticación y los permisos de acceso deben estar bien implementados para proteger la información y evitar accesos no autorizados.

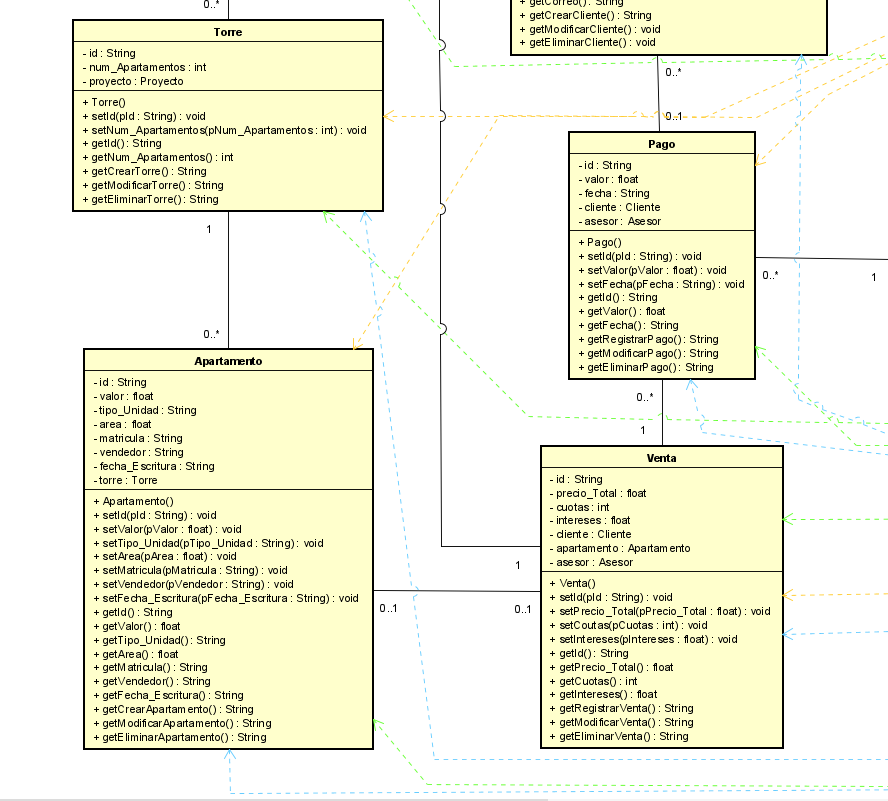
**DIAGRAMAS**

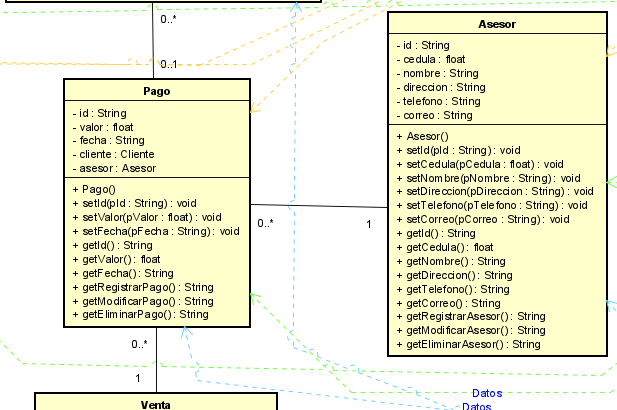
**Diagrama de casos de usos:**

****

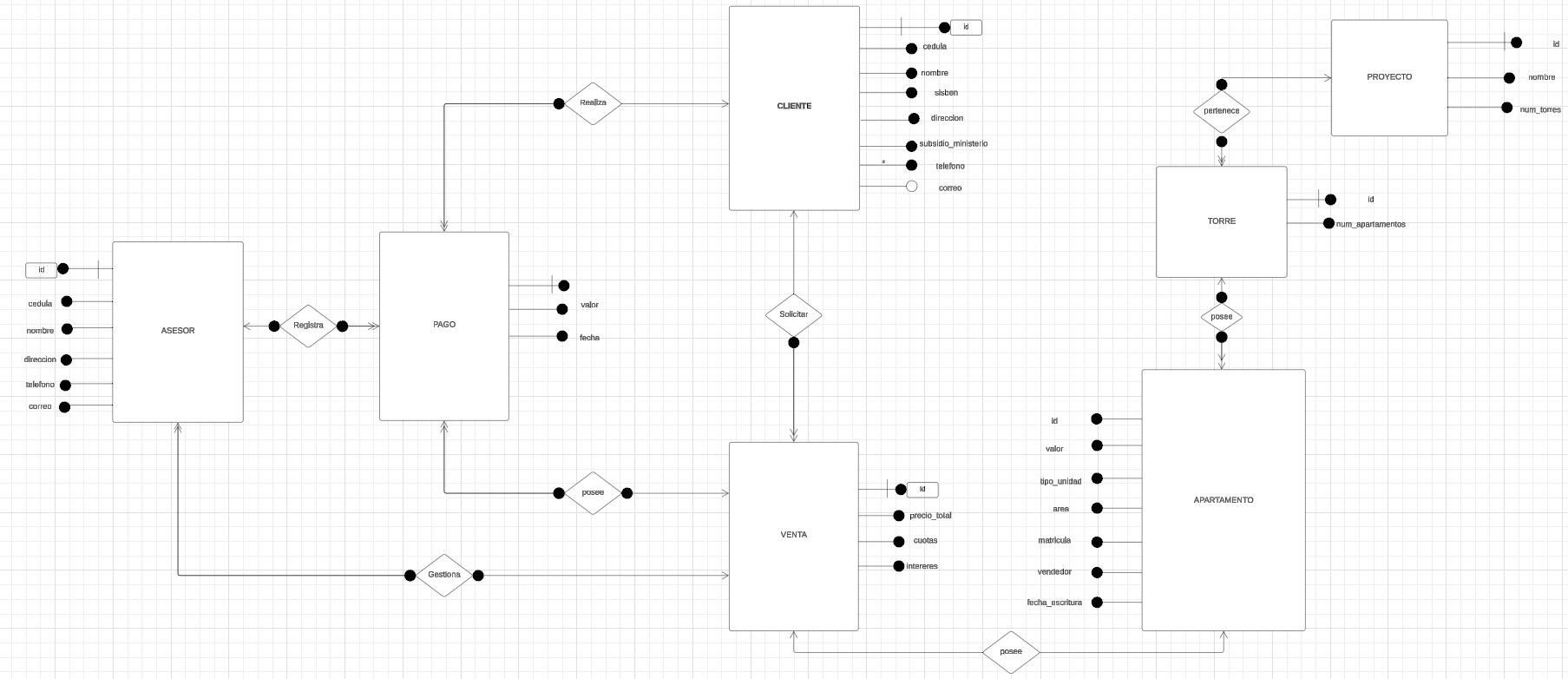
**Diagrama de clases(Hecho en astah):**

****

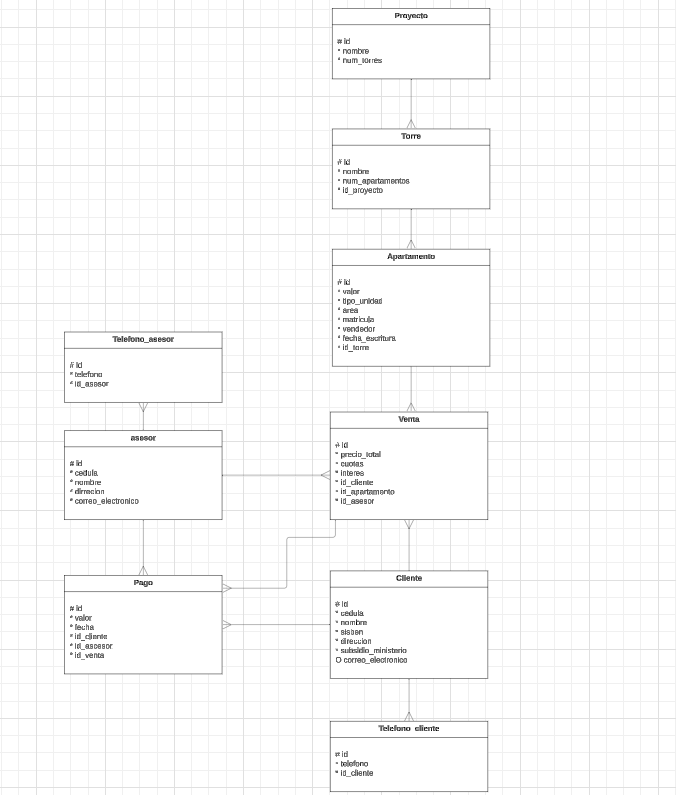
****

****

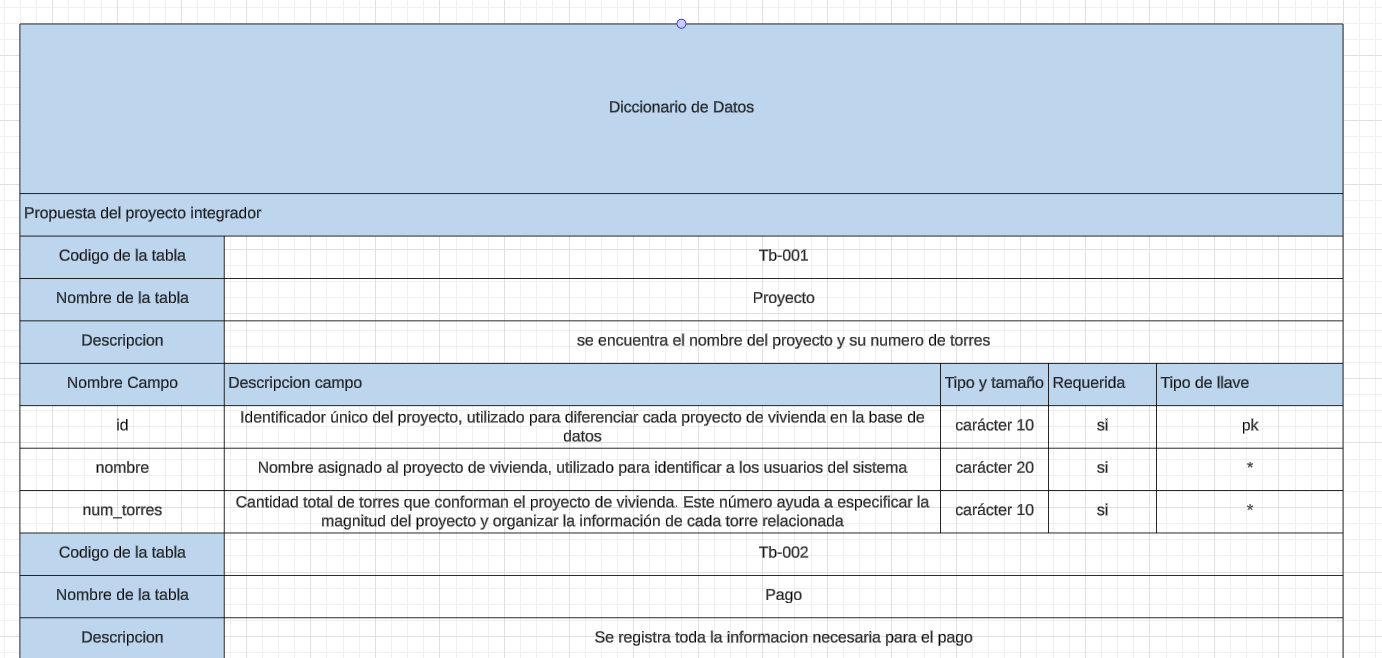
**Diagrama entidad relación:**

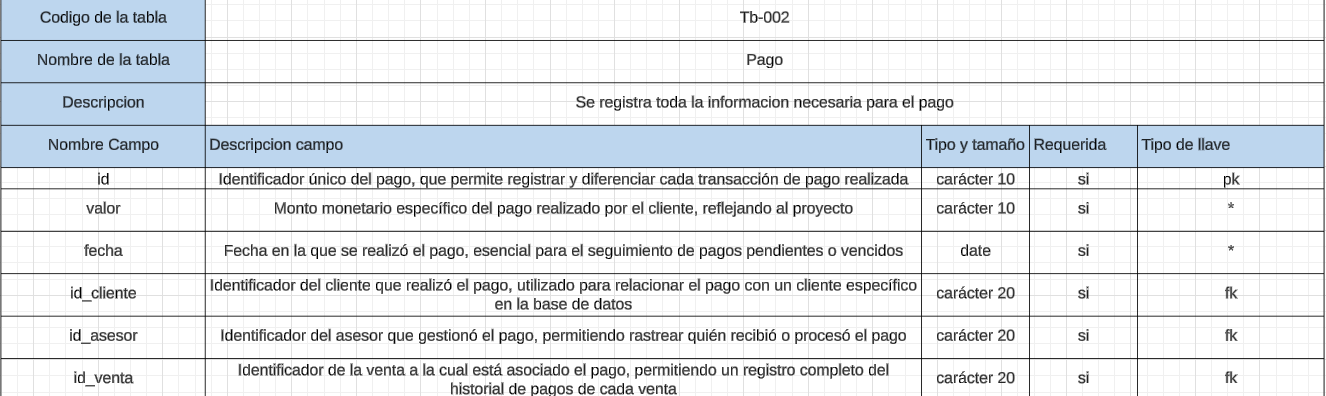


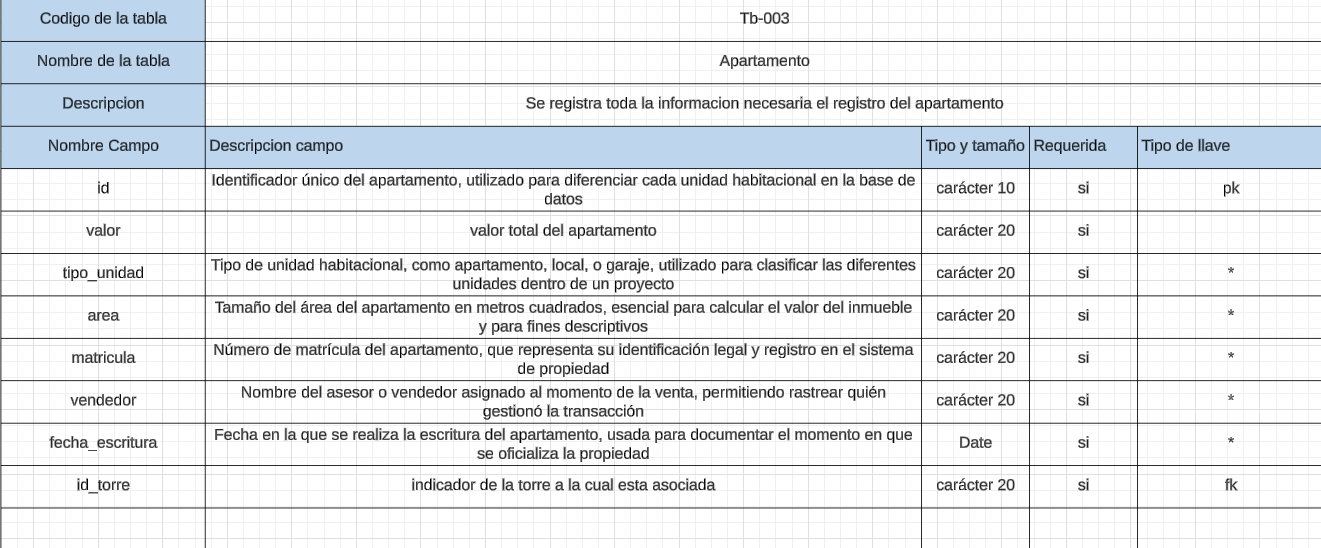
**Diagrama relacional:**

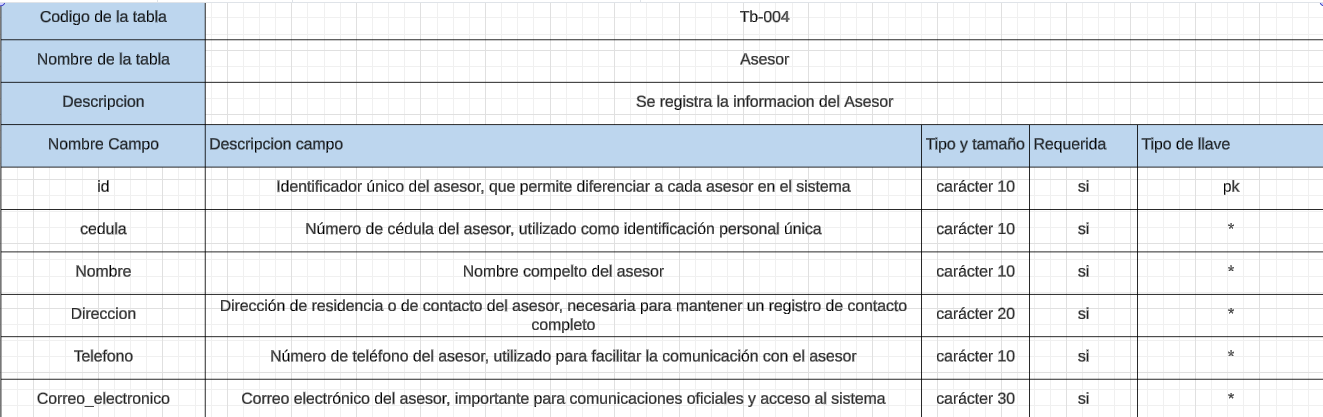


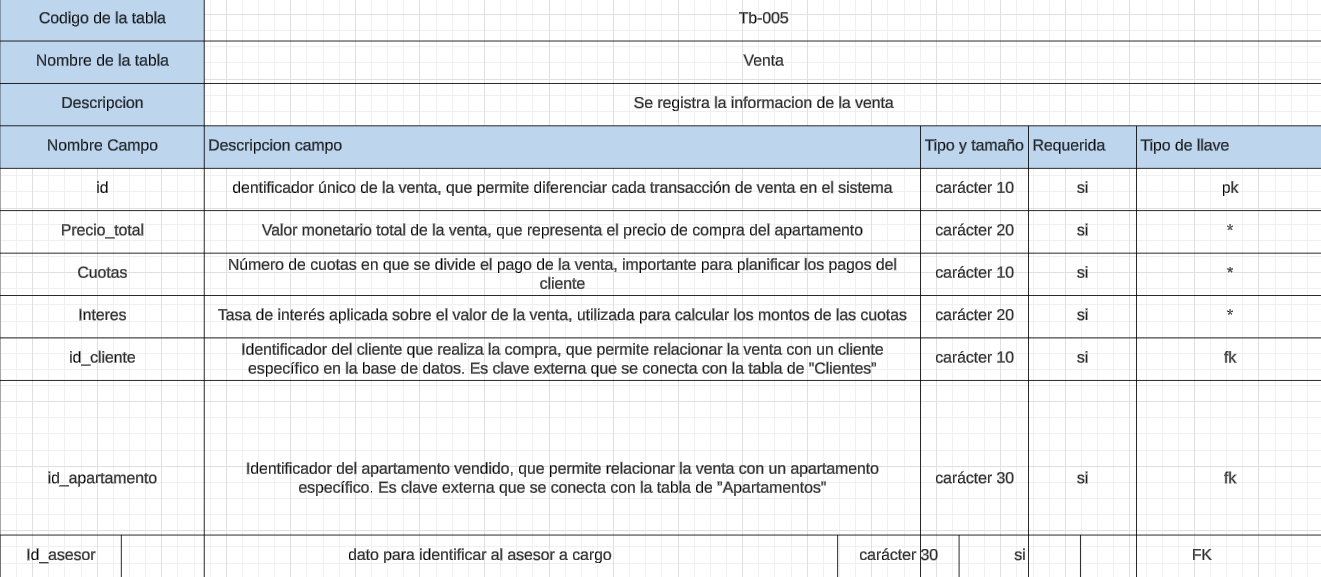
**Diccionario de datos:**

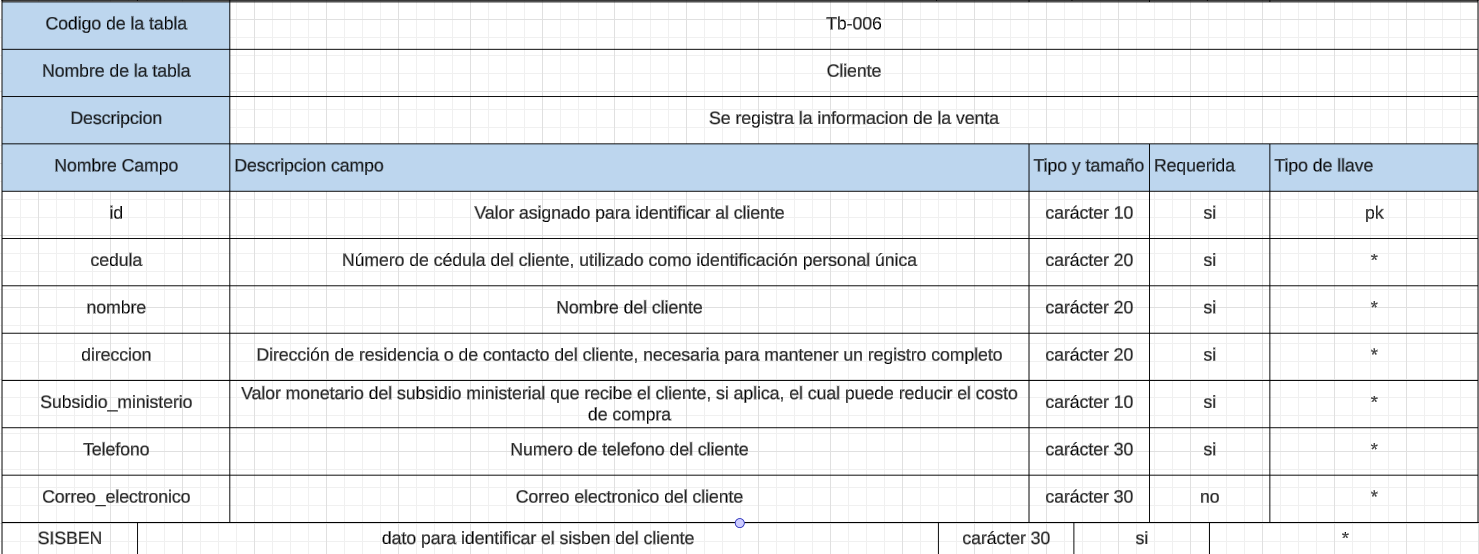


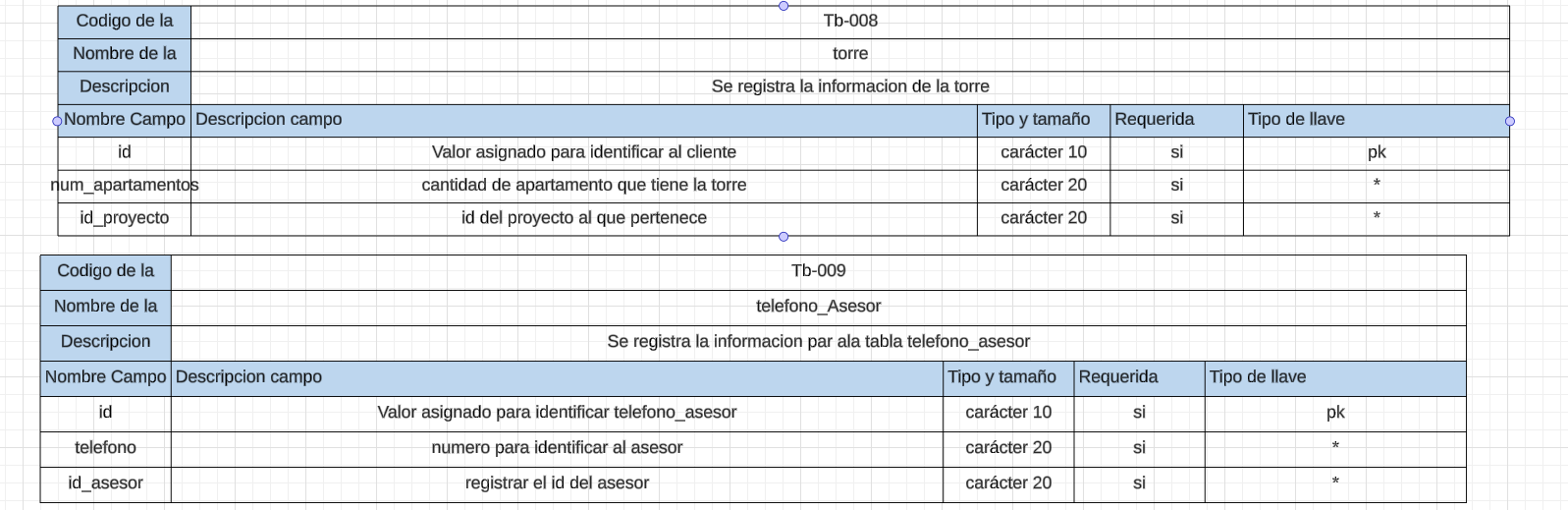


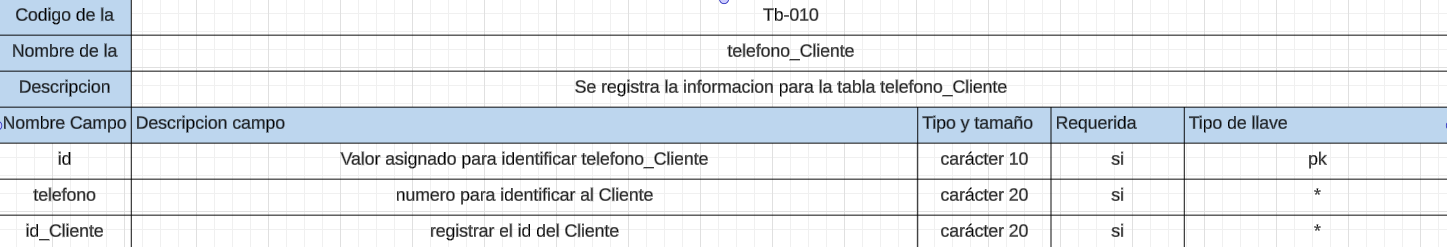












**INTERFAZ:**







