

SEMINARIO DE PRÁCTICA DE INFORMATICA

“PROYECTO

SISTEMA GESTIÓN PARA GENERACION DISTRIBUIDA DE MISIONES ENERGIA”



PROFESOR TITULAR DISCIPLINAR:

PABLO ALEJANDRO VIRGOLINI

TITULAR EXPERTO:

PABLO ALEJANDRO VIRGOLINI

ANA CAROLINA FERREYRA

ALUMNO:

ALBERTO BASILIO AGUIRRE

AÑO 2024

INTRODUCCION:

En el mundo actual, donde la tecnología y la sostenibilidad están en el centro de las discusiones empresariales, el desarrollo de sistemas de gestión eficientes para empresas de energía se ha vuelto crucial. Esta solución está especialmente diseñada para adaptarse a las necesidades cambiantes del sector energético, permitiendo a la empresa no solo mantener un control eficiente sobre sus operaciones sino también responder de manera proactiva a las demandas del mercado y regulaciones ambientales.

JUSTIFICACION:

En la ciudad de Posadas, Misiones, se encuentra la startup Misiones Energía, la cual se dedica a las energías renovables, tanto fotovoltaicas, como térmicas, comprometida con los clientes, el medio ambiente y el cambio climático, la misma se dedica a la instalación de sistemas fotovoltaicos conectados a la red y fuera de la red.

Con este proyecto se pretende, diseñar, desarrollar e implementar un sistema de gestión, clientes, venta, facturación, control de stock, gestión de proveedores, gestión de instalaciones, monitoreo, análisis de datos, soporte al cliente entre otros factores relevantes del negocio.

Con este proyecto se busca reorganizar la generación distribuida y contribuir con la organización de la empresa, ayudando a lograr mayor sostenibilidad ambiental y la disminución de las emisiones de carbono.

Además de lograr un ahorro de costos de energía, al generar la propia energía, se reduce significativamente los costos de energía a largo plazo, es especialmente beneficioso dado el aumento constante de las tarifas eléctricas convencionales. Rentabilidad a largo plazo, la inversión inicial, de un sistema fotovoltaico bien diseñado puede generar retorno de la inversión atractivo a lo largo de su vida útil, en varias décadas. En algunos casos hay incentivos fiscales y financieros disponibles.

Independencia energética, al producir la propia energía, se reduce la dependencia a la red eléctrica convencional. Brinda mayor autonomía energética y protege contra aumentos repentinos en los precios de la electricidad o incluso de cortes de energía.

Se profundiza los antecedentes que dan origen al proyecto.

DEFINICIONES DEL PROYECTO Y DEL SISTEMA:

Objetivo General del Proyecto:

Recolectar, organizar, gestionar, analizar los datos de clientes, ventas, servicios de postventa, soporte técnico, y marketing para/con los sistemas de generación distribuida comercializados por la empresa. Todo esto para optimizar y maximizar el beneficio de los accionistas y organización de la empresa.

ELICITACION:

Entrevistas: Las entrevistas se utilizan para profundizar en la información. Obteniendo una comprensión más detallada de las necesidades de los usuarios específicos o partes interesadas.

Encuestas: Para obtener datos cuantitativos y una visión general de las necesidades.

Utilizamos un software de recolección de información, el mismo se nutre de los sistemas instalados en los domicilios de los clientes, que brindan información diariamente de generación, consumo, costes, temperatura, entre otros valores.

En gestión de clientes, un ERP y CRM, el cual brinda información de ventas, contratos, costos, proveedores, finanzas, entre otros datos.

CONOCIMIENTO DEL NEGOCIO:

El sistema que se desarrollara debe permitir la gestión y monitoreo de cada activo fotovoltaico, denominado planta de generación distribuida, que se instalara en el domicilio del cliente.

La energía solar fotovoltaica es aquella que se obtiene al convertir la luz solar en electricidad empleando una tecnología basada en el efecto fotoeléctrico. Los paneles solares compuestos por sus células fotovoltaicas producen electricidad.

La energía solar fotovoltaica es una fuente de energía renovable y limpia que utiliza la radiación solar para producir electricidad.

Se basa en el llamado efecto fotoeléctrico, por el cual determinados materiales son capaces de absorber fotones (partículas lumínicas) y liberar electrones, generando una corriente eléctrica.

Para ello se emplea un dispositivo semiconductor denominado celda o célula fotovoltaica, que puede ser de silicio monocristalino, policristalino o amorfo, o bien otros materiales semiconductores de capa fina.

Las de silicio monocristalino se obtienen a partir de un único cristal de silicio puro y alcanzan la máxima eficiencia, entre un 18% y un 20% de media.

Existen dos tipos de plantas fotovoltaicas, las que están conectadas a la red y las que no.

Central fotovoltaica: toda la energía producida por los paneles solares se vierte a la red eléctrica.

Generador con autoconsumo: Parte de la electricidad generada es consumida por el propio productor (en una vivienda) y el resto se vierte a la red (inyecta). Al mismo tiempo, el productor toma de la red la energía necesaria para cubrir su demanda cuando la unidad no le suministra la suficiente.

Estas instalaciones con conexión a la red cuentan con tres elementos básicos:

Paneles fotovoltaicos: Se trata de grupos de celdas fotovoltaicas, montadas entre capas de silicio que captan la radiación solar y transforman la luz (fotones) en energía eléctrica (electrones)

Inversores: convierten la corriente eléctrica continua (CC) que producen los paneles en corriente alterna (CA) apta para los equipos electrónicos y su consumo.

PROPUESTA DE SOLUCION:

Se presenta una propuesta de solución funcional, la propuesta técnica, la arquitectura planteada para el despliegue, y la lista de requerimientos a incluir en el sistema.

Propuesta funcional: Se propone desarrollar un sistema de gestión completo, para los sistemas de generación distribuidos comercializados a clientes. De esta forma se logrará gestionar de manera mas optima el funcionamiento de cada uno.

Propuesta técnica: Se va a realizar en Java, que cuenta con una gran variedad de bibliotecas y frameworks (marcos de trabajo) que permiten el escalamiento. La persistencia se realizará en MySQL, que es una base de datos relacional abierta, flexible y de alto rendimiento para el almacenamiento de datos en tiempo real.

Comunicaciones: Se empleara para el sistema gestión, red cableada Ethernet, y para las conexiones con los sistemas de generación distribuida, en caso de estar en un la ciudad, conexión wifi (proporcionado por el cliente) o conexión celular 4G, facilitando la transmisión de datos de los sensores, dispositivos IoT, conectándose al Servidor de Misiones Energía.

Requerimientos funcionales:

RFS01	El sistema debe permitir el ingreso mediante un usuario, contraseña, y código
RFS02	El sistema debe permitir registrar la información del equipo comercializado y demás productos. (gestión de proveedores)
RFS03	El sistema debe permitir registrar clientes, gestionar su información de contacto, realizar la venta, historial de venta y contratos (ejecutivo de cuentas)
RFS04	El sistema debe permitir almacenar contratos realizados por la administración en los perfiles del cliente para su respectiva firma (ejecutivo de cuentas)
RFS05	El sistema debe registrar la venta por parte de los ejecutivos de cuentas. (ejecutivo de cuentas)
RFS06	El sistema debe permitir registrar los pagos de los clientes, como así también emitir un recibo por el mismo o la factura correspondiente. (gestión financiera)
RFS07	El sistema debe permitir la generación de recibos por cada pago recibido del cliente. (gestión financiera)
RFS08	El sistema debe generar la factura cuando el pago sea recibido en su totalidad de parte del cliente (gestión financiera)
RFS09	El sistema debe emitir una orden de trabajo luego de la venta para el instalador profesional. (ejecutivo de cuentas)
RFS10	El sistema debe permitir registrar proyectos de energía solar, asignar recursos, planificar tareas, y realizar seguimiento del progreso. (ingeniería)
RFS11	El sistema debe registrar el stock debitado luego de la venta. (inventarios)
RFS12	El sistema debe permitir registrar los equipos adquiridos a proveedores, con sus respectivos costos, características entre otros datos. (inventarios)
RFS13	El sistema debe permitir registrar la instalación realizada y datos del personal que realice. (instalador profesional)
RFS14	El sistema debe permitir a los operadores detener el sistema de generación distribuida, reiniciar, o configurarlo en remoto. Registrar eventos del mismo en un log.(operador)
RFS15	El sistema debe registrar las condiciones de funcionamiento, los datos de mantenimientos programados realizados o a realizar, como así también el personal que realiza. (soporte técnico)
RFS16	El sistema debe registrar la información generada, del equipo, sensores, estadísticas. (centro de operaciones)
RFS17	El sistema debe mostrar la energía total generada por cada equipo instalado, sensores, cliente, o el total de las mismas, por región y país.
RFS18	El sistema enviar alertas, y notificaciones en caso de fallas en la operación.
RFS19	El sistema debe generar estadísticas del funcionamiento de los equipos y su desempeño . (centro de operaciones)
RFS20	El sistema debe registrar el llamado de servicio de postventa al cliente, como así también su nivel de satisfacción, reclamo, queja o sugerencia. (postventa)
RFS21	El sistema debe permitir el envío de informes de manera automática.

INICIO DE ANALISIS: CASOS DE USO.

Actores:

Administradores: Se encargan de altas, bajas, modificaciones y del funcionamiento del sistema y sus diferentes módulos además mantener el sistema, asignar roles, entre otros.

Ejecutivo de Ventas: Persona encargada de la comercialización de los productos, como así también el asesoramiento.

Postventa: Persona encargada de realizar el seguimiento del servicio de postventa, como la satisfacción del cliente, calidad de la instalación, dudas, sugerencias, reclamos entre otras.

Gestión financiera: Persona encargada de realizar cobros a clientes, controlar, emitir recibos, facturas y gestionar pagos a proveedores.

Inventarios: Persona encargada del inventario de equipos solares, gestionar proveedores, realizar pedidos de suministros, y asociar materiales con proyectos específicos.

Instalador Profesional: Persona encargada de realizar la instalación de los sistemas de generación distribuida en el domicilio del cliente.

Soporte técnico: Persona encargada de brindar soporte técnico al cliente, como así también registrar el funcionamiento del equipo.

Operador: Persona que está autorizada a realizar modificaciones en cada central de generación distribuida, utilizando el sistema.

Centro de Monitoreo: Monitorean, controlan, cada central de generación, como así también las fallas que pudieran existir.

Administrador de datos: sistema que se encarga de gestionar la persistencia en la base de datos.

Estadísticas: Persona encargada de evaluar el desempeño de los ejecutivos de venta, postventa, soporte técnico, operador, gestión financiera, gestión de inventarios.

Ingeniería: Persona encargada de la gestión de proyectos, antes de la instalación, para ser comercializado el equipo de generación distribuida.

CASOS DE USO:

CASOS DE USO:

Gestión de clientes y contratos:

CU001: Registrar un nuevo cliente

CU002: Almacenar un contrato realizado por la administración y asignarlo al perfil del cliente.

CU003: Registrar la información del equipo comercializado y demás productos.

CU004: Actualizar la información del cliente.

CU005: Eliminar un registro de clientes.

CU006: Buscar clientes por nombre, DNI, domicilio, código de proyecto.

CU007: Actualizar los detalles del contrato existentes.

CU008: Eliminar un contrato asociado a un cliente.

Gestión de Ventas y Facturación:

CU001: Registrar la venta.

CU002: Actualizar una venta registrada

CU003: Registrar los pagos de los clientes.

CU004: Generar recibo por cada pago recibido de acuerdo al cliente que paga.

CU005: Generar factura al cliente cuando el pago recibido sea el total.

CU006: Cancelar una venta

CU007: Buscar ventas por fecha, cliente o producto.

CU008: Generar reportes de ventas, mensuales, trimestrales y anuales.

Gestión de Instalaciones y Proyectos:

CU001: Generar una orden de trabajo para el instalador profesional

CU002: Registrar el proyecto de energía solar, asignar recursos, planificar tareas y realizar seguimiento del progreso.

CU003: Registrar el stock debitado luego de la venta y asignación de recursos.

CU004: Registrar la instalacion realizada y datos del personal que realizo.

CU005: Actualizar la información de una instalacion

CU006: Eliminar una instalacion registrada

CU007: Asignar tareas a instaladores y seguir su proyecto, de mantenimiento.

CU008: Actualizar el estado de un proyecto de energía solar.

CU009: Generar reportes de progreso de los proyectos.

Gestión de Equipos y Sensores:

CU001: El sistema de generación distribuida puede por parte del operador: Iniciar, detener, reiniciar o configurarlo en remoto.

CU002: Actualizar información de equipos y sensores.

CU003: Eliminar un registro de un equipo o sensor.

CU004: Realizar mantenimiento preventivo y correctivo de equipos

CU005: Generar alertas de fallas en equipos y sensores.

CU006: Registrar información generada del equipo, sensores, estadísticas.

Gestión de inventarios y Productos:

CU001: Registrar equipos adquiridos a proveedores, costos, características, marcas, modelos etc.

CU002: Mostrar la energía total generada por cada equipo instalado

CU003: Mostrar la energía total generada por cada equipo instalado

CU004: Generar estadísticas del funcionamiento de los equipos y su desempeño.

CU005: Registrar el llamado de servicio de postventa al cliente.

CU006: Permitir envío de informes de manera automática

CU007: Generar orden de compras al proveedor.

CU008: Registrar unidades vendidas a clientes registrados

CU009: Actualizar la información de productos y equipos

CU010: Eliminar un registro de productos

CU011: Realizar inventario de productos y equipos

CU012: Generar alertas de bajo stock.

Gestión de Servicios y Soporte:

CU001: Asignar técnicos para llamados de servicio

CU002: Seguir el progreso y resolución de llamados de servicio.

CU003: Actualizar un registro de postventa

CU004: Eliminar un registro de postventa

Gestión Proveedores:

CU005: Registrar información de proveedores

CU006: Actualizar información de proveedores

CU003: Eliminar un registro de proveedor

CU004: Buscar proveedores por nombre, cuit, productos suministrados, o ubicación.

CU005: Evaluar el desempeño de proveedores.

CU006: Generar estadísticas de compras a proveedores, día, mes, año, trimestre.

CU007: Generar reportes de compras a proveedores.

Gestión Financiera:

CU001: Generar reportes financieros mensuales, trimestrales y anuales.

CU002: Gestionar cuentas por pagar, y por cobrar.

CU003: Integrar con sistemas de pago electrónicos.

CU004: Actualizar la información de pagos realizados por los clientes.

CU005: Eliminar un registro de pago. Requiere doble factor de autenticación.

Gestión de Recursos Humanos:

CU001: Registrar y actualizar la información de empleados.

CU002: Asignar roles y permisos a empleados

CU003: Gestionar el horario y asistencia de empleados

CU004: Generar reportes de desempeños de empleados.

Gestión de Seguridad y Configuración:

CU001: Gestionar roles y permisos del sistema

CU002: Configurar parámetros de seguridad y autenticación

CU003: Realizar auditorías de seguridad y accesos al sistema

CU004: Generar copias de seguridad de la base de datos.

Gestión de Integración y Automatización:

CU001: Integrar con sistema de gestión ERP

CU067: Automatizar el envío de notificaciones por correo electrónico.

CU068: Implementar APIs para integraciones externas.

CU069: Configurar reglas de negocio para automatización de procesos.

CU070: Generar alertas y notificaciones en tiempo real.

ITEM	REQUERIMIENTO	CASO DE USO	ACTOR PRINCIPAL	PAQUETE DEL ANALISIS	COMENTARIO
RFS01	Registrar un nuevo cliente	CU001	Ejecutivo de ventas	Gestión de Clientes	Permitir registrar nuevos clientes en el sistema
RFS02	Almacenar un contrato y asignarlo al cliente	CU002	Administradores	Gestión de Contratos	Asociar contratos a perfiles de clientes
RFS03	Registrar información de productos	CU003	Inventarios	Gestión de Productos	Registrar productos y equipos comercializados
RFS04	Registrar una venta	CU004	Ejecutivos de Ventas	Gestión de Ventas	Registrar ventas realizadas a clientes
RFS05	Generar orden de trabajo para instalación	CU005	Administradores	Gestión de Instalaciones	Generar órdenes de trabajo para instaladores
RFS06	Registrar pagos de clientes	CU006	Gestión Financiera	Gestión de Pagos	Registrar pagos realizados por los clientes
RFS07	Generar recibo por cada pago recibido	CU007	Gestión Financiera	Gestión de Pagos	Generar recibos automáticamente
RFS08	Generar factura al cliente	CU008	Gestión Financiera	Gestión de Facturación	Generar facturas cuando se reciba el pago total
RFS09	Registrar proyecto de energía solar	CU009	Ingeniería	Gestión de Proyectos	Planificar y asignar recursos para proyectos
RFS10	Registrar stock debitado tras la venta	CU010	Inventarios	Gestión de Inventarios	Registrar disminución de stock tras ventas
RFS11	Registrar instalación realizada	CU011	Instalador Profesional	Gestión de Instalaciones	Registrar detalles de las instalaciones realizadas
RFS12	Controlar sistema de generación distribuida	CU012	Operador	Gestión de Operaciones	Permitir control remoto de sistemas de generación

RFS13	Registrar información de equipos y sensores	CU013	Soporte Técnico	Gestión de Equipos	Registrar datos de funcionamiento de equipos y sensores
RFS14	Registrar equipos adquiridos a proveedores	CU014	Inventarios	Gestión de Inventarios	Registrar detalles de equipos comprados a proveedores
RFS15	Mostrar energía total generada	CU015	Centro de Monitoreo	Gestión de Energía	Mostrar energía generada por equipos instalados
RFS16	Generar estadísticas del funcionamiento	CU016	Estadísticas	Gestión de Estadísticas	Generar informes de desempeño de equipos
RFS17	Registrar llamado de servicio postventa	CU017	Postventa	Gestión de Servicios	Registrar seguimiento de postventa
RFS18	Enviar informes automáticamente	CU018	Administradores	Gestión de Informes	Automatizar envío de informes
RFS19	Generar orden de compra al proveedor	CU019	Inventarios	Gestión de Compras	Generar órdenes de compra para proveedores
RFS20	Registrar unidades vendidas	CU020	Ejecutivos de Ventas	Gestión de Ventas	Registrar unidades vendidas a clientes
RFS21	Actualizar información del cliente	CU021	Ejecutivos de Ventas	Gestión de Clientes	Permitir actualizar datos de clientes
RFS22	Eliminar registro de cliente	CU022	Ejecutivos de Ventas	Gestión de Clientes	Permitir eliminar datos de clientes
RFS23	Buscar cliente por varios criterios	CU023	Ejecutivos de Ventas	Gestión de Clientes	Facilitar la búsqueda de clientes
RFS24	Actualizar detalles de contrato	CU024	Administradores	Gestión de Contratos	Permitir actualizar contratos existentes
RFS25	Eliminar contrato	CU025	Administradores	Gestión de Contratos	Permitir eliminar contratos

RFS26	Actualizar información de productos	CU026	Inventarios	Gestión de Productos	Permitir actualizar detalles de productos
RFS27	Eliminar producto	CU027	Inventarios	Gestión de Productos	Permitir eliminar productos
RFS28	Realizar inventario de productos	CU028	Inventarios	Gestión de Inventarios	Facilitar inventarios periódicos de productos
RFS29	Generar alertas de bajo stock	CU029	Inventarios	Gestión de Inventarios	Alertar sobre productos con bajo stock
RFS3	Actualizar una venta registrada	CU030	Ejecutivos de Ventas	Gestión de Ventas	Permitir actualizar detalles de ventas
RFS31	Cancelar una venta	CU031	Ejecutivos de Ventas	Gestión de Ventas	Permitir la cancelación de ventas
RFS32	Buscar ventas por varios criterios	CU032	Ejecutivos de Ventas	Gestión de Ventas	Facilitar la búsqueda de ventas
RFS33	Generar reportes de ventas	CU033	Estadísticas	Gestión de Estadísticas	Generar informes de ventas periódicos
RFS34	Actualizar detalles de instalación	CU034	Instalador Profesional	Gestión de Instalaciones	Permitir actualizar información de instalaciones
RFS35	Eliminar instalación registrada	CU035	Instalador Profesional	Gestión de Instalaciones	Permitir eliminar registros de instalaciones
RFS36	Asignar tareas a instaladores	CU036	Administradores	Gestión de Instalaciones	Facilitar la asignación de tareas y seguimiento
RFS37	Actualizar estado de proyecto	CU037	Ingeniería	Gestión de Proyectos	Permitir actualización del estado de proyectos
RFS38	Generar reportes de progreso de proyectos	CU038	Estadísticas	Gestión de Estadísticas	Generar informes de progreso de proyectos

RFS39	Actualizar registro de postventa	CU039	Postventa	Gestión de Servicios	Permitir actualizar datos de postventa
RFS40	Eliminar registro de postventa	CU040	Postventa	Gestión de Servicios	Permitir eliminar registros de postventa
RFS41	Asignar técnicos para servicios	CU041	Administradores	Gestión de Servicios	Facilitar la asignación de técnicos a servicios
RFS42	Seguimiento de resolución de servicios	CU042	Soporte Técnico	Gestión de Servicios	Seguir el progreso y resolución de servicios
RFS43	Actualizar información de equipos	CU043	Soporte Técnico	Gestión de Equipos	Permitir actualizar detalles de equipos y sensores
RFS44	Eliminar equipo o sensor	CU044	Soporte Técnico	Gestión de Equipos	Permitir eliminar registros de equipos
RFS45	Realizar mantenimiento de equipos	CU045	Soporte Técnico	Gestión de Equipos	Facilitar el mantenimiento preventivo y correctivo
RFS46	Generar alertas de fallos en equipos	CU046	Centro de Monitoreo	Gestión de Equipos	Alertar sobre fallos en equipos y sensores
RFS47	Generar reportes financieros	CU047	Estadísticas	Gestión de Estadísticas	Generar informes financieros periódicos
RFS48	Gestionar cuentas por pagar y cobrar	CU048	Gestión Financiera	Gestión Financiera	Facilitar la gestión de cuentas
RFS49	Integrar con sistemas de pago electrónicos	CU049	Gestión Financiera	Gestión Financiera	Permitir la integración con sistemas de pago externos
RFS50	Actualizar información de pagos	CU050	Gestión Financiera	Gestión de Pagos	Permitir actualizar detalles de pagos

RFS51	Eliminar registro de pago	CU051	Gestión Financiera	Gestión de Pagos	Permitir eliminar registros de pagos
RFS52	Actualizar información de proveedores	CU052	Inventarios	Gestión de Inventarios	Permitir actualizar datos de proveedores
RFS53	Eliminar registro de proveedor	CU053	Inventarios	Gestión de Inventarios	Permitir eliminar registros de proveedores
RFS54	Buscar proveedores por varios criterios	CU054	Inventarios	Gestión de Inventarios	Facilitar la búsqueda de proveedores
RFS55	Generar reportes de compras a proveedores	CU055	Estadísticas	Gestión de Estadísticas	Generar informes de compras periódicos
RFS56	Evaluar desempeño de proveedores	CU056	Estadísticas	Gestión de Estadísticas	Evaluar y reportar sobre el desempeño de proveedores
RFS57	Registrar y actualizar información de empleados	CU057	Administradores	Gestión de Recursos Humanos	Facilitar la gestión de datos de empleados
RFS58	Asignar roles y permisos a empleados	CU058	Administradores	Gestión de Recursos Humanos	Permitir la asignación de roles y permisos
RFS	Gestionar horario y asistencia de empleados	CU059	Administradores	Gestión de Recursos Humanos	Facilitar la gestión de horarios y asistencias
RFS60	Generar reportes de desempeño de empleados	CU060	Estadísticas	Gestión de Estadísticas	Generar informes de desempeño de empleados
RFS61	Gestionar roles y permisos del sistema	CU061	Administradores	Seguridad y Configuración	Facilitar la gestión de roles y permisos
RFS62	Configurar parámetros de seguridad	CU062	Administradores	Seguridad y Configuración	Configurar y gestionar parámetros de seguridad

RFS63	Realizar auditorías de seguridad	CU063	Administradores	Seguridad y Configuración	Realizar y gestionar auditorías de seguridad
RFS64	Generar copias de seguridad de la base de datos	CU064	Administradores	Seguridad y Configuración	Facilitar la generación de copias de seguridad
RFS65	Integrar con sistemas de gestión ERP	CU065	Administradores	Integración y Automatización	Permitir la integración con sistemas ERP
RFS66	Automatizar envío de notificaciones	CU066	Administradores	Integración y Automatización	Automatizar el envío de notificaciones
RFS67	Implementar APIs para integraciones externas	CU067	Administradores	Integración y Automatización	Facilitar la integración con APIs externas
RFS68	Configurar reglas de negocio para automatización	CU068	Administradores	Integración y Automatización	Permitir la configuración de reglas de negocio
RFS69	Generar alertas y notificaciones en tiempo real	CU069	Administradores	Integración y Automatización	Facilitar la generación de alertas y notificaciones en tiempo real

TABLA CU001:

CASO DE USO:	CU001 REGISTRAR NUEVO CLIENTE	
ACTORES:	Ejecutivo de cuentas	
REFERENCIAS:	RFS03, RFS05	
DESCRIPCION	<ul style="list-style-type: none"> • Permite al ejecutivo de ventas registrar un cliente. 	
PRECONDICION:	<p>El ejecutivo de cuenta a iniciado sesión en el sistema</p> <p>El Ejecutivo de Ventas debe estar autenticado en el sistema</p> <p>El ejecutivo de cuenta tiene los permisos para agregar a un cliente</p>	
FLUJO PRINCIPAL:	1	El Ejecutivo de cuentas selección la opción “REGISTRAR NUEVO CLIENTE” en la interfaz de usuario
	2	El sistema muestra un formulario vacío para agregar los detalles del cliente
	3	<p>1. El Ejecutivo de Ventas selecciona la opción de registrar cliente.</p> <p>2. El sistema presenta un formulario de registro de cliente.</p> <p>3. El Ejecutivo de Ventas completa los campos requeridos (nombre, dirección, contacto, etc.).</p> <p>4. El Ejecutivo de Ventas envía el formulario.</p> <p>5. El sistema valida los datos y almacena la información del cliente en la base de datos.</p> <p>6. El sistema confirma la creación del cliente y muestra un mensaje de éxito.</p> <p>El ejecutivo de cuentas completa el formulario proporcionando la siguiente información:</p> <p>*DNI</p> <p>*CUIT/CUIL</p> <p>*RAZON SOCIAL EN CASO DE SER EMPRESA</p> <p>* APELLIDO Y NOMBRE</p> <p>*TELEFONO</p> <p>*DOMICILIO.</p> <p>*EMAIL.</p> <p>*CONSUMO DE CADA MES EN KWH POR UN AÑO</p> <p>*CONSUMO PROMEDIO EN KWH</p> <p>*EQUIPO INTERESADO EN ADQUIRIR</p>
POSTCONDICION:		La información del nuevo cliente se almacena en la base de datos.
FLUJO ALTERNATIVO		<p>Validaciones de datos con falla:</p> <p>1. Si los datos están incompletos, el sistema muestra un mensaje de error y solicita corrección. El sistema muestra el mensaje de ERROR EN LA CARGA DE DATOS.</p> <p>2. 2: EL sistema regresa al paso anterior para que el operador corrija los errores.</p>

EXCEPCIONES:		Si la base de datos no está disponible, el sistema muestra un mensaje de error y no se guarda la información.
--------------	--	---

TABLA CU002:

CASO DE USO:	CU002 ALMACENAR LOS CONTRATOS REALIZADOS AL CLIENTE	
ACTORES:	Administradores	
REFERENCIAS:	RFS02 Almacenar un contrato realizado por la administración y asignarlo al perfil del cliente	
DESCRIPCION	<ul style="list-style-type: none"> Asociar contratos a perfiles de clientes existentes en el sistema 	
PRECONDICION:	<p>El ejecutivo de cuenta a iniciado sesión en el sistema</p> <p>El Administrador debe estar autenticado en el sistema</p> <p>El ejecutivo de cuenta tiene los permisos para almacenar el contrato firmado digitalizado</p>	
FLUJO PRINCIPAL:	1	El Administrador selecciona la opción de asignar contrato a cliente.
	2	El sistema muestra un formulario vacío para agregar los detalles del contrato, al ingresar el DNI o CUIL/CUIL el sistema regresa los datos del cliente El sistema presenta una lista de clientes.
	3	El Administrador selecciona un cliente de la lista. El ejecutivo de cuentas completa el formulario proporcionando la siguiente información: *DNI *CUIT/CUIL *CONTRATO FIRMADO DIGITALIZADO.
	4	El Administrador carga el contrato correspondiente.
	5	El sistema valida y almacena el contrato asignándolo al perfil del cliente.
	6	El sistema confirma la operación y muestra un mensaje de éxito.
POSTCONDICION:	El contrato se asocia y almacena en el perfil del cliente.	
FLUJO ALTERNATIVO	<p>Validaciones de datos con falla:</p> <ol style="list-style-type: none"> Si el contrato no es válido, el sistema muestra un mensaje de error y solicita corrección. El sistema muestra un mensaje de ERROR. 2: EL sistema regresa al paso anterior para que el operador corrija los errores. 	
EXCEPCIONES:		Si la base de datos no está disponible, el sistema muestra un mensaje de error y no se guarda la información.

CASO DE USO:	CU003 REGISTRAR INFORMACION DE PRODUCTOS	
ACTORES:	Inventarios	
REFERENCIAS:	RF003: Registrar información del equipo comercializado y demás productos.	
DESCRIPCION	<ul style="list-style-type: none"> Permite al ejecutivo de cuentas registrar productos y equipos comercializados en el sistema 	
PRECONDICION:	<p>El ejecutivo de cuenta a iniciado sesión en el sistema</p> <p>El encargado de Inventarios debe estar autenticado en el sistema</p> <p>El ejecutivo de cuenta tiene los permisos para poder registrar y realizar la venta</p>	
FLUJO PRINCIPAL:	1	El encargado de Inventarios selecciona la opción de registrar producto.
	2	El sistema presenta un formulario de registro de producto.
	3	El encargado de Inventarios completa los campos requeridos (nombre, marca, modelo, características, etc.). El formulario proporcionando la siguiente información:
		*DNI *CUIT/CUIL *SELECCIÓN DE EQUIPOS EN STOCK: *INVESOR. *MARCA *MODELO *CANTIDAD *PANEL SOLAR: *MARCA *MODELO *CANTIDAD *BATERIAS: *MARCA *MODELO *CANTIDAD *TIPO DE TECHO: *MARCA *MODELO *CANTIDAD *SOPORTES: *MARCA *MODELO *CANTIDAD *CABLES: *MARCA *MODELO *CANTIDAD *FICHAS. *MARCA *MODELO *CANTIDAD

		*INGRESO DE DATOS DE LA INSTALACION: *DOMICILIO *CODIGO POSTAL. *LOCALIDAD *PROVINCIA *PAIS *HORARIO DISPONIBLE *VISTOR DE TOTAL A PAGAR
	4	El sistema muestra el consumo del cliente cargado con anterioridad cuando se registró. El encargado de Inventarios envía el formulario.
	5	El sistema valida los datos y almacena la información del producto en la base de datos
	6	El sistema confirma la creación del producto y muestra un mensaje de éxito.
	7	El sistema calcula el total de los equipos seleccionados y muestra al ejecutivo de cuentas e informa a área de caja
POSTCONDICION:		La información del producto se almacena en la base de datos.
FLUJO ALTERNATIVO		Si los datos están incompletos, el sistema muestra un mensaje de error y solicita corrección. Validaciones de datos con falla: <ol style="list-style-type: none"> 1. El sistema muestra un mensaje de ERROR 2. 2: EL sistema regresa al paso anterior para que el operador corrija los errores.
EXCEPCIONES:		Si la base de datos no está disponible, el sistema muestra un mensaje de error y no se guarda la información. *SELECCIÓN DE MODELO DE KITS DISPONIBLES EN STOCK. En este caso se selecciona alguno de los kits disponibles ya armados para su venta si el cliente opta por esta opción.

TABLA CU003:

CASO DE USO:	CU003 REGISTRAR LA VENTA	
ACTORES:	Ejecutivo de cuentas	
REFERENCIAS:	RF004: Registrar la venta	
DESCRIPCION	<ul style="list-style-type: none"> Permite al ejecutivo de ventas registrar la venta realizadas a clientes 	
PRECONDICION:	<p>El ejecutivo de cuenta a iniciado sesión en el sistema</p> <p>El Ejecutivo de Ventas debe estar autenticado en el sistema</p> <p>El ejecutivo de cuenta tiene los permisos para poder registrar y realizar la venta</p>	
FLUJO PRINCIPAL:	1	<p>El Ejecutivo de cuentas selección la opción "NUEVA VENTA" en la interfaz de usuario</p> <p>Ejecutivo de Ventas selecciona la opción de registrar venta</p>
	2	El sistema presenta un formulario de registro de venta.
	3	<p>El Ejecutivo de Ventas completa los campos requeridos (cliente, productos, cantidad, precio, etc.).</p> <p>El ejecutivo de ventas completa el formulario proporcionando la siguiente información:</p> <p>*DNI</p> <p>*CUIT/CUIL</p> <p>*SELECCIÓN DE EQUIPOS EN STOCK:</p> <p>*INVESOR.</p> <p>*MARCA</p> <p>*MODELO</p> <p>*CANTIDAD</p> <p>*PANEL SOLAR:</p> <p>*MARCA</p> <p>*MODELO</p> <p>*CANTIDAD</p> <p>*BATERIAS:</p> <p>*MARCA</p> <p>*MODELO</p> <p>*CANTIDAD</p> <p>*TIPO DE TECHO:</p> <p>*MARCA</p> <p>*MODELO</p> <p>*CANTIDAD</p> <p>*SOPORTES:</p> <p>*MARCA</p> <p>*MODELO</p> <p>*CANTIDAD</p> <p>*CABLES:</p> <p>*MARCA</p> <p>*MODELO</p> <p>*CANTIDAD</p> <p>*FICHAS.</p> <p>*MARCA</p> <p>*MODELO</p> <p>*CANTIDAD</p>

		<p>*INGRESO DE DATOS DE LA INSTALACION:</p> <p>*DOMICILIO</p> <p>*CODIGO POSTAL.</p> <p>*LOCALIDAD</p> <p>*PROVINCIA</p> <p>*PAIS</p> <p>*HORARIO DISPONIBLE</p> <p>*VISTOR DE TOTAL A PAGAR</p>
	4	El sistema muestra el consumo del cliente cargado con anterioridad cuando se registró.
	5	El Ejecutivo de Ventas envía el formulario
	6	El sistema muestra un mensaje de confirmación indicando que fue registrada la venta con éxito. El sistema valida los datos y almacena la información de la venta en la base de datos.
	7	<p>El sistema calcula el total de los equipos seleccionados y muestra al ejecutivo de cuentas e informa a área de caja</p> <p>El sistema confirma la creación de la venta y muestra un mensaje de éxito.</p>
POSTCONDICION:		<p>Se ha realizado con éxito la venta al cliente</p> <p>La información de la venta se almacena en la base de datos.</p>
FLUJO ALTERNATIVO		<p>Si los datos están incompletos, el sistema muestra un mensaje de error y solicita corrección.</p> <p>Validaciones de datos con falla:</p> <ol style="list-style-type: none"> 3. El sistema muestra un mensaje de ERROR 4. 2: El sistema regresa al paso anterior para que el operador corrija los errores.
EXCEPCIONES:		<p>Si la base de datos no está disponible, el sistema muestra un mensaje de error y no se guarda la información.</p> <p>*SELECCIÓN DE MODELO DE KITS DISPONIBLES EN STOCK.</p> <p>En este caso se selecciona alguno de los kits disponibles ya armados para su venta si el cliente opta por esta opción.</p>

CASO DE USO:	CU006 REGISTRAR PAGOS DE CLIENTES	
ACTORES:	Gestión Financiera	
REFERENCIAS:	RF006: Registrar los pagos de los clientes	
DESCRIPCION	<ul style="list-style-type: none"> Registrar pagos realizados por los clientes en el sistema 	
PRECONDICION:	<p>El ejecutivo de ventas, administradores a a iniciado sesión en el sistema</p> <p>El encargado de Gestión Financiera debe estar autenticado en el sistema</p> <p>El ejecutivo de cuenta tiene los permisos para poder registrar el pago de los clientes</p>	
FLUJO PRINCIPAL:	1	El encargado de Gestión Financiera selecciona la opción de registrar pago.
	2	El sistema presenta un formulario de orden de trabajo.
	3	<p>El sistema presenta un formulario de registro de pago.
 3. El encargado de Gestión Financiera completa los campos requeridos (cliente, monto, método de pago, fecha, etc.).</p> <p>El formulario proporcionando la siguiente información del cliente</p> <ul style="list-style-type: none"> *DNI *CUIL/CUIT *IMPORTE A COBRAR *METODO DE PAGO *EFECTIVO *DEBITO *CREDITO *IMPORTE QUE PAGA *EFECTIVO *DEBITO *CREDITO
	4	El encargado de Gestión Financiera envía el formulario.
	5	El sistema valida los datos y almacena la información del pago en la base de datos
	6	El sistema confirma la creación del registro de pago y muestra un mensaje de éxito.
POSTCONDICION:	La información del pago se almacena en la base de datos.	
FLUJO ALTERNATIVO		<p>Si los datos están incompletos, el sistema muestra un mensaje de error y solicita corrección.</p> <p>Validaciones de datos con falla:</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje de ERROR 2. 2: EL sistema regresa al paso anterior para que el operador corrija los errores.
EXCEPCIONES:	Si la base de datos no está disponible, el sistema muestra un mensaje de error y no se guarda la información.	

CASO DE USO:	CU007 Generar recibo por cada pago recibido	
ACTORES:	Gestión Financiera	
REFERENCIAS:	RF007: Generar recibo por cada pago recibido de acuerdo al cliente que paga	
DESCRIPCION	<ul style="list-style-type: none"> Generar recibos automáticamente por cada pago recibido 	
PRECONDICION:	El ejecutivo de cuenta a iniciado sesión en el sistema El pago debe estar registrado en el sistema El de Gestión Financiera tiene los permisos para poder registrar y realizar el recibo correspondiente	
FLUJO PRINCIPAL:	1	El sistema detecta el registro de un nuevo pago
	2	El sistema genera un recibo correspondiente al pago realizado.
	3	El personal de gestión financiera completa el formulario proporcionando la siguiente información del cliente *DNI *CUIL/CUIT *IMPORTE A COBRAR *METODO DE PAGO *EFECTIVO *DEBITO *CREDITO *IMPORTE QUE PAGA *EFECTIVO *DEBITO *CREDITO
	4	El sistema muestra el importe cobrado y el saldo a pagar en caso de que no sea el total abonado.
	5	El ejecutivo de cuenta confirma la entrada de los datos. Elige la opción guardar.
	6	El sistema muestra un mensaje de confirmación indicando que fue registrado el pago parcial o total.
	7	El sistema emite el recibo correspondiente. El recibo se genera, se envía al cliente. El sistema envía el recibo al cliente y lo almacena en la base de dato
POSTCONDICION:	Se ha realizado con éxito el cobro al cliente	
FLUJO ALTERNATIVO	Validaciones de datos con falla: 3. El sistema muestra un mensaje de ERROR 4. 2: EL sistema regresa al paso anterior para que el operador corrija los errores.	
EXCEPCIONES:	Si la generación del recibo falla, el sistema muestra un mensaje de error y reintenta.	

CASO DE USO:	CU008 GENERAR FACTURA AL CLIENTE CUANDO EL PAGO RECIBIDO SEA EL TOTAL	
ACTORES:	Gestión Financiera	
REFERENCIAS:	RF008: Generar factura al cliente cuando el pago recibido sea el total	
DESCRIPCION	<ul style="list-style-type: none"> Generar facturas automáticamente cuando se recibe el pago total 	
PRECONDICION:	<p>El equipo de Gestión Financiera ha iniciado sesión en el sistema</p> <p>El pago total debe estar registrado en el sistema</p> <p>El equipo de Gestión Financiera tiene los permisos para poder registrar y generar la factura</p>	
FLUJO PRINCIPAL:	1	El sistema detecta que se ha recibido el pago total de una venta.
	2	El sistema genera una factura correspondiente al pago total realizado.
	3	<p>El sistema envía la factura al cliente y la almacena en la base de datos.</p> <p>El personal de gestión financiera completa el formulario proporcionando la siguiente información del cliente</p> <p>*DNI</p> <p>*CUIL/CUIT</p> <p>*TIPO DE FACTURA (*A, *B)</p> <p>*IMPORTE A FACTURAR</p> <p>*METODO DE PAGO: (*EFECTIVO*DEBITO*CREDITO)</p> <p>*IMPORTE QUE PAGAGO</p> <p>*EFECTIVO*DEBITO*CREDITO</p>
	4	El sistema confirma la generación y envío de la factura.
POSTCONDICION:		La factura se genera, se envía al cliente y se almacena en la base de datos.
FLUJO ALTERNATIVO		
EXCEPCIONES:		Si la generación de la factura falla, el sistema muestra un mensaje de error y reintenta.

CASO DE USO:	CU009 REGISTRAR PROYECTOS DE ENERGIA SOLAR	
ACTORES:	Ingeniería	
REFERENCIAS:	RF009: Registrar el proyecto de energía solar, asignar recursos, planificar tareas y realizar seguimiento del progreso	
DESCRIPCION	<ul style="list-style-type: none"> Planificar y asignar recursos para proyectos de energía solar, así como realizar seguimiento del progreso 	
PRECONDICION:	El Ingeniero ha iniciado sesión en el sistema El Ingeniero debe estar autenticado en el sistema El Ingeniero tiene los permisos para poder registrar el proyecto de energía solar	
FLUJO PRINCIPAL:	1	El Ingeniero selecciona la opción de registrar proyecto.
	2	El sistema presenta un formulario de registro de proyecto.
	3	El Ingeniero completa los campos requeridos (nombre del proyecto, cliente, recursos, tareas, etc.). Completa el formulario proporcionando la siguiente información del cliente *DNI *CUIL/CUIT *NUMERO DE PROYECTO *NUMERO DE INSTALACION *NUMERO DE CLIENTE *RECURSOS A UTILIZAR *PRODUCTOS *TAREAS *PERSONAL A CARGO *FECHA INICIO *CANTIDAD DE DIAS EJECUCION *FECHA FINAL *COMENTARIOS *FOTOS *VISITAS *DATOS RELEVANTES
	4	El Ingeniero envía el formulario.
	5	El sistema valida los datos y almacena la información del proyecto en la base de datos.
	6	El sistema confirma la creación del proyecto y muestra un mensaje de éxito.
POSTCONDICION:		La información del proyecto se almacena en la base de datos.
FLUJO ALTERNATIVO		Si los datos están incompletos, el sistema muestra un mensaje de error y solicita corrección. Validaciones de datos con falla: <ol style="list-style-type: none"> El sistema muestra un mensaje de ERROR 2: EL sistema regresa al paso anterior para que el operador corrija los errores.
EXCEPCIONES:		Si la base de datos no está disponible, el sistema muestra un mensaje de error y no se guarda la información.

CASO DE USO:	CU010 REGISTRAR STOCK DEBITADO TRAS LA VENTA	
ACTORES:	Inventarios	
REFERENCIAS:	RF010: Registrar el stock debitado luego de la venta y asignación de recursos	
DESCRIPCION	<ul style="list-style-type: none"> Registrar la disminución de stock tras la venta y asignación de recursos 	
PRECONDICION:	El de inventarios ha iniciado sesión en el sistema La venta debe estar registrada en el sistema El de inventarios tiene los permisos para poder rebitar el stock tras la venta.	
FLUJO PRINCIPAL:	1	El encargado de Inventarios selecciona la opción de registrar stock debitado
	2	El sistema presenta una lista de ventas recientes.
	3	El encargado de Inventarios selecciona la venta correspondiente.
	4	El sistema actualiza el stock de los productos vendidos.
	5	El sistema confirma la actualización del stock y muestra un mensaje de éxito.
POSTCONDICION:		El stock de los productos vendidos se actualiza en la base de datos.
FLUJO ALTERNATIVO		Si la venta no es válida, el sistema muestra un mensaje de error y solicita corrección. Validaciones de datos con falla: <ol style="list-style-type: none"> El sistema muestra un mensaje de ERROR EL sistema regresa al paso anterior para que el operador corrija los errores.
EXCEPCIONES:		Si la base de datos no está disponible, el sistema muestra un mensaje de error y no se actualiza el stock.

CASO DE USO:	CU011 REGISTRAR INSTALACION REALIZADA	
ACTORES:	Instalador Profesional	
REFERENCIAS:	RF011: Registrar la instalación realizada y datos del personal que la realizó	
DESCRIPCION	<ul style="list-style-type: none"> Registrar detalles de las instalaciones realizadas por el personal 	
PRECONDICION:	El instalador profesional ha iniciado sesión en el sistema El Instalador Profesional debe estar autenticado en el sistema El de inventarios tiene los permisos para poder registrar la instalacion realizada	
FLUJO PRINCIPAL:	1	El Instalador Profesional selecciona la opción de registrar instalación.
	2	El sistema presenta un formulario de registro de instalación
	3	El Instalador Profesional completa los campos requeridos (cliente, dirección, productos instalados, fecha, etc.).
		* DNI * CUIT * NOMBRE * APELLIDO * DOMICILIO CLIENTE * TELEFONO CLIENTE * LOCALIDAD * PROVINCIA * PAIS * PERSONA RESPONSABLE CLIENTE * DNI * TELEFONO * DOMICILIO * LOCALIDAD * PROVINCIA * PAIS * COORDENADAS INSTALACION * SUPERFICIE * CANTIDAD DE MATERIALES: * PANELES * CANTIDAD DE PANELES * SOPORTES * TIPO DE SOPORTES * CANTIDAD DE SOPORTES * FUSIBLES * TIPOS DE FUSIBLES * CANTIDAD DE FUSIBLES * SECCIONES * MTS DE CABLE * FICHAS

		*TIPOS DE FICHAS *BATERIAS *TIPOS DE BATERIAS *CANTIDAD DE BATERIAS *CONEXIONADO *TIPO DE CONEXIONADO *TIPO DE MANTENIMIENTO *FECHA PRIMER MANTENIMIENTO APROXIMADO *COMENTARIOS *CANTIDAD DE PERSONAS EN INSTALACION *APELLIDO *NOMBRE *DNI *MATRICULA DEL INSTALADOR *FECHA DE ENTREGA OBRA *COMENTARIOS DE OBRA *REQUERIMIENTOS ESPECIALES *REQUERIMIENTOS INESPERADOS *SUPERVISOR *DNI SUPERVISOR *APELLIDO SUPERVISOR *NOMBRE SUPERVISOR *MATRICULA *COSTO DE LA OBRA *IMPORTE TOTAL ABONADO
	4	El Instalador Profesional envía el formulario
	5	El sistema valida los datos y almacena la información de la instalación en la base de datos
	6	El sistema confirma la creación del registro de instalación y muestra un mensaje de éxito.
POSTCONDICION:		La información de la instalación se almacena en la base de datos.
FLUJO ALTERNATIVO		Si los datos están incompletos, el sistema muestra un mensaje de error y solicita corrección. Si la venta no es válida, el sistema muestra un mensaje de error y solicita corrección. Validaciones de datos con falla: <ol style="list-style-type: none"> 9. El sistema muestra un mensaje de ERROR 10. 2: EL sistema regresa al paso anterior para que el operador corrija los errores.

EXCEPCIONES:		Si la base de datos no está disponible, el sistema muestra un mensaje de error y no se guarda la información.
--------------	--	---

CASO DE USO:	CU012 CONTROLAR SISTEMA DE GENERACION DISTRIBUIDA	
ACTORES:	Operador	
REFERENCIAS:	RF012: El sistema de generación distribuida puede por parte del operador: Iniciar, detener, reiniciar o configurarlo en remoto	
DESCRIPCION	<ul style="list-style-type: none"> Permitir el control remoto del sistema de generación distribuida (iniciar, detener, reiniciar, configurar) 	
PRECONDICION:	El operador ha iniciado sesión en el sistema El Operador debe estar autenticado en el sistema El operador tiene los permisos para poder registrar y realizar la venta	
FLUJO PRINCIPAL:	1	El Operador selecciona la opción de control del sistema de generación distribuida.
	2	El sistema presenta las opciones de control disponibles (iniciar, detener, reiniciar, configurar).
	3	El Operador selecciona la acción deseada.
	4	El sistema ejecuta la acción seleccionada y actualiza el estado del sistema
	5	El sistema confirma la ejecución de la acción y muestra un mensaje de éxito.
POSTCONDICION:		El sistema de generación distribuida se actualiza según la acción seleccionada.
FLUJO ALTERNATIVO		Si la acción no es válida, el sistema muestra un mensaje de error y solicita corrección. Validaciones de datos con falla: 11. El sistema muestra un mensaje de ERROR 12. 2: EL sistema regresa al paso anterior para que el operador corrija los errores.
EXCEPCIONES:		Si el sistema de generación distribuida no responde, el sistema muestra un mensaje de error y reintenta

CASO DE USO:	CU0013 REGISTRAR INFORMACION GENERADA DEL EQUIPO	
ACTORES:	Soporte Técnico	
REFERENCIAS:	RF013: Registrar información generada del equipo, sensores, estadísticas	
DESCRIPCION	<ul style="list-style-type: none"> Registrar información generada por los equipos y sensores instalados 	
PRECONDICION:	<p>El encargado de Soporte Técnico ha iniciado sesión en el sistema</p> <p>El encargado de Soporte Técnico debe estar autenticado en el sistema</p> <p>El Soporte Técnico tiene los permisos para poder registrar y realizar la venta</p>	
FLUJO PRINCIPAL:	1	El encargado de Soporte Técnico selecciona la opción de registrar información del equipo
	2	El sistema presenta un formulario de registro de información
	3	<p>El encargado de Soporte Técnico completa los campos requeridos (equipo, datos de sensores, estadísticas, fecha, etc.).</p> <p>*FECHA</p> <p>*EQUIPO</p> <p>*NUMERO DE EQUIPO</p> <p>*HORAS DE USO</p> <p>*DATOS DE SENSORES</p> <p>*SENSOR_1</p> <p>*SENSOR_2</p> <p>*SENSOR_3</p> <p>*SENSOR_4</p> <p>*SENSOR_5</p> <p>*INTERVALOS DE MEDICION</p> <p>*VALORES NO CONTEMPLADOS</p> <p>*FALLA</p> <p>*TIPO DE FALLA</p> <p>*DESSCRIPCION</p> <p>*COMENTARIO</p>
	4	El encargado de Soporte Técnico envía el formulario
	5	El sistema valida los datos y almacena la información en la base de datos.
	6	El sistema confirma la creación del registro de información y muestra un mensaje de éxito.
POSTCONDICION:		La información generada del equipo se almacena en la base de datos.
FLUJO ALTERNATIVO		<p>Si los datos están incompletos, el sistema muestra un mensaje de error y solicita corrección.</p> <p>Validaciones de datos con falla:</p> <p>13. El sistema muestra un mensaje de ERROR</p> <p>14. 2: EL sistema regresa al paso anterior para que el operador corrija los errores.</p>
EXCEPCIONES:		Si la base de datos no está disponible, el sistema muestra un mensaje de error y no se guarda la información.

BASE DE DATOS MYSQL:

La base de datos se llama: sistema_gestion, consta de las siguientes tablas:

- **CLIENTES:** registra los clientes
- **VENTAS:** registra todas las ventas
- **OPERADORES:** almacena todas las funcionalidades y procedimientos realizadas por los operadores del centro de operaciones.
- **USUARIOS:** todos los usuarios del sistema con sus roles respectivos
- **INSTALADORES:** registra los instaladores capacitados que han realizado instalaciones para la empresa, pueden ser propios o externos. También las ordenes de trabajo
- **INGENIERIA:** registra los proyectos realizados o próximos a realizarse
- **FACTURAS:** registra todas las facturas emitidas, con sus respectivos estados
- **PAGOS:** registra los pagos recibidos y/o emitidos
- **PRODUCTOS:** Registra los productos disponibles para la venta
- **PROVEEDORES:** Registra los productos adquiridos a los diferentes proveedores como así también la información de cada uno de ellos
- **MONITORIZACION:** registra toda la información procedente de los equipos de generación distribuida
- **ESTADISTICAS:** almacena las estadísticas de los equipos y las generadas por la empresa
- **POSTVENTA:** almacena información de postventa realizado a cada cliente.

CODIGO:

<https://github.com/albertoaguirre/seminario21>

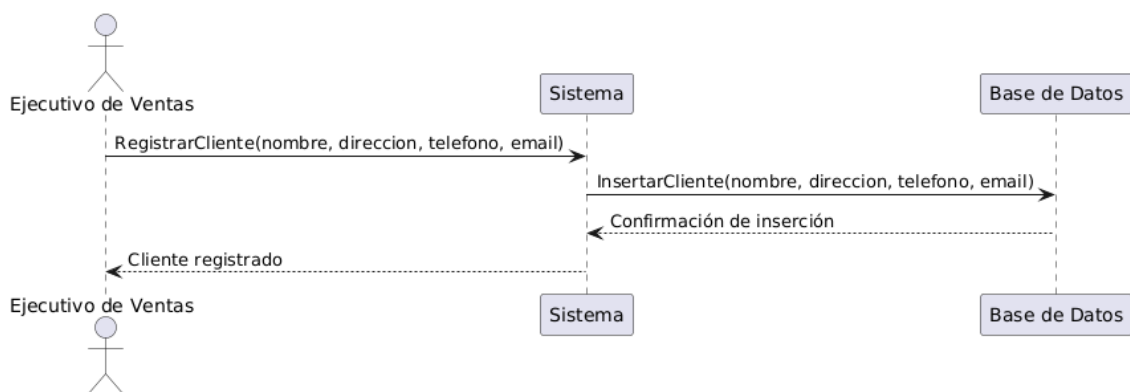
SEGUNDA ENTREGA:

Eta de Análisis:

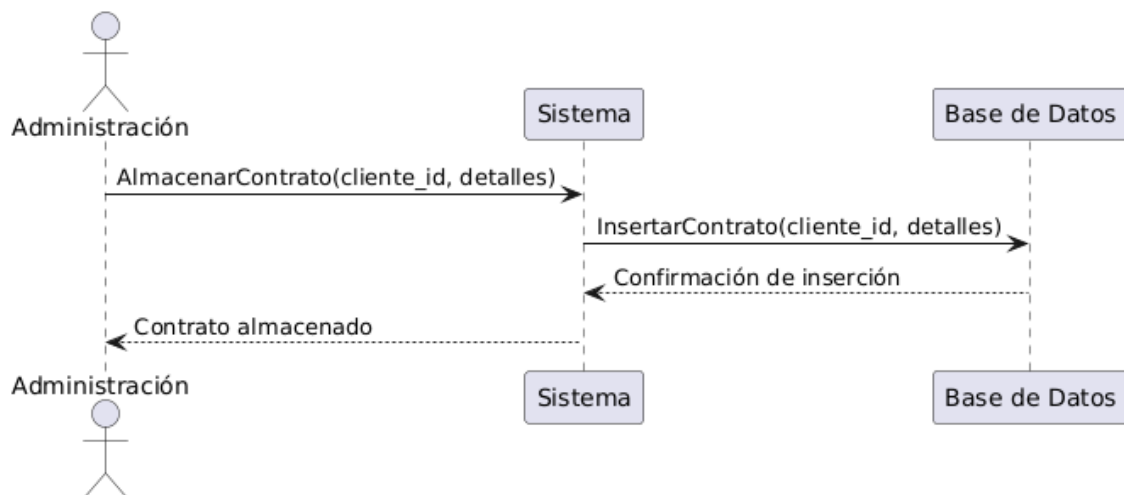
Diagramas de Secuencia de los CU.

Herramienta utilizada: Plant UML. <https://plantuml.com/es/>

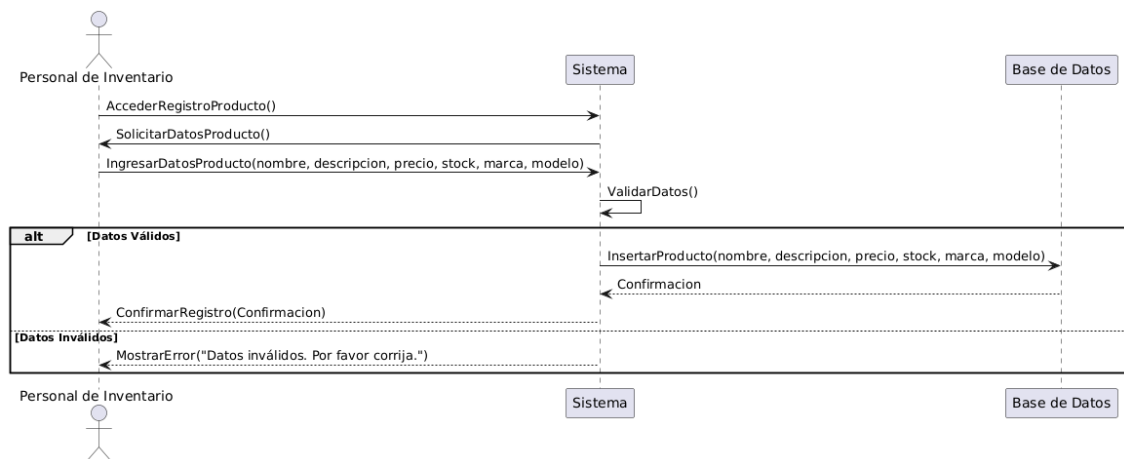
CU 001: REGISTRAR NUEVO CLIENTE:



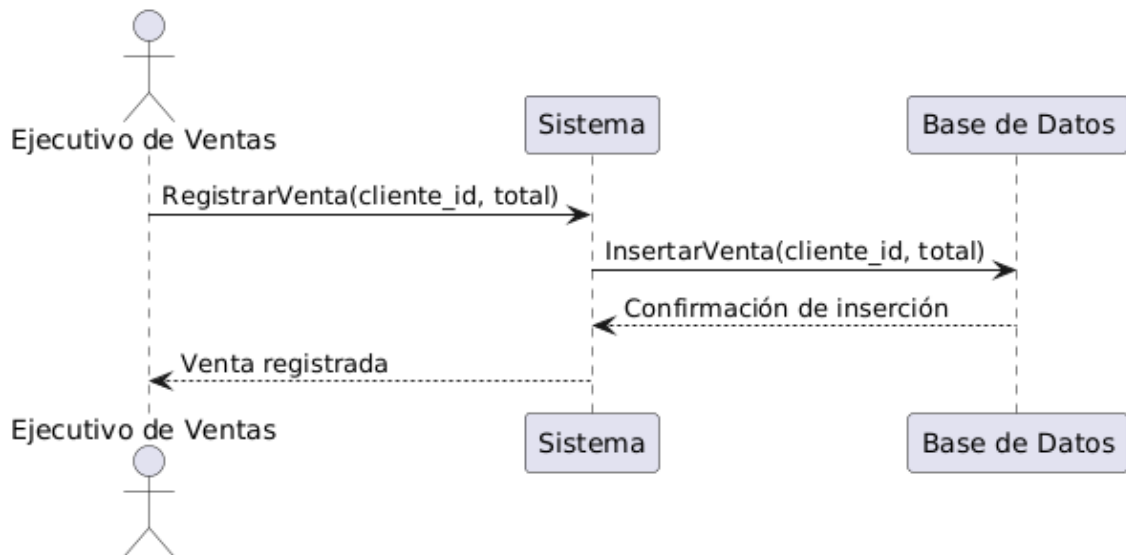
CU 002: ALMACENAR UN CONTRATO:



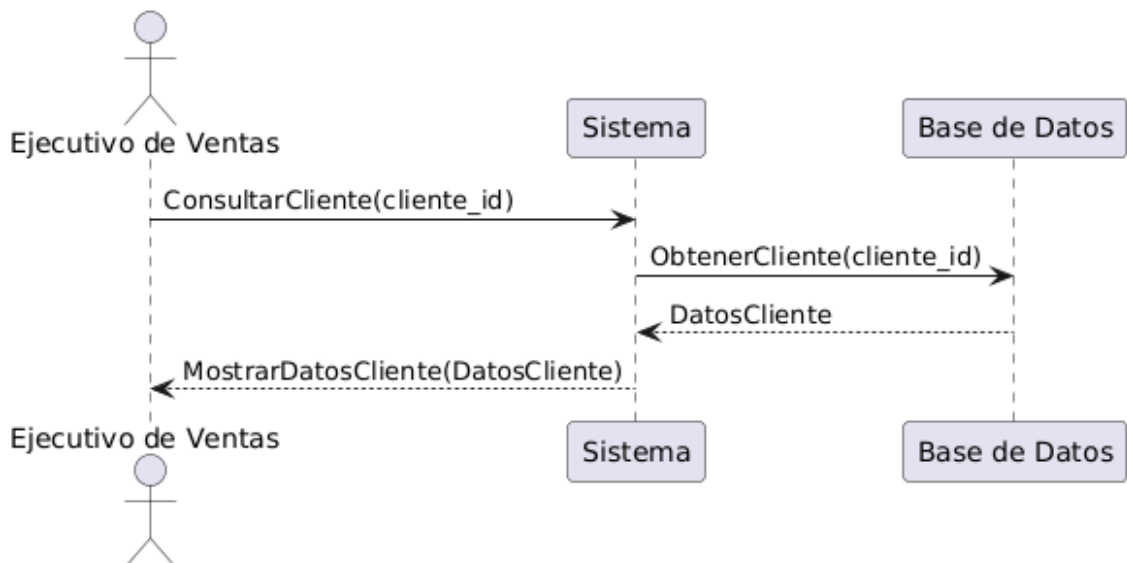
CU 003: REGISTRAR INFORMACION DE LOS PRODUCTOS EN EL SISTEMA:



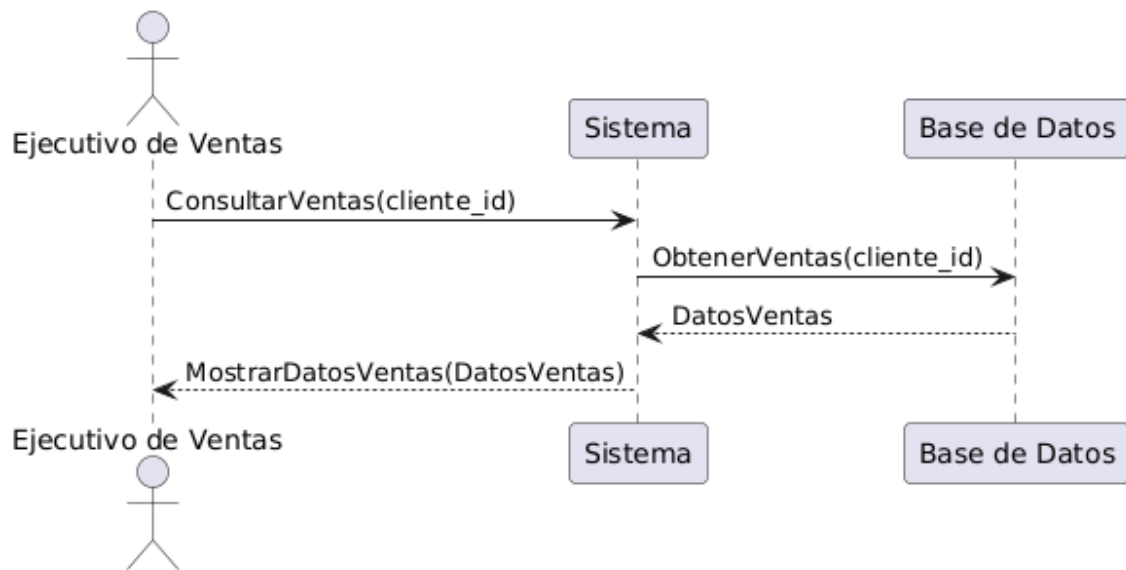
CU 004: REGISTRAR LA VENTA:



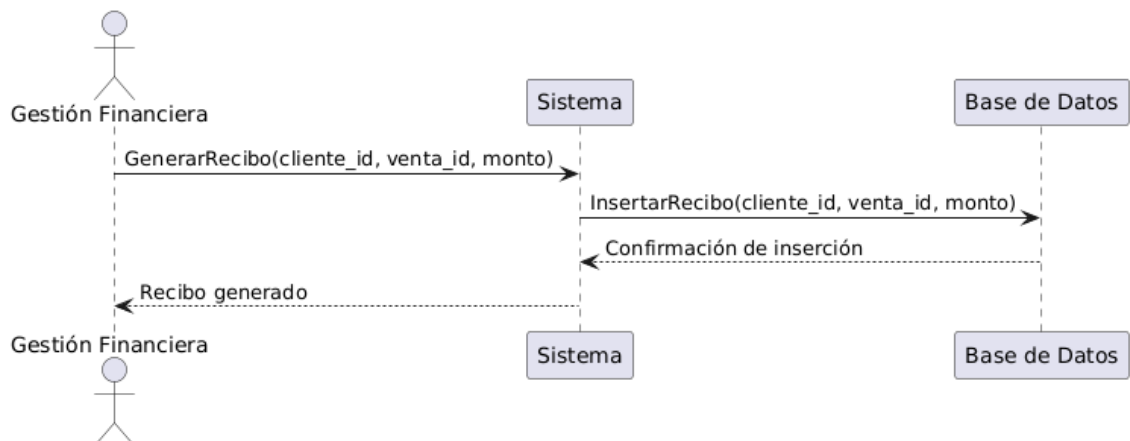
CU 005: CONSULTAR CLIENTE/S:



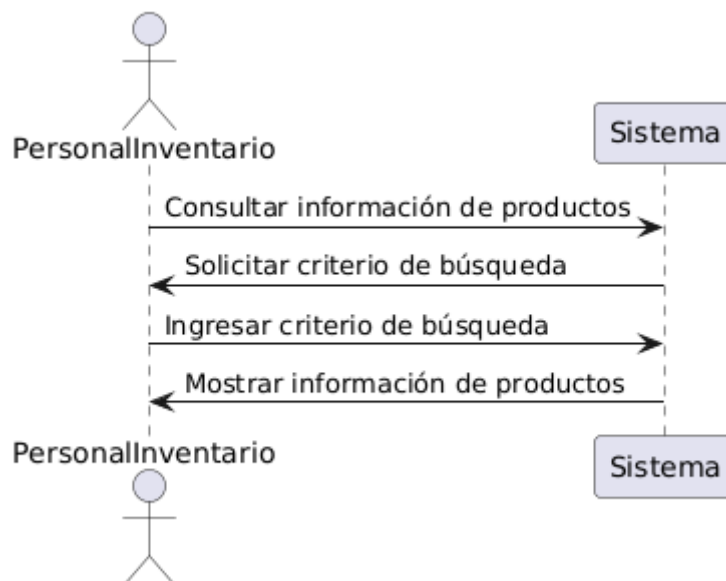
CU 006: CONSULTAR VENTAS:



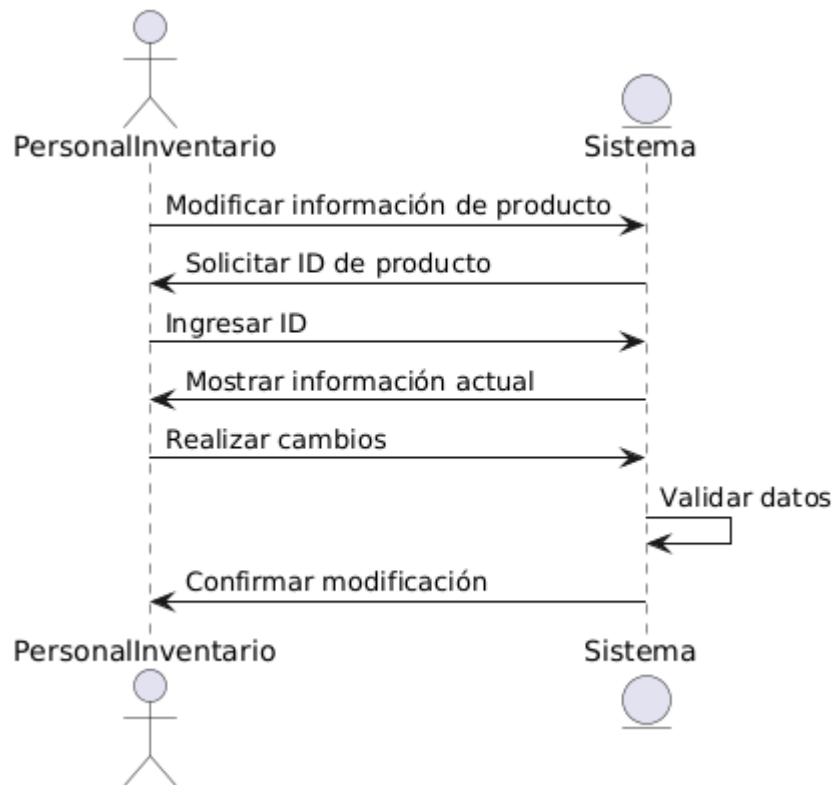
CU 007: GENERAR RECIBO POR PAGO:



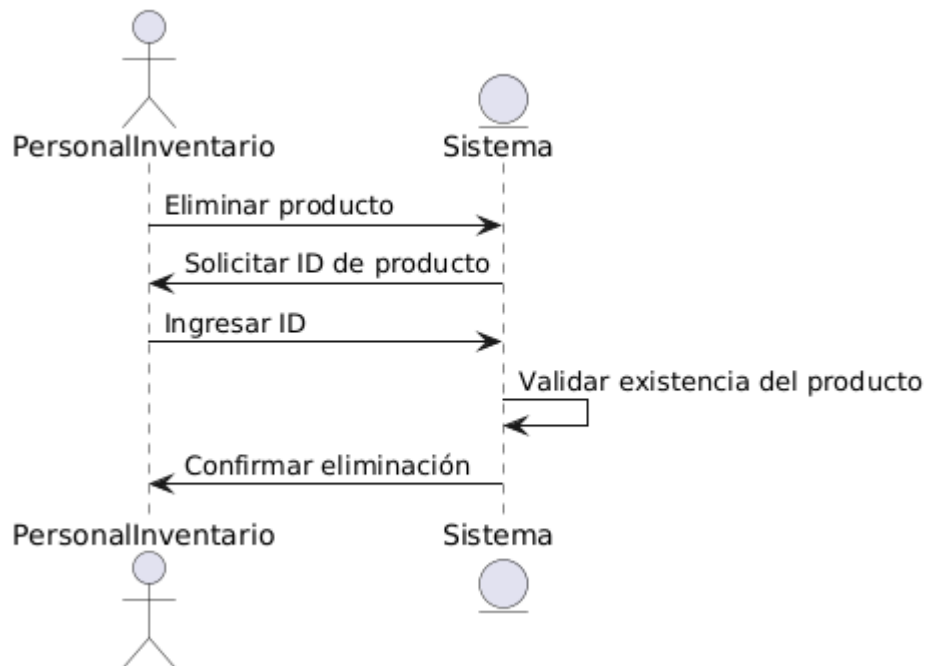
CU 008: CONSULTAR INFORMACION DE PRODUCTOS:



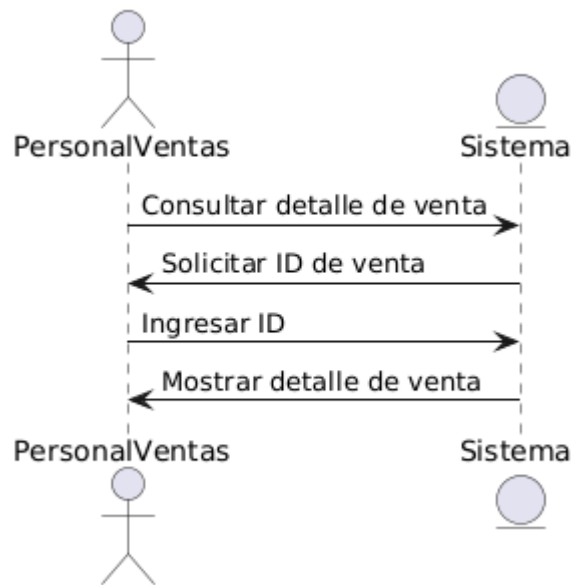
CU 009: MODIFICAR INFORMACION DE PRODUCTOS:



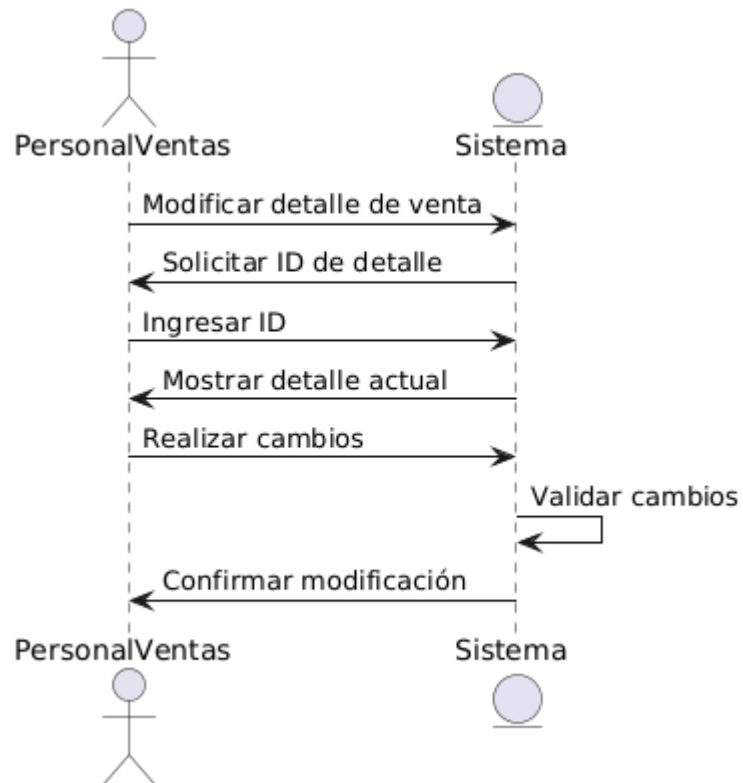
CU010: ELIMINAR PRODUCTOS:



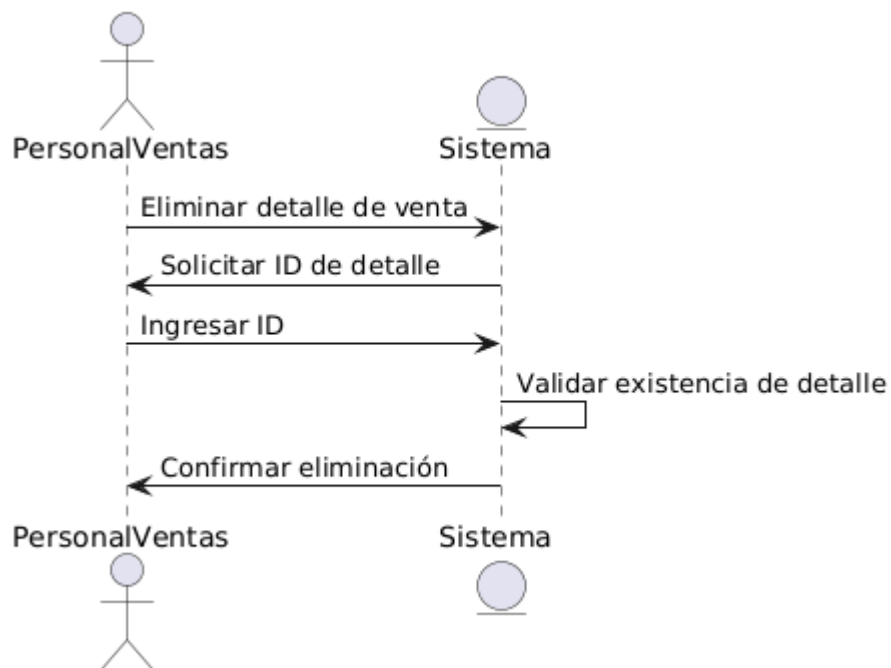
CU011: CONSULTAR DETALLE VENTAS:



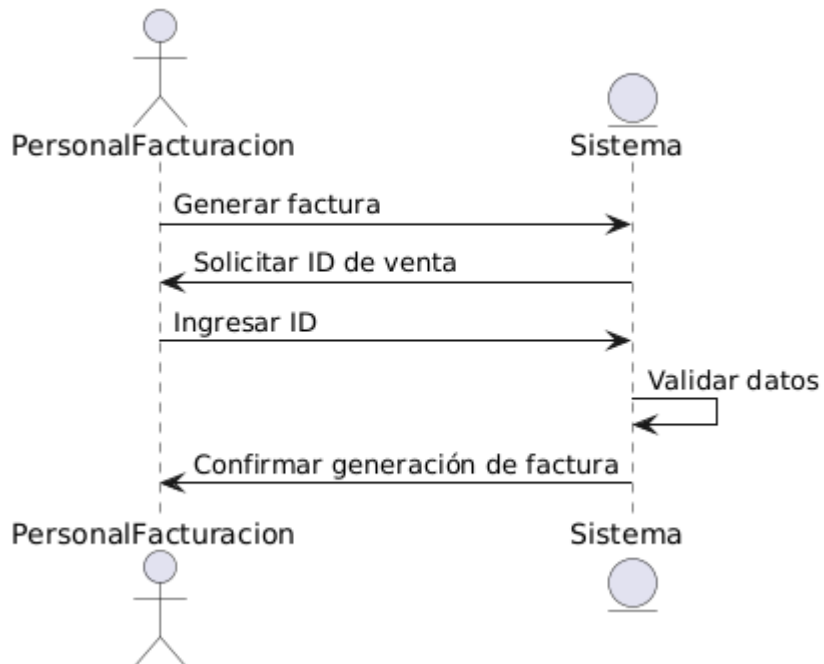
CU012: MODIFICAR DETALLE VENTAS:



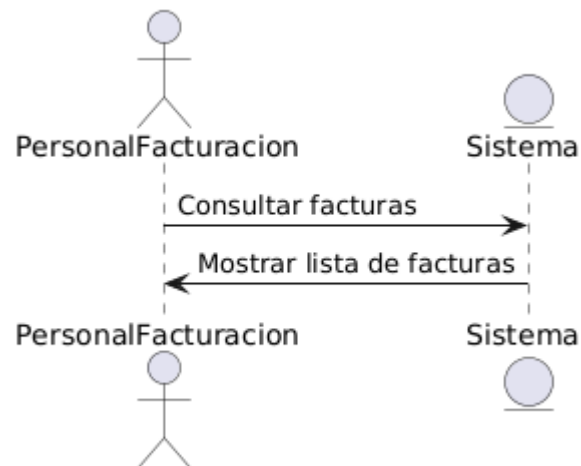
CU013: ELIMINAR DETALLE VENTAS:



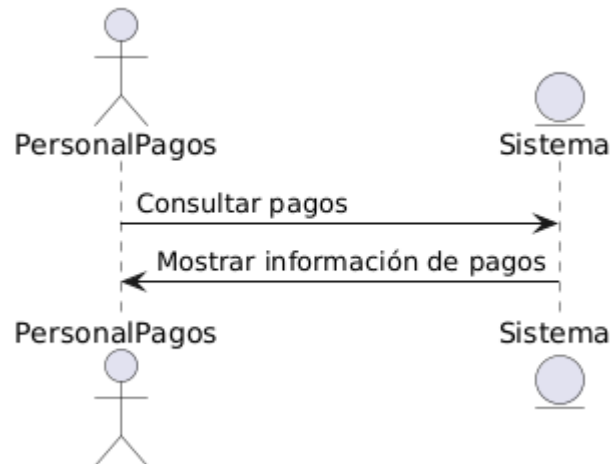
CU014: GENERAR FACTURA/S:



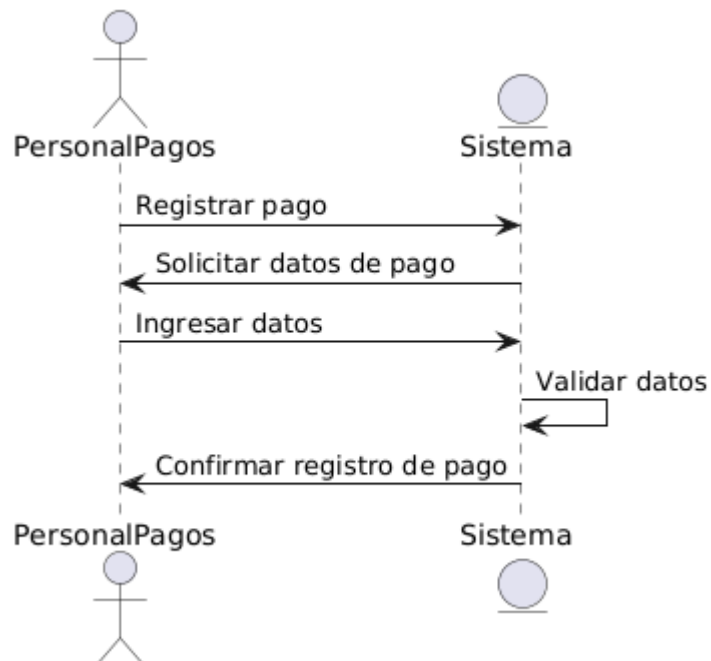
CU015: CONSULTAR FACTURAS:



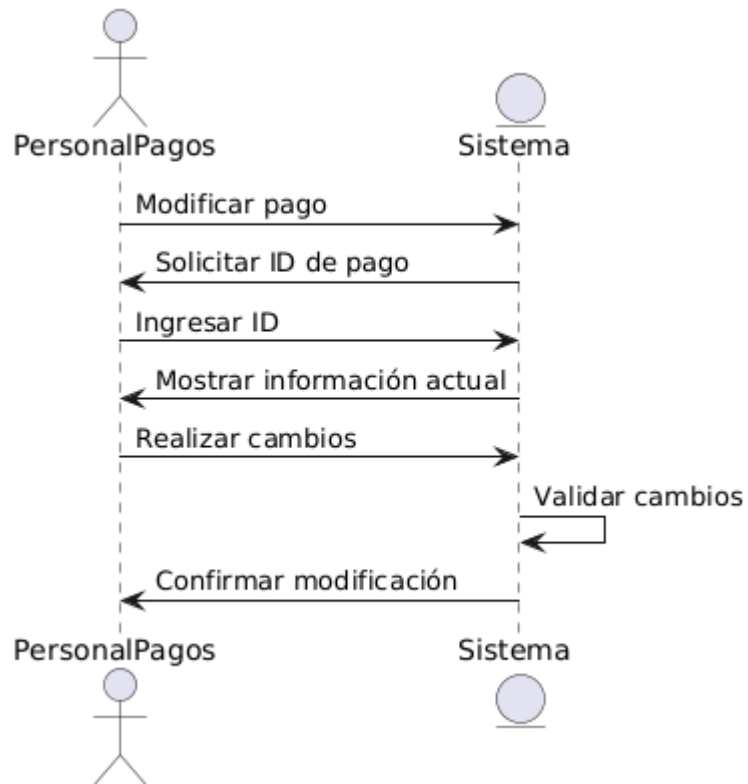
CU016: CONSULTAR PAGOS:



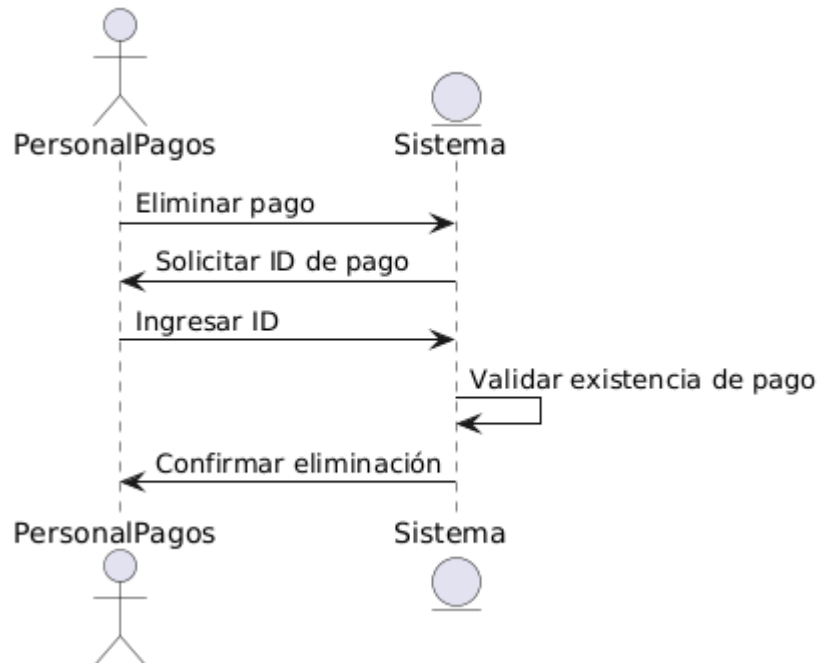
CU017: REGISTRAR PAGOS:



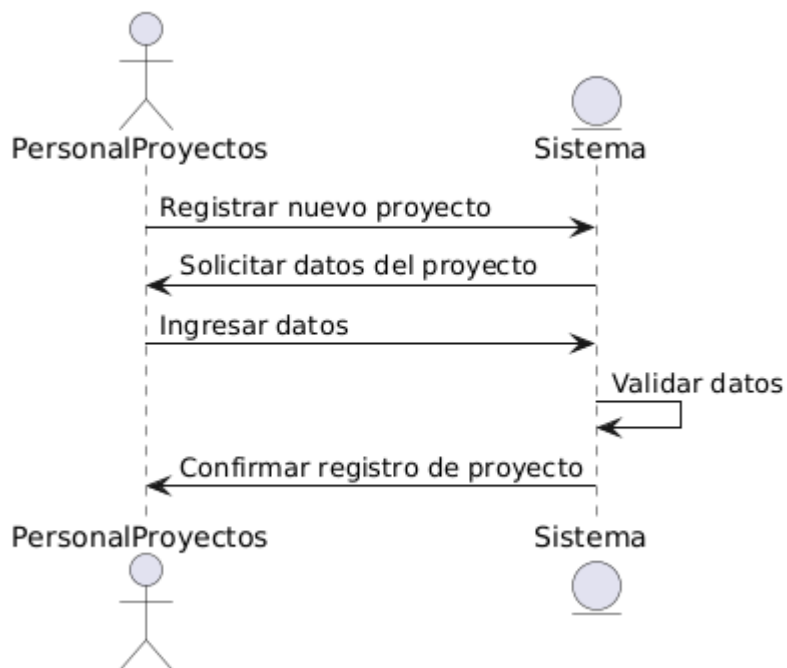
CU018: MODIFICAR PAGOS:



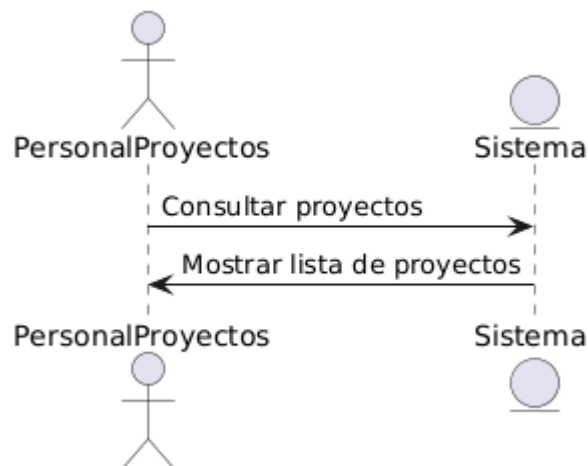
CU019: ELIMINAR PAGOS:



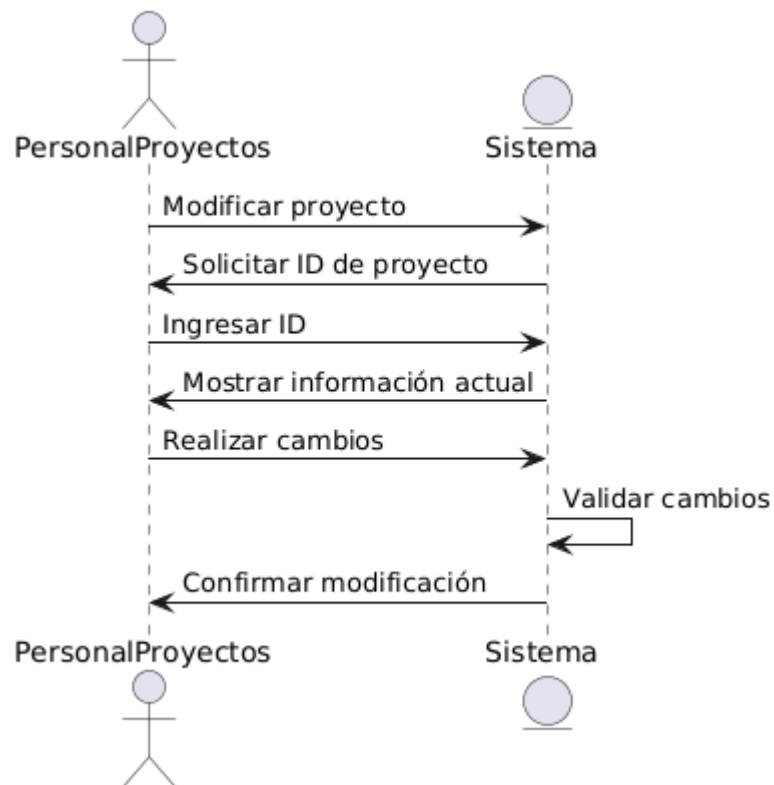
CU020: REGISTRAR PROYECTOS:



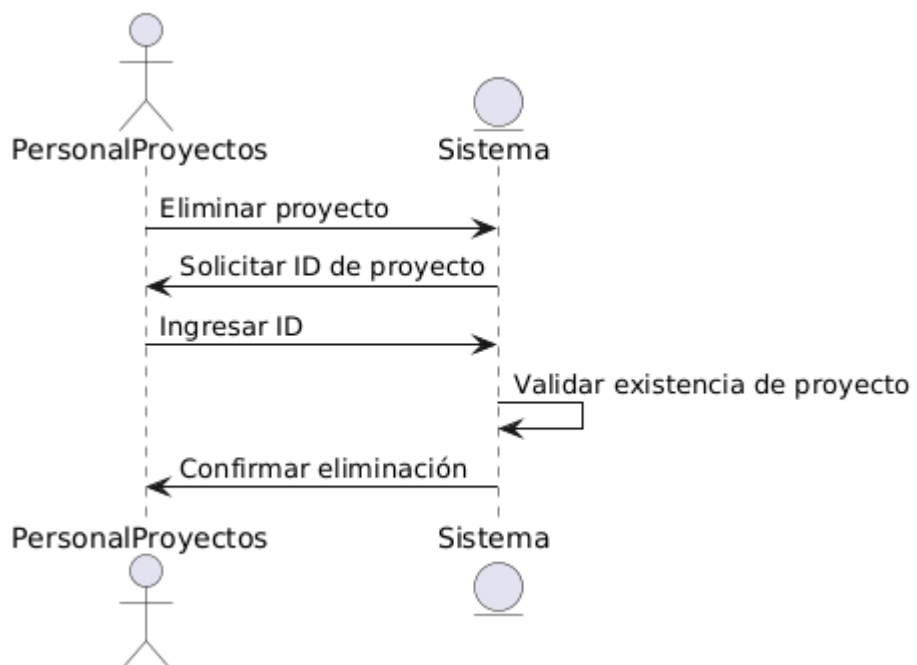
CU021: CONSULTAR PROYECTOS:



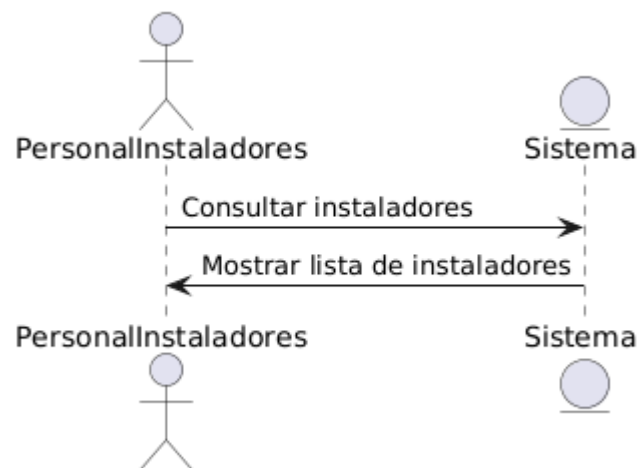
CU022 MODIFICAR PROYECTOS:



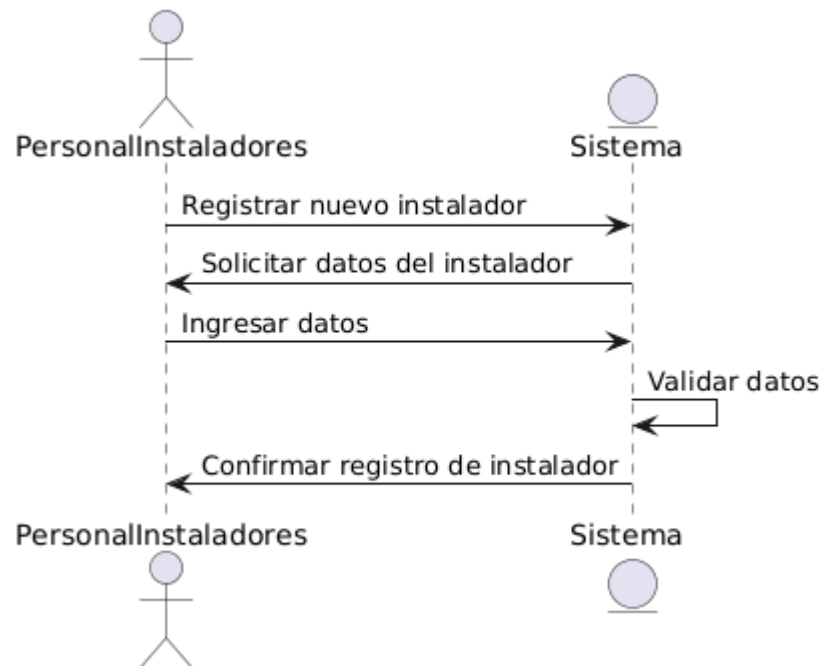
CU023: ELIMINAR PROYECTOS:



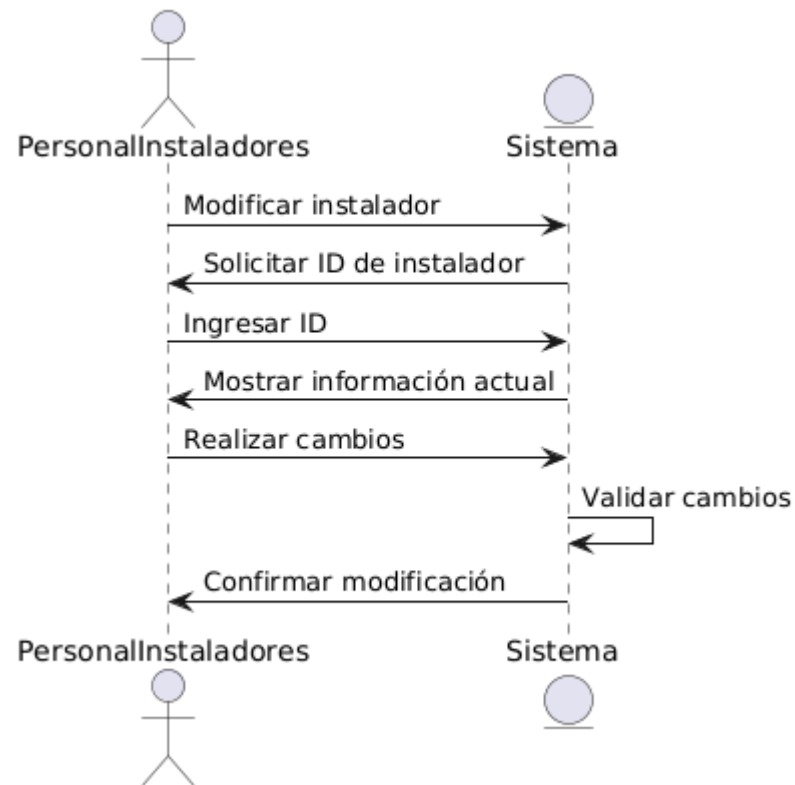
CU024: CONSULTAR INSTALADORES:



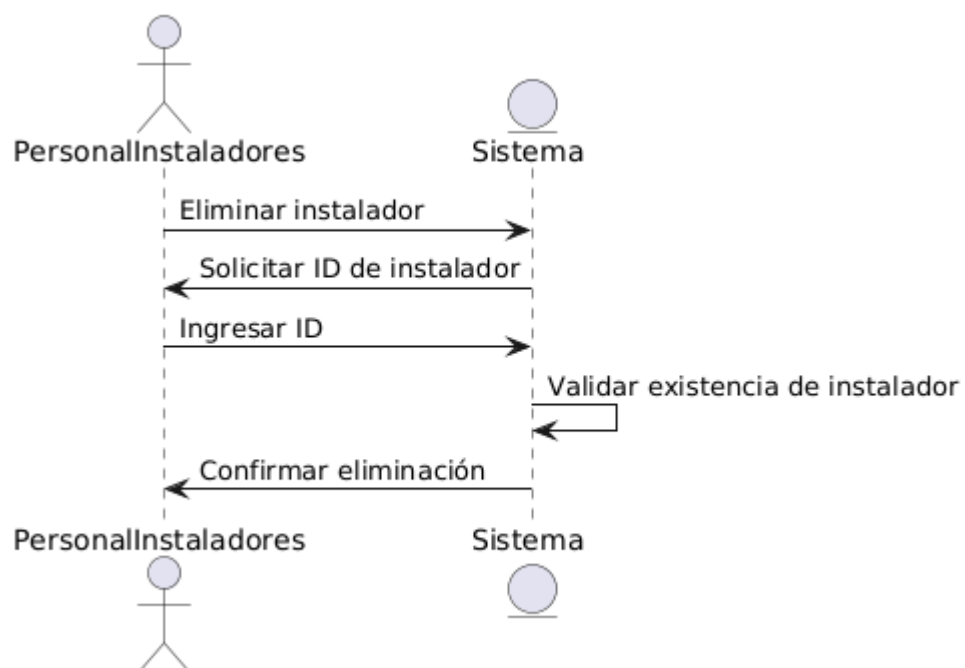
CU025: REGISTRAR INSTALADORES:



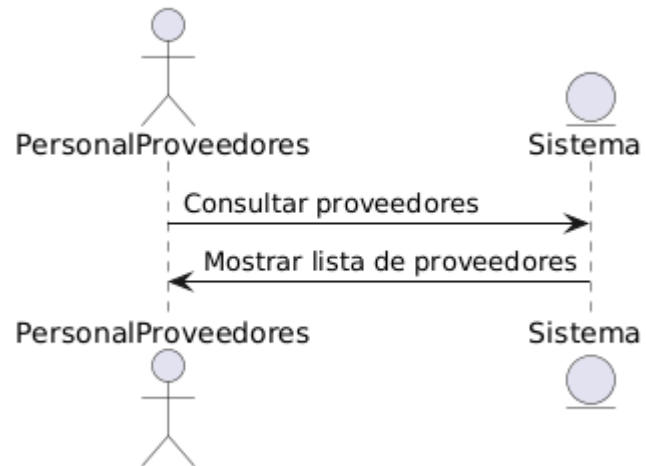
CU026: MODIFICAR INSTALADORES:



CU027: ELIMINAR INSTALADORES:

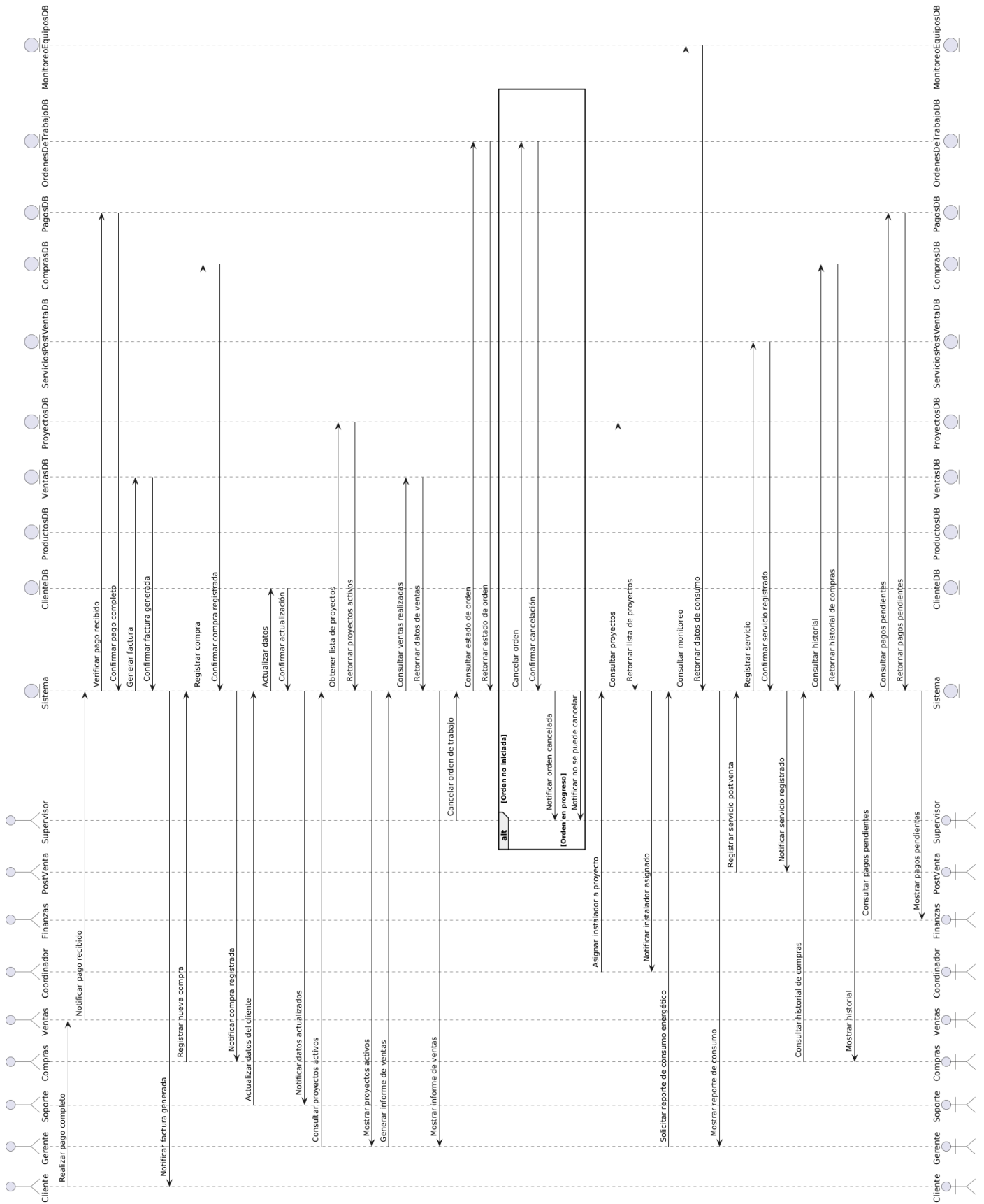


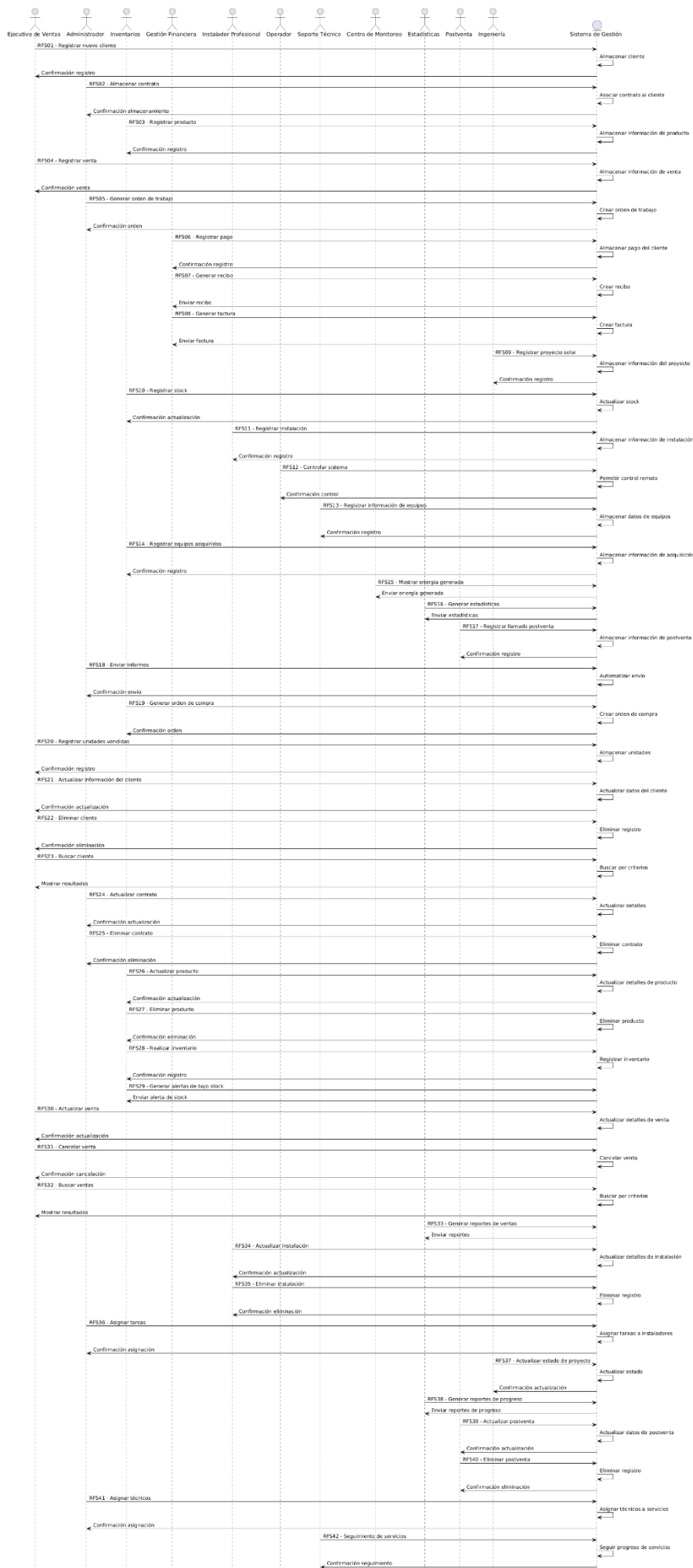
CU028: CONSULTAR PROVEEDORES:



Mas diagramas en Github.....

DIAGRAMA DE SECUENCIA COMPLETO:

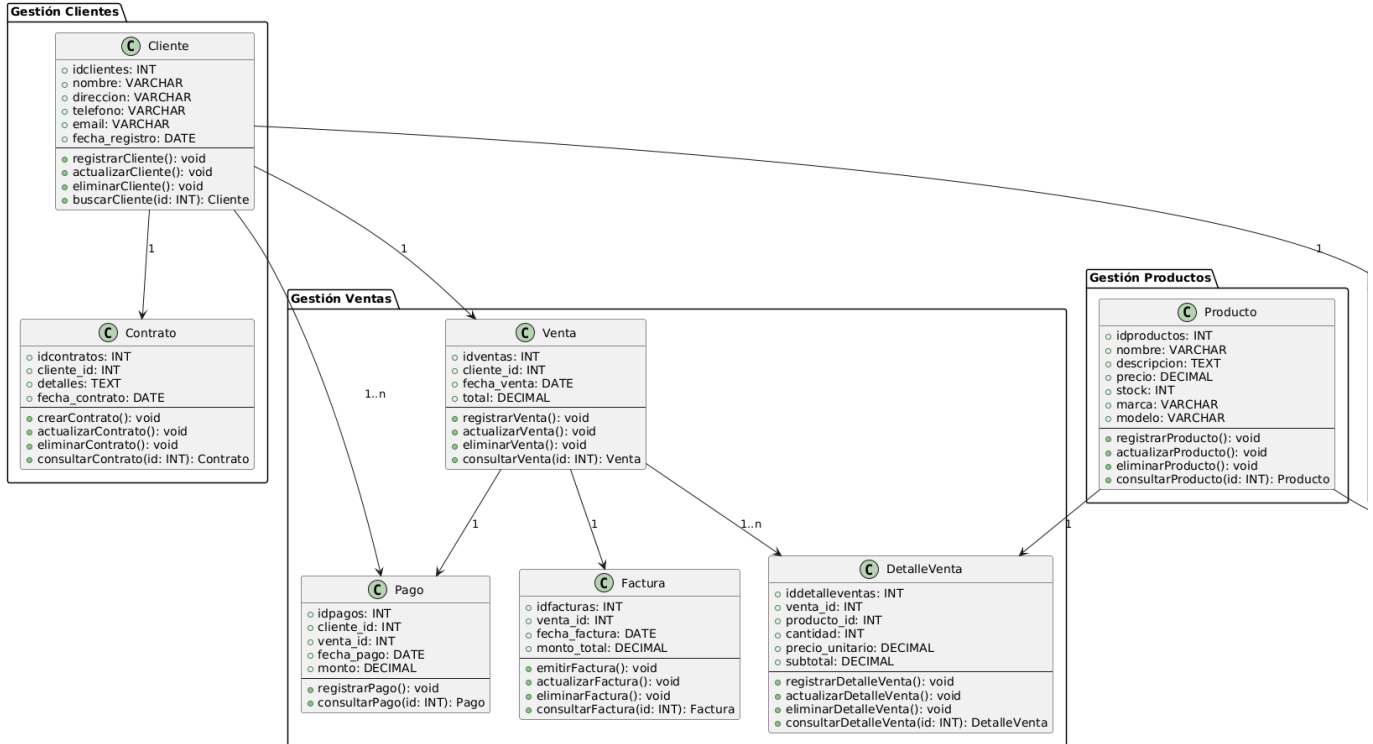




Etapa de diseño:

Diagrama de Clases y paquetes

PRIMER DIAGRAMA



SEGUNDO DIAGRAMA

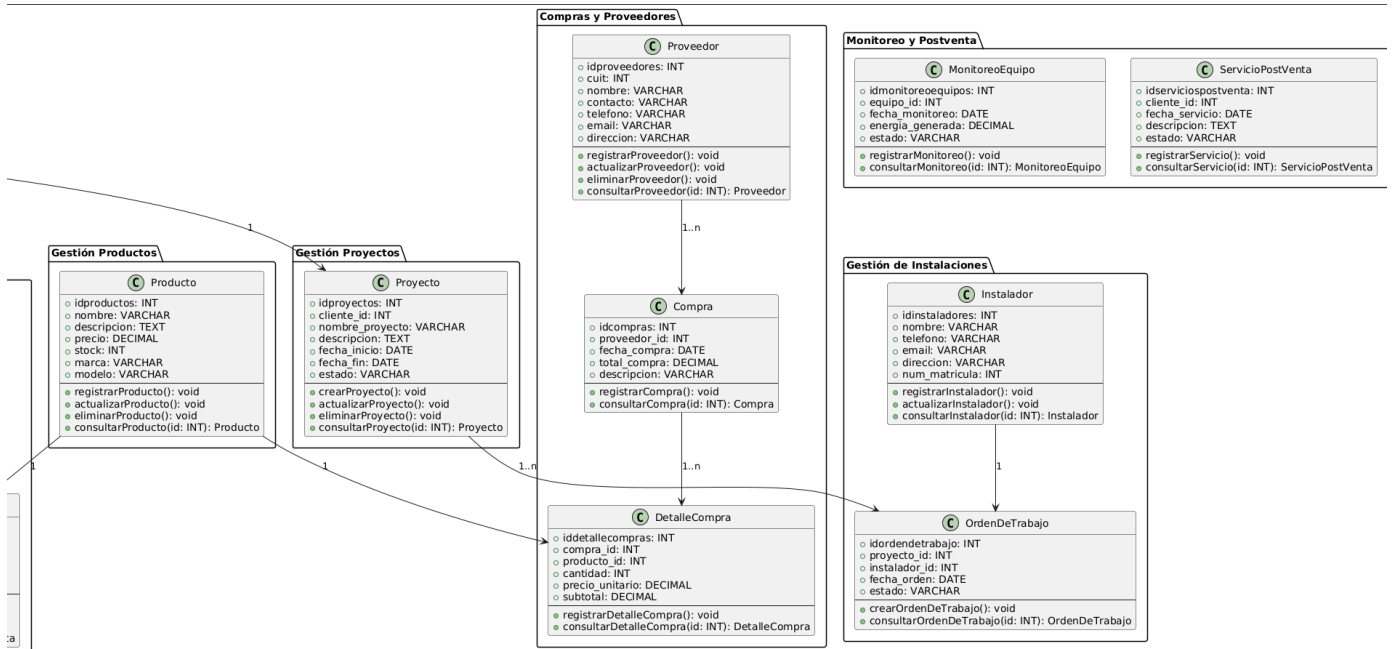
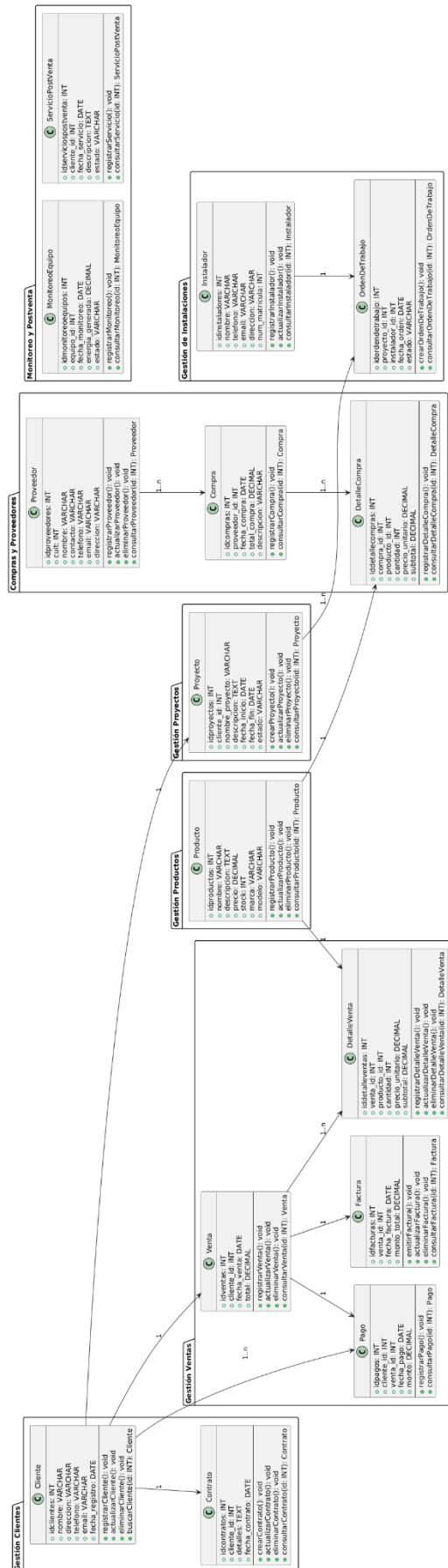


DIAGRAMA COMPLETO:



Etapa de implementación:

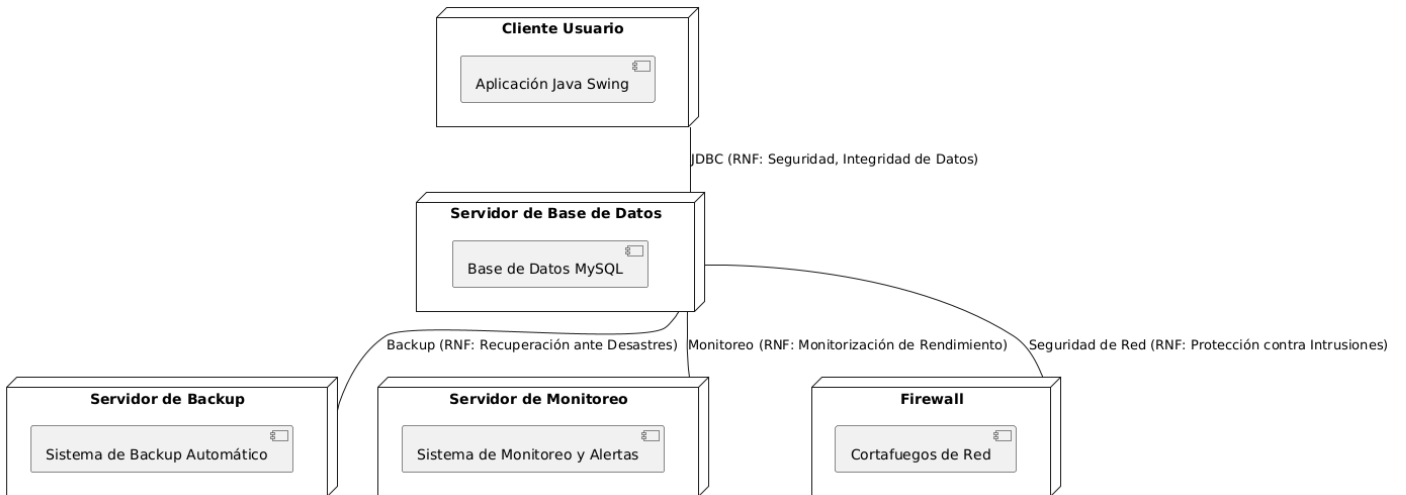
Requerimientos No funcionales a utilizar en la etapa de implementación.

La misma esta conformada por 10 RNF:

REQUERIMIENTO N°	TIPO	DESCRIPCION
RNF01	DISPONIBILIDAD	El sistema debe estar disponible el 99,99% del tiempo con un tiempo de inactividad máximo de 1 (una) hora al mes para mantenimiento
RNF02	ESCALABILIDAD	El sistema debe ser capaz de manejar un crecimiento del 300% en la cantidad de transacciones y usuarios durante los próximos tres años sin degradación significativa en el rendimiento.
RNF03	SEGURIDAD	El sistema debe implementar mecanismos de autenticación de dos factores (2FA) para usuarios administrativos y cifrado de datos sensibles como contraseñas y transacciones.
RNF04	RENDIMIENTO	Las transacciones de inserción, actualización y consulta de datos no debe exceder los 2 segundos bajo cargas de trabajo típicas de hasta 500 usuarios recurrentes.
RNF05	MANTENIBILIDAD	El sistema debe permitir actualizaciones y parches sin la necesidad de interrumpir el servicio. Se debe poder realizar una actualización completa en un tiempo máximo de 30 minutos.
RNF06	COMPATIBILIDAD	El sistema debe ser compatible con sistema operativo Windows 10, de 32 o 64 bits. Con un minimo de 4 GB de memoria RAM.
RNF07	PORTABILIDAD	El sistema debe poder ser migrado a otro entorno de servidor con una arquitectura similiar (LINUX o WINDOWS SERVER) sin cambios significativos en el código.
RNF08	FACILIDAD DE USO	El sistema debe cumplir con las pautas de accesibilidad WCAG 2.1 nivel AA para garantizar que sea ACCESIBLE para personas con DISCAPACIDADES.
RNF09	EFICIENCIA DE ALMACENAMIENTO	El sistema debe mantener una base de datos optimizada, con un uso de espacio que no exceda el 80% de la capacidad disponible en el servidor, aplicando políticas de compresión y purga de datos históricos cuando fuese necesario.
RNF10	TOLERANCIA A FALLOS	El sistema debe ser capaz de seguir funcionando con un rendimiento reducido en caso de fallos en uno o mas componentes del sistema, utilizando mecanismos de redundancia y conmutación por error (FAILOVER) para evitar la interrupción del servicio.

DIAGRAMA DE DESPLIEGUE:

Herramienta utilizada: PlantUML.



Etapas de pruebas:

Plan de Pruebas:

Se realiza la tabla con información relacionada con las decisiones que se fueron tomando a lo largo del proceso.

Se realizarán tipos de prueba funcionales, no funcionales, utilizando técnicas de caja negra, caja blanca, pruebas de recuperación, pruebas de usabilidad y pruebas de auditoría.

CASOS DE USO (CU)	CODIGO DE PRUEBA	TIPO DE PRUEBA	TECNICA PROPUESTA	OBSERVACIONES
RFS01- REGISTRAR UN NUEVO CLIENTE	TC001	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDEN REGISTRAR NUEVOS CLIENTES
RFS02- ALMACENAR UN CONTRATO Y ASIGNARLO AL CLIENTE	TC002	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE LOS CONTRATOS SE ASOCIAN CORRECTAMENTE.
RFS03- REGISTRAR INFORMACION DE PRODUCTOS	TC003	FUNCIONAL	PRUEBAS DE CAJA NEGRA	VERIFICAR QUE SE PUEDE REGISTRAR INFORMACION DE PRODUCTOS.
RFS04-REGISTRAR UNA VENTA	TC004	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE REGISTRAN LAS VENTAS
RFS05- GENERAR ORDEN DE TRABAJO PARA INSTALACION	TC005	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE GENERAN ORDENES DE TRABAJO ADECUADAMENTE.
RFS06 – REGISTRAR PAGOS DE CLIENTES	TC006	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDEN REGISTRAR PAGOS DE CLIENTES.
RFS07 – GENERAR RECIBO POR CADA PAGO RECIBIDO	TC007	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE GENERAN RECIBOS CORRECTAMENTE.
RFS08 – GENERAR FACTURA AL CLIENTE	TC008	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE LAS FACTURAS SE GENERAN AL RECIBIR EL PAGO TOTAL
RFS09 – REGISTRAR PROYECTO DE ENERGIA SOLAR	TC009	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDE REGISTRAR UN NUEVO PROYECTO
RFS10 – REGISTRAR STOCK DEBITADO TRAS LA VENTA	TC010	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE EL STOCK SE ACTUALIZA TRAS UNA VENTA.
RFS11 – REGISTRAR INSTALACION REALIZADA	TC011	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE EL STOCK SE ACTUALIZA TRAS UNA VENTA.
RFS12 – CONTROLAR SISTEMA DE GENERACION DISTRIBUIDA	TC012	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDE CONTROLAR EL SISTEMA REMOTO.
RFS13 – REGISTRAR INFORMACION DE EQUIPOS Y SENSORES	TC013	FUNCIONAL	PUREBAS DE CAJA NEGRA	VERIFICAR QUE SE PUEDEN REGISTRAR DATOS DE EQUIPOS Y SENSORES.

RFS14 – REGISTRAR EQUIPOS ADQUIRIDOS A PROVEEDORES	TC014	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDAN REGISTRAR EQUIPOS COMPRADOS.
RFS15 – MOSTRAR ENERGIA TOTAL GENERADA	TC015	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE MUESTRA CORRECTAMENTE LA ENERGIA GENERADA.
RFS16 – GENERAR ESTADISTICAS DEL FUNCIONAMIENTO	TC016	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE GENERAN INFORMES DE DESEMPEÑO.
RFS17 – REGISTRAR LLAMADO DE SERVICIO POSTVENTA	TC017	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE REGISTREN LOS LLAMADOS DE SERVICIO POSTVENTA.
RFS18 – ENVIAR INFORMES AUTOMATICAMENTE	TC018	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE LOS INFORMES SE ENVIEN AUTOMATICAMENTE.
RFS19 – GENERAR ORDEN DE COMPRA AL PROVEEDOR	TC019	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE GENENERAN ORDENES DE COMPRA CORRECTAMENTE.
RFS20 – REGISTRAR UNIDADES VENDIDAS	TC020	FUNCIONAL	PRUEBAS DE CAJA NEGRA	ASEGURARSE DE QUE SE REGISTREN LAS UNIDADES VENDIDAS
RFS21 - ACTUALIZAR INFORMACIÓN DEL CLIENTE	TC021	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDE ACTUALIZAR LA INFORMACIÓN DEL CLIENTE.
RFS22 - ELIMINAR REGISTRO DE CLIENTE	TC022	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDE ELIMINAR UN CLIENTE.
RFS23 - BUSCAR CLIENTE POR VARIOS CRITERIOS	TC023	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDEN BUSCAR CLIENTES POR DIFERENTES CRITERIOS.
RFS24 - ACTUALIZAR DETALLES DE CONTRATO	TC024	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDEN ACTUALIZAR LOS DETALLES DEL CONTRATO.
RFS25 - ELIMINAR CONTRATO	TC025	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDE ELIMINAR UN CONTRATO.
RFS26 - ACTUALIZAR INFORMACIÓN DE PRODUCTOS	TC026	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDEN ACTUALIZAR DETALLES DE PRODUCTOS.
RFS27 - ELIMINAR PRODUCTO	TC027	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDE ELIMINAR UN PRODUCTO.

RFS28 - REALIZAR INVENTARIO DE PRODUCTOS	TC028	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDEN REALIZAR INVENTARIOS DE PRODUCTOS.
RFS29 - GENERAR ALERTAS DE BAJO STOCK	TC029	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE GENERAN ALERTAS PARA PRODUCTOS CON BAJO STOCK.
RFS30 - ACTUALIZAR UNA VENTA REGISTRADA	TC030	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDEN ACTUALIZAR DETALLES DE UNA VENTA.
RFS31 - CANCELAR UNA VENTA	TC031	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDE CANCELAR UNA VENTA.
RFS32 - BUSCAR VENTAS POR VARIOS CRITERIOS	TC032	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDEN BUSCAR VENTAS POR DIFERENTES CRITERIOS.
RFS33 - GENERAR REPORTES DE VENTAS	TC033	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDEN GENERAR INFORMES DE VENTAS.
RFS34 - ACTUALIZAR DETALLES DE INSTALACIÓN	TC034	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDEN ACTUALIZAR DETALLES DE INSTALACIONES.
RFS35 - ELIMINAR INSTALACIÓN REGISTRADA	TC035	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDE ELIMINAR UN REGISTRO DE INSTALACIÓN.
RFS36 - ASIGNAR TAREAS A INSTALADORES	TC036	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDEN ASIGNAR TAREAS A INSTALADORES.
RFS37 - ACTUALIZAR ESTADO DE PROYECTO	TC037	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDE ACTUALIZAR EL ESTADO DE UN PROYECTO.
RFS38 - GENERAR REPORTES DE PROGRESO DE PROYECTOS	TC038	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE GENERAN INFORMES DE

				PROGRESO DE PROYECTOS.
RFS39 - ACTUALIZAR REGISTRO DE POSTVENTA	TC039	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDEN ACTUALIZAR DATOS DE POSTVENTA.
RFS40 - ELIMINAR REGISTRO DE POSTVENTA	TC40	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDE ELIMINAR UN REGISTRO DE POSTVENTA.
RFS41 - ASIGNAR TÉCNICOS PARA SERVICIOS	TC041	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDEN ASIGNAR TÉCNICOS A SERVICIOS.
RFS42 - SEGUIMIENTO DE RESOLUCIÓN DE SERVICIOS	TC042	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDE SEGUIR LA RESOLUCIÓN DE SERVICIOS.
RFS43 - ACTUALIZAR INFORMACIÓN DE EQUIPOS	TC043	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDEN ACTUALIZAR DETALLES DE EQUIPOS Y SENSORES.
RFS44 - ELIMINAR EQUIPO O SENSOR	TC044	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDE ELIMINAR UN REGISTRO DE EQUIPO.
RFS45 - REALIZAR MANTENIMIENTO DE EQUIPOS	TC045	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PUEDE REALIZAR MANTENIMIENTO A EQUIPOS
RFS46 - GENERAR ALERTAS DE FALLOS EN EQUIPOS	TC046	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE GENERAN ALERTAS PARA FALLOS EN EQUIPOS.
RFS47 - GENERAR REPORTES FINANCIEROS	TC047	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE GENERAN INFORMES FINANCIEROS
RFS48 - GESTIONAR CUENTAS POR PAGAR Y COBRAR	TC048	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDEN GESTIONAR CUENTAS POR PAGAR Y COBRAR.

RFS49 - INTEGRAR CON SISTEMAS DE PAGO ELECTRÓNICOS	TC049	FUNCIONAL	PRUEBA DE CAJA NEGRA	VERIFICAR QUE SE PERMITE LA INTEGRACIÓN CON SISTEMAS DE PAGO.
RFS50 - ACTUALIZAR INFORMACIÓN DE PAGOS	TC050	FUNCIONAL	PRUEBA DE CAJA NEGRA	ASEGURARSE DE QUE SE PUEDEN ACTUALIZAR DETALLES DE PAGOS.

Caso de Prueba:

Evaluación de Pruebas:

Pasos del procedimiento:

Preparación del Entorno: configurar el entorno de pruebas con datos iniciales.

Ejecución de Pruebas: Para cada RFS, realizo los pasos definidos en los casos de prueba.

Evaluación de Resultados: Comparo resultados obtenidos con los resultados esperados y clasifico en diferentes estados (APROBADA, FALLIDA, PENDIENTE).

Registro los defectos.

Tablero Evaluación de Pruebas:

RFS	ESTADO	DESCRIPCION	COMENTARIOS
RFS01	APROBADA	REGISTRAR UN NUEVO CLIENTE	REGISTRO EXITOSO DEL CLIENTE
RFS02	FALLIDA	ALMACENAR UN CONTRATO Y ASIGNARLO AL CLIENTE	ERRRO EN LA ASOCIACION DEL CONTRATO.
RFS03	AROBADA	REGISTRAR INFORMACION DE PRODUCTOS	PRODUCTO REGISTRADO CORRECTAMENTE.
RFS04	APROBADA	REGISTRAR UNA VENTA	VENTA REGISTRADA CORRECTAMENTE
RFS05	APROBADA	GENERAR ORDEN DE TRABAJO PARA INSTALACION	ORDEN GENERADA SIN ERRORES
RFS06	FALLIDA	REGISTRAR PAGO DE CLIENTES	PAGO NO SE REGISTRO CORRECTAMENTE
RFS07	APROBADA	GENERAR RECIBO POR CADA PAGO RECIBIDO	RECIBO GENERADO CORRECTAMENTE.
RFS08	APROBADA	GENERAR FACTURA AL CLIENTE	FACTURA GENERADA SIN ERRORES
RFS09	PENDIENTE	REGISTRAR PROYECTO DE ENERGIA SOLAR	ESPERANDO REVISION DE DATOS.
RFS10	APROBADA	REGISTRAR STOCK DEBITADO TRAS LA VENTA	STOCK ACTUALIZADO CORRECTAMENTE.
RFS11	FALLIDA	REGISTRAR INSTALACION REALIZADA	DETALLES DE INSTALACION NO GUARDADOS

Procedimientos de Prueba:

Para verificar que las funcionalidades del sistema cumplen con los requisitos establecidos:

Para el procedimiento de prueba se utilizan los siguientes RFS:

- RFS01: Registrar un nuevo cliente
- RFS02: Almacenar un contrato y asignarlo al cliente
- RFS03: Registrar información de productos
- RFS04: Registrar una venta
- RFS05: Generar orden de trabajo para instalación
- RFS06: Registrar pagos de clientes
- RFS07: Generar recibo por cada pago recibido
- RFS08: Generar factura al cliente
- RFS09: Registrar proyecto de energía solar
- RFS10: Registrar stock debitado tras la venta
- RFS11: Registrar instalación realizada

Tratamientos de Defectos:

ID DEFECTO	DESCRIPCION	PRIORIDAD	ESTADO	RESPONSABLE	FECHA REGISTRO	FECHA ESTIMADA RESOLUCION
DEF001	ERROR EN LA ASOCIACION DEL CONTRATO	ALTA	EN PROGRESO	Ana Rotela	06/10/2024	10/10/2024
DEF002	PAGO NO SE REGISTRO CORRECTAMENTE	ALTA	EN PROGRESO	Carlos Suarez	06/10/2024	07/10/2024
DEF003	DETALLES DE INSTALACION NO GUARDADOS	MEDIA	ABIERTO	ALBERTO AGUIRRE	06/10/2024	31/12/2024

Definición de base de datos para el sistema:

El motor de base de datos para el sistema será realizado en MYSQL, versión 8.0.36, estable, pero para las pruebas se utilizará la versión: 8.0.21 Puerto 3302.

La base de datos se llama: ***misionesenergia_gestion***.

MySQL - Wikipedia, la enciclopedia libre:

(<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjo4eGnnvmlAxUQrpUCHef1LOYQFnoECBgQAw&url=https%3A%2F%2Fes.wikipedia.org%2Fwiki%2FMySQL&usg=AOvVaw2AcYhwh8LjZo-SUx7BRaVP&opi=89978449>)

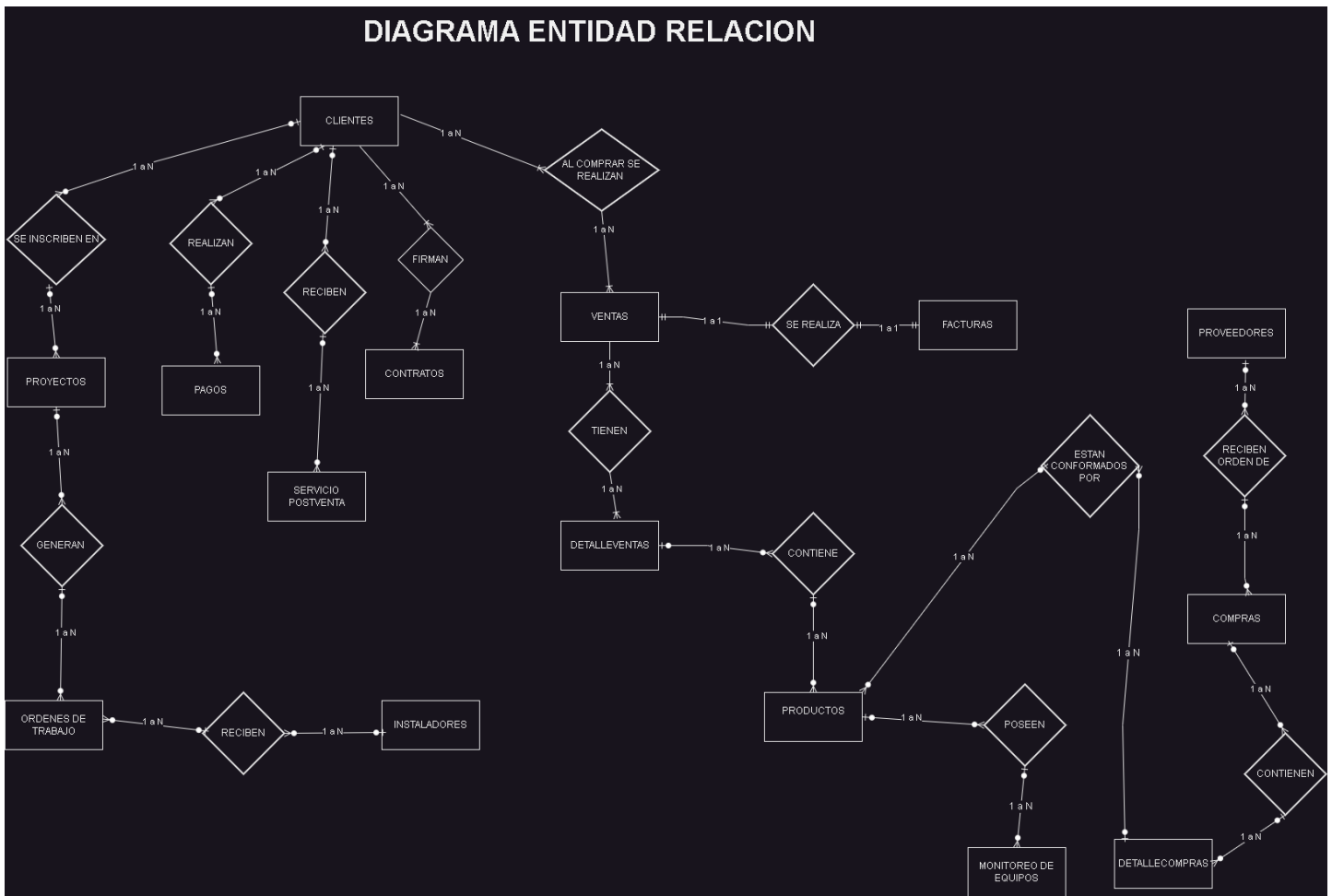
Las tablas a crear son las siguientes:

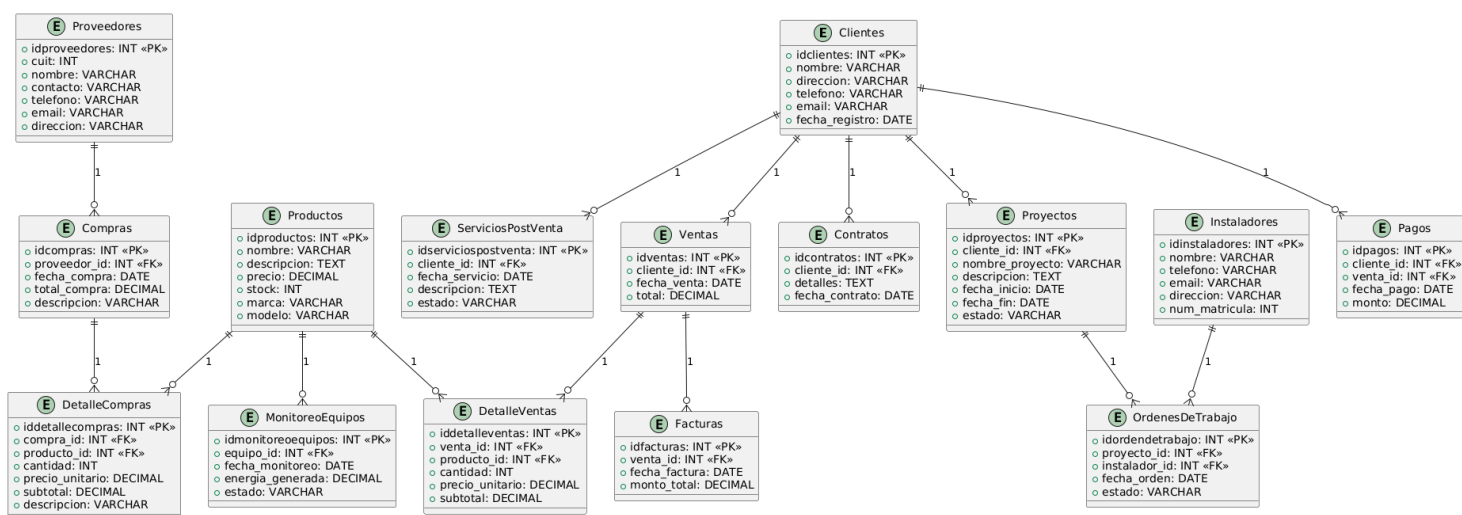
Cientes, Contratos, Productos, Ventas, DetallesVentas, Proveedores, Facturas, Pagos, Proyectos, OrdenesDeTrabajo, Instaladores, Compras, DetalleCompras, MonitoreoEquipos, ServiciosPostVenta.

Relaciones:

- Cientes tiene 1 a N Contratos
- Cientes tiene 1 a N Ventas
- Ventas tiene 1 a N DetalleVentas
- Ventas tiene 1 a 1 Facturas
- Cientes tiene 1 a N Pagos
- Cientes tiene 1 a N Proyectos
- Proyectos tiene 1 a N OrdenesDeTrabajo
- Instaladores tiene 1 a N OrdenesDeTrabajo
- Proveedores tiene 1 a N Compras
- Compras tiene 1 a N DetalleCompras
- Productos tiene 1 a N DetalleVentas
- Productos tiene 1 a N DetalleCompras
- Productos tiene 1 a N MonitoreoEquipos
- Cientes tiene 1 a N ServiciosPostVenta

Diagrama Entidad – Relación de la base de datos:





Herramienta utilizada: DRAWIO aplicación y PlantUML. Para variantes 01 y 02, Planttxt.com

SE ADJUNTA COPIA DEL DER EN GITHUB.

Archivo nombre: DIAGRAMA_ENTIDAD_RELACION.png

DIAGRAMA_SECUENCIA_COMPLETO_ETAPA_DE_ANALISIS_VARIANTE_01

Creación de las tablas:

```
/*CREACION DE LA BASE DE DATOS Y TABLAS:*/
```

```
CREATE DATABASE misionesenergia_gestion;  
USE misionesenergia_gestion;
```

```
-- Tabla Clientes
```

```
CREATE TABLE Clientes (  
    idclientes INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    direccion VARCHAR(255) NOT NULL,  
    telefono VARCHAR(20),  
    email VARCHAR(255),  
    fecha_registro DATE DEFAULT CURRENT_DATE  
);
```

```
-- Tabla Contratos
```

```
CREATE TABLE Contratos (  
    idcontratos INT AUTO_INCREMENT PRIMARY KEY,  
    cliente_id INT,  
    detalles TEXT,  
    fecha_contrato DATE DEFAULT CURRENT_DATE,  
    FOREIGN KEY (cliente_id) REFERENCES Clientes(idclientes)  
);
```

```
-- Tabla Productos
```

```
CREATE TABLE Productos (  
    idproductos INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    descripcion TEXT,  
    precio DECIMAL(10, 2) NOT NULL,  
    stock INT NOT NULL,  
    marca VARCHAR(255),  
    modelo VARCHAR(255)  
);
```

```
-- Tabla Ventas
```

```
CREATE TABLE Ventas (  
    idventas INT AUTO_INCREMENT PRIMARY KEY,  
    cliente_id INT,  
    fecha_venta DATE DEFAULT CURRENT_DATE,  
    total DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (cliente_id) REFERENCES Clientes(idclientes)  
);
```

```
-- Tabla DetalleVentas
```

```
CREATE TABLE DetalleVentas (  
    iddetalleventas INT AUTO_INCREMENT PRIMARY KEY,  
    venta_id INT,  
    producto_id INT,  
    cantidad INT NOT NULL,  
    precio_unitario DECIMAL(10, 2) NOT NULL,  
    subtotal DECIMAL(10, 2) AS (cantidad * precio_unitario),  
    FOREIGN KEY (venta_id) REFERENCES Ventas(idventas),  
    FOREIGN KEY (producto_id) REFERENCES Productos(idproductos)  
);
```

```
-- Tabla Facturas
```

```

CREATE TABLE Facturas (
    idfacturas INT AUTO_INCREMENT PRIMARY KEY,
    venta_id INT,
    fecha_factura DATE DEFAULT CURRENT_DATE,
    monto_total DECIMAL(10, 2),
    FOREIGN KEY (venta_id) REFERENCES Ventas(idventas)
);

-- Tabla Pagos
CREATE TABLE Pagos (
    idpagos INT AUTO_INCREMENT PRIMARY KEY,
    cliente_id INT,
    venta_id INT,
    fecha_pago DATE DEFAULT CURRENT_DATE,
    monto DECIMAL(10, 2),
    FOREIGN KEY (cliente_id) REFERENCES Clientes(idclientes),
    FOREIGN KEY (venta_id) REFERENCES Ventas(idventas)
);

/*CREACION TABLA MONITOREO DE EQUIPOS*/

CREATE TABLE MonitoreoEquipos (
    idmonitoreoequipos INT AUTO_INCREMENT PRIMARY KEY,
    equipo_id INT,
    fecha_monitoreo DATE DEFAULT CURRENT_DATE,
    energia_generada DECIMAL(10, 2),
    estado VARCHAR(50),
    FOREIGN KEY (equipo_id) REFERENCES Productos(idproductos)
);

/*CREACION TABLA SERVICIOSPOSTVENTA*/

CREATE TABLE ServiciosPostVenta (
    idserviciospostventa INT AUTO_INCREMENT PRIMARY KEY,
    cliente_id INT,
    fecha_servicio DATE DEFAULT CURRENT_DATE,
    descripcion TEXT,
    estado VARCHAR(50),
    FOREIGN KEY (cliente_id) REFERENCES Clientes(idclientes)
);

-- Tabla Proyectos
CREATE TABLE Proyectos (
    idproyectos INT AUTO_INCREMENT PRIMARY KEY,
    cliente_id INT,
    nombre_proyecto VARCHAR(255),
    descripcion TEXT,
    fecha_inicio DATE,
    fecha_fin DATE,
    estado VARCHAR(50),
    FOREIGN KEY (cliente_id) REFERENCES Clientes(idclientes)
);

/*CREACION TABLA COMPRAS: */

CREATE TABLE Compras (
    idcompras INT AUTO_INCREMENT PRIMARY KEY,

```

```

    proveedor_id INT,
    fecha_compra DATE DEFAULT CURRENT_DATE,
    total_compra DECIMAL(10, 2),
    descripcion VARCHAR(255),
    FOREIGN KEY (proveedor_id) REFERENCES Proveedores(idproveedores)
);

/*CREACION TABLA DETALLE COMPRAS*/

CREATE TABLE DetalleCompras (
    iddetallecompras INT AUTO_INCREMENT PRIMARY KEY,
    compra_id INT,
    producto_id INT,
    cantidad INT,
    precio_unitario DECIMAL(10, 2),
    subtotal DECIMAL(10, 2) AS (cantidad * precio_unitario),
    descripcion VARCHAR (255) NOT NULL,
    FOREIGN KEY (compra_id) REFERENCES Compras(idcompras),
    FOREIGN KEY (producto_id) REFERENCES Productos(idproductos)
);

/*CREACION DE TABLA INSTALADORES: */

CREATE TABLE Instaladores (
    idinstaladores INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255),
    telefono VARCHAR(20) NOT NULL,
    email VARCHAR(255) NOT NULL,
    direccion VARCHAR(255) NOT NULL,
    num_matricula INT(20) NOT NULL
);

/*CREACION DE TABLA PROVEEDORES: */

CREATE TABLE Proveedores (
    idproveedores INT AUTO_INCREMENT PRIMARY KEY,
    cuit INT(10) NOT NULL,
    nombre VARCHAR(255) NOT NULL,
    contacto VARCHAR(255) NOT NULL,
    telefono VARCHAR(20) NOT NULL,
    email VARCHAR(255) NOT NULL,
    direccion VARCHAR(255) NOT NULL
);

/*CREACION TABLA PROYECTOS: */

CREATE TABLE Proyectos (
    idproyectos INT AUTO_INCREMENT PRIMARY KEY,
    cliente_id INT,
    nombre_proyecto VARCHAR(255),
    descripcion TEXT,
    fecha_inicio DATE,
    fecha_fin DATE,
    estado VARCHAR(50),
    FOREIGN KEY (cliente_id) REFERENCES Clientes(idclientes)
);

```



```
/*CREACION DE ORDENES DE TRABAJO:*/
```

```
CREATE TABLE OrdenesDeTrabajo (  
    idordendetrabajo INT AUTO_INCREMENT PRIMARY KEY,  
    proyecto_id INT,  
    instalador_id INT,  
    fecha_orden DATE DEFAULT CURRENT_DATE,  
    estado VARCHAR(50),  
    FOREIGN KEY (proyecto_id) REFERENCES Proyectos(idproyectos)  
);
```

Inserción, consultas, y borrado de registros:

INSERCIÓN:

EJEMPLO TABLA CLIENTES:

```
/*INSERCIÓN DE REGISTROS EN LA TABLA CLIENTES: */
```

```
-- Clientes
```

```
INSERT INTO Clientes (nombre, direccion, telefono, email)  
VALUES  
(  
    'Juan Pérez', 'Av. Libertador 1234', '3412345678',  
    'juanperez@gmail.com'),  
(  
    'María López', 'Calle Falsa 456', '3512345678',  
    'marialopez@hotmail.com'),  
(  
    'Carlos González', 'Av. San Martín 789', '3812345678',  
    'carlosg@hotmail.com'),  
(  
    'Ana García', 'Ruta 12 Km 100', '3415671234',  
    'anagarcia@hotmail.com'),  
(  
    'Pedro Fernández', 'Calle Córdoba 1500', '2612349876',  
    'pedrofernandez@hotmail.com'),  
(  
    'Laura Sánchez', 'Boulevard Oroño 2500', '3417896543',  
    'laurasanchez@hotmail.com'),  
(  
    'Luis Ramírez', 'Pasaje Colon 123', '3515554321',  
    'luisramirez@gmail.com'),  
(  
    'Patricia Mendoza', 'Av. Corrientes 850', '3819876543',  
    'patriciamendoza@outlook.com'),  
(  
    'Diego Castro', 'Calle Entre Ríos 678', '3418881234',  
    'diegocastro@gmail.com'),  
(  
    'Sofía Herrera', 'Ruta 9 Km 20', '3517779876',  
    'sofiaher@outlook.com'),  
(  
    'Jorge Vega', 'Calle San Juan 543', '3816549876',  
    'jorgevega@outlook.com'),  
(  
    'Valeria Morales', 'Av. Belgrano 123', '3415556667',  
    'valeriamorales@electroradio.com'),  
(  
    'Alberto Ríos', 'Calle Mendoza 789', '3516549876',  
    'albertorios@hormisa.com'),  
(  
    'Rosa Torres', 'Ruta Nacional 40', '3817896543',  
    'rosatorres@cocacola.com.ar'),  
(  
    'Manuel Díaz', 'Av. Colón 2500', '3419876543',  
    'manueldiaz@outlook.com');
```

EJEMPLO TABLA CONTRATOS:

```
-- Contratos asociados a los clientes
INSERT INTO Contratos (cliente_id, detalles)
VALUES
(1, 'Contrato de instalación de sistema solar de 3kW'),
(2, 'Contrato para sistema fotovoltaico residencial de 5kW'),
(3, 'Contrato para suministro y montaje de paneles solares de 10kW'),
(4, 'Contrato de energía solar para instalación comercial de 20kW'),
(5, 'Contrato de instalación para finca agrícola'),
(6, 'Contrato para generación distribuida urbana 5kW'),
(7, 'Contrato de paneles solares para instalación industrial de 50kW'),
(8, 'Contrato para generación distribuida rural'),
(9, 'Contrato de instalación de paneles solares en hospital'),
(10, 'Contrato de proyecto solar con almacenamiento de energía'),
(11, 'Contrato de instalación de paneles solares en escuela pública'),
(12, 'Contrato de instalación fotovoltaica para residencia privada'),
(13, 'Contrato para proyecto solar comercial'),
(14, 'Contrato de instalación de sistemas solares en comunidad rural'),
(15, 'Contrato de proyecto de generación distribuida en fábrica industrial');
```

EJEMPLO TABLA PROVEEDORES:

```
/*INSERCIÓN DE REGISTROS TABLA PROVEEDORES:*/
-- Proveedores
INSERT INTO Proveedores (cuit, nombre, contacto, telefono, email, direccion)
VALUES
(20451234567, 'Energía Solar SRL', 'Laura Pérez', '3412345678', 'lperez@energiasolar.com', 'Av. Rivadavia 4321'),
(20345678901, 'Tecnología Verde SA', 'José García', '3512349876', 'jgarcia@tecverde.com', 'Calle San Martín 987'),
(20341234567, 'EcoPower Ltda', 'Ana Rodríguez', '3812345678', 'arodriguez@ecopower.com', 'Calle Corrientes 1234'),
(20459876543, 'Soluciones Energéticas SRL', 'Carlos Torres', '3415671234', 'ctorres@solenergeticas.com', 'Boulevard Belgrano 2500'),
(20452345678, 'Distribuciones Solares SA', 'María Gómez', '2612345678', 'mgomez@distsolares.com', 'Av. Libertador 987'),
(20543216789, 'Paneles y Energía SA', 'Patricia Ramírez', '3417896543', 'pramirez@panelsolar.com', 'Ruta Nacional 40'),
(20456789012, 'Energías Renovables SA', 'Juan Fernández', '3515554321', 'jfernandez@renovables.com', 'Calle Mendoza 5678'),
(20348976543, 'Instalaciones Verdes SRL', 'Pedro Martínez', '3819876543', 'pmartinez@instalverde.com', 'Calle San Juan 123'),
(20451238976, 'Generación Solar Ltda', 'Sofía Torres', '3418881234', 'storres@gensolar.com', 'Calle Córdoba 678'),
(20345671234, 'EcoInnovación SA', 'Diego Herrera', '3517779876', 'dherrera@ecoinnovacion.com', 'Ruta Nacional 9'),
(20459872345, 'Energía Limpia SRL', 'Patricia Díaz', '3816549876', 'pdiaz@energialimpia.com', 'Calle Entre Ríos 890'),
(20341239876, 'Paneles del Futuro SA', 'Marcelo Quiroga', '3415556667', 'mquiroga@panelfuturo.com', 'Calle San Luis 123'),
(20452349876, 'Energías del Sol Ltda', 'Valeria Mendoza', '3516549876', 'vmendoza@energiassol.com', 'Av. Belgrano 456'),
(20543219876, 'Soluciones Verdes SRL', 'Manuel Díaz', '3817896543', 'mdiaz@solucverdes.com', 'Calle Tucumán 789');
```

```
(20459872349, 'Paneles y Equipos SRL', 'Alberto Ríos', '3419876543',  
'arios@panelsolar.com', 'Calle Belgrano 2500');
```

CONSULTAS:

```
-- CONSULTA DE CLIENTES:  
SELECT * FROM Clientes;
```

```
-- CONSULTA DE CONTRATOS POR CLIENTES:  
SELECT * FROM Contratos WHERE cliente_id = 1;
```

```
-- CONSULTA DE VENTAS Y SUS TOTALES:  
SELECT v.idventas, c.nombre, v.fecha_venta, v.total FROM Ventas v  
JOIN Clientes c ON v.cliente_id = c.idclientes;
```

```
-- CONSULTA DE DETALLES DE VENTAS POR VENTA:  
  
SELECT dv.*, p.nombre FROM DetalleVentas dv  
JOIN Productos p ON dv.producto_id = p.idproductos  
WHERE dv.venta_id = 1;
```

```
-- CONSULTA DE FACTURAS:  
SELECT * FROM Facturas;
```

BORRADO DE REGISTROS:

```
/*BORRADO DE REGISTROS*/
```

```
-- BORRADO DE UN CLIENTE:  
DELETE FROM Clientes WHERE idclientes = 5;
```

```
-- BORRADO DE UN CONTRATO:  
DELETE FROM Contratos WHERE idcontratos = 3;
```

```
-- BORRADO DE UNA VENTA:  
DELETE FROM Ventas WHERE idventas = 2;
```

```
-- BORRADO DE UN DETALLE DE VENTA:  
DELETE FROM DetalleVentas WHERE iddetalleventas = 1;
```

```
-- BORRADO DE UNA FACTURA:  
  
DELETE FROM Facturas WHERE idfacturas = 4;
```

```
-- BORRADO DE UN PAGO:  
DELETE FROM Pagos WHERE idpagos = 3;
```

Definiciones de comunicación:

Requerimientos de Comunicación del Sistema

Interacción entre Componentes del Sistema:

Cliente-Servidor: Los clientes (usuarios) interactuarán con el servidor a través de una interfaz gráfica, donde las solicitudes se enviarán a un servidor de aplicaciones desarrollado en Java.

Base de Datos: El servidor se comunicará con la base de datos MySQL para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los datos.

Interacciones del Sistema:

JDBC (Java Database Connectivity): Para la conexión entre el servidor Java y la base de datos MySQL, se utilizará JDBC, que permite ejecutar consultas y recuperar datos.

Entorno de Red e Infraestructura Física

Entorno de Red:

Firewall: Se debe configurar un firewall para permitir solo el tráfico necesario, bloqueando accesos no autorizados.

VPN (opcional): Si se requiere acceso remoto, se puede implementar una VPN para asegurar las conexiones remotas.

Infraestructura Física:

Servidor Windows: Configurar el servidor Windows para ejecutar la aplicación Java y el servidor de base de datos MySQL.

Base de Datos: La base de datos MySQL debe estar instalada y configurada en el mismo servidor o en un servidor separado, con copias de seguridad automáticas programadas.

Control de Enlace de Datos:

Ethernet: Para la comunicación dentro de la red local.

Velocidad: Gigabyte. (1GB).

TERCERA ENTREGA:

Entregables:

- Explicación del desarrollo en Java.
- Presentación del desarrollo en Java.
- El programa compila y se ejecuta correctamente.

Contemplar:

- Correcta utilización de sintaxis, tipos de datos, estructuras de control.
- Tratamiento y manejo de excepciones.
- Adecuada aplicación del encapsulamiento, herencia, polimorfismo y abstracción.
- Disponibilidad de un menú de selección.
- Empleo de estructuras condicionales y repetitivas.
- Declaración y creación de objetos en Java.
- Utilización de constructores para inicializar objetos.
- Uso de algoritmos de ordenación y búsqueda (opcional según el desarrollo).

Explicación desarrollo en JAVA:

Se realiza un CRUD utilizando JAVA y la librería para su interfaz grafica JSWING.

Clases: InstalacionSolarApp.java

Clase principal que ejecuta el programa y muestra el menú.

- ProductoForm - Para gestionar los productos (como agregar, actualizar y listar productos).
- VentaForm - Para registrar y consultar ventas.
- ClienteForm - Para agregar y gestionar clientes.
- ContratoForm - Para gestionar contratos con clientes.
- ProyectoForm - Para la gestión de proyectos relacionados.
- OrdenTrabajoForm - Para gestionar las órdenes de trabajo.
- MonitoreoEquipoForm - Para registrar y consultar el monitoreo de equipos.

Las funcionalidades son : Crear la ventana principal de la aplicación. Manejar la navegación entre diferentes formularios (ventas, productos, clientes, ordentrabajo, monitoreoequipo.). Inicializar la conexión a la base de datos y otros recursos necesarios.

Clase VentaForm.java

Clase que representa la interfaz gráfica para la gestión de ventas. Permite ingresar y mostrar datos relacionados con las ventas.

Las funcionalidades son:

Formulario para capturar datos de una venta (cliente, fecha, total, etc.). Botones para agregar, editar y eliminar ventas. Listar todas las ventas en un componente de tabla. Manejar eventos de usuario para interactuar con los datos de ventas.

Clase DetalleVentaDAO:

Clase que gestiona las operaciones de acceso a datos para la tabla de detalles de ventas.

Las funcionalidades son: Métodos para agregar, actualizar, eliminar y consultar detalles de ventas en la base de datos. Manejo de conexiones a la base de datos y ejecución de consultas SQL.

Clase VentaDAO:

Clase que gestiona las operaciones de acceso a datos para la tabla de ventas.

Las funcionalidades son:

Métodos para agregar, actualizar, eliminar y consultar ventas en la base de datos. Implementar la lógica para manejar las relaciones entre ventas y sus detalles.

Clase DetalleVenta:

Clase que representa un detalle de una venta. Contiene atributos que reflejan las columnas de la tabla de detalles de ventas.

Atributos:

- a) idDetalle: ID único del detalle de la venta.
- b) ventald: ID de la venta a la que pertenece este detalle.
- c) productold: ID del producto vendido.
- d) cantidad: Cantidad del producto vendido.
- e) precioUnitario: Precio por unidad del producto.
- f) subtotal: Total calculado para este detalle.

Clase Venta:

Clase que representa una venta. Contiene atributos que reflejan las columnas de la tabla de ventas.

Atributos:

- a) idVenta: ID único de la venta.
- b) clienteld: ID del cliente que realizó la compra.
- c) fechaVenta: Fecha en que se realizó la venta.
- d) total: Monto total de la venta.
- e) detalles: Lista de objetos DetalleVenta asociados con la venta.

Clase ProductoForm:

Clase que representa la interfaz gráfica para la gestión de productos. Permite ingresar y mostrar datos relacionados con los productos.

Las funcionalidades son:

Formulario para capturar datos de un producto (nombre, descripción, precio, stock, etc.). Botones para agregar, editar y eliminar productos. Listar todos los productos en un componente de tabla. Manejar eventos de usuario para interactuar con los datos de productos.

Clase ProductoDAO:

Clase que gestiona las operaciones de acceso a datos para la tabla de productos.

Las funcionalidades son:

Métodos para agregar, actualizar, eliminar y consultar productos en la base de datos. Manejo de conexiones a la base de datos y ejecución de consultas SQL

Clase Producto:

Clase que representa un producto. Contiene atributos que reflejan las columnas de la tabla de productos.

Atributos:

- idProducto: ID único del producto.
- nombre: Nombre del producto.
- descripcion: Descripción del producto.
- precio: Precio del producto.
- stock: Cantidad disponible en stock.
- marca: Marca del producto.
- modelo: Modelo del producto.

Clase ClienteForm:

Clase que representa la interfaz gráfica para la gestión de clientes. Permite ingresar y mostrar datos relacionados con los clientes. Las funcionalidades son: Formulario para capturar datos de un cliente (nombre, dirección, teléfono, etc.). Botones para agregar, editar y eliminar clientes. Listar todos los clientes en un componente de tabla. Manejar eventos de usuario para interactuar con los datos de clientes.

Clase ClienteDAO:

Clase que gestiona las operaciones de acceso a datos para la tabla de clientes.

Las funcionalidades son:

Métodos para agregar, actualizar, eliminar y consultar clientes en la base de datos. Manejo de conexiones a la base de datos y ejecución de consultas SQL.

Clase Cliente:

Clase que representa un cliente. Contiene atributos que reflejan las columnas de la tabla de clientes.

Atributos:

- idCliente: ID único del cliente.
- nombre: Nombre del cliente.
- direccion: Dirección del cliente.
- telefono: Teléfono de contacto.
- email: Correo electrónico.
- fechaRegistro: Fecha en que se registró el cliente.

Presentación del desarrollo en JAVA:

Clase Principal:

```
//CLASE PRINCIPAL.

package seminario;

import javax.swing.*;

public class InstalacionSolarApp {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            String[] opciones = {"Clientes", "Productos", "Ventas", "Salir"};
            boolean continuar = true;

            while (continuar) {
                int opcion = JOptionPane.showOptionDialog(null, "Seleccione una opción:", "Menú Principal",
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null, opciones, opciones[0]);

                switch (opcion) {
                    case 0:
                        ClienteForm clienteForm = new ClienteForm();
                        clienteForm.setVisible(true);
                        break;
                    case 1:
                        ProductoForm productoForm = new ProductoForm();
                        productoForm.setVisible(true);
                        break;
                    case 2:
                        VentaForm ventaForm = new VentaForm();
                        ventaForm.setVisible(true);
                        break;
                    default:
                        continuar = false;
                        break;
                }
            }
        });
    }
}
```

Clase Cliente.java

```
1 package seminario;
2
3 import java.awt.EventQueue;
4
5 public class Cliente {
6     private int id;
7     private String nombre;
8     private String direccion;
9     private String telefono;
10    private String email;
11
12    public Cliente(int id, String nombre, String direccion, String telefono, String email) {
13        this.id = id;
14        this.nombre = nombre;
15        this.direccion = direccion;
16        this.telefono = telefono;
17        this.email = email;
18    }
19
20    // Getters y Setters
21    public int getId() { return id; }
22    public void setId(int id) { this.id = id; }
23    public String getNombre() { return nombre; }
24    public void setNombre(String nombre) { this.nombre = nombre; }
25    public String getDireccion() { return direccion; }
26    public void setDireccion(String direccion) { this.direccion = direccion; }
27    public String getTelefono() { return telefono; }
28    public void setTelefono(String telefono) { this.telefono = telefono; }
29    public String getEmail() { return email; }
30    public void setEmail(String email) { this.email = email; }
31 }
```

Clase Venta.java

```
//CLASE Venta.java

package seminario;

public class Venta {
    private int id;
    private int clienteId;
    private String fechaVenta;
    private double total;

    public Venta(int id, int clienteId, String fechaVenta, double total) {
        this.id = id;
        this.clienteId = clienteId;
        this.fechaVenta = fechaVenta;
        this.total = total;
    }

    // Getters y Setters
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }
    public int getClienteId() { return clienteId; }
    public void setClienteId(int clienteId) { this.clienteId = clienteId; }
    public String getFechaVenta() { return fechaVenta; }
    public void setFechaVenta(String fechaVenta) { this.fechaVenta = fechaVenta; }
    public double getTotal() { return total; }
    public void setTotal(double total) { this.total = total; }
}
```

Clase DetalleVenta.java

```
//Clase DetalleVenta.java

package seminario;

public class DetalleVenta {
    private int id;
    private int ventaId;
    private int productoId;
    private int cantidad;
    private double precioUnitario;
    private double subtotal;

    public DetalleVenta(int id, int ventaId, int productoId, int cantidad, double precioUnitario) {
        this.id = id;
        this.ventaId = ventaId;
        this.productoId = productoId;
        this.cantidad = cantidad;
        this.precioUnitario = precioUnitario;
        this.subtotal = cantidad * precioUnitario;
    }

    // Getters y Setters
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }
    public int getVentaId() { return ventaId; }
    public void setVentaId(int ventaId) { this.ventaId = ventaId; }
    public int getProductoId() { return productoId; }
    public void setProductoId(int productoId) { this.productoId = productoId; }
    public int getCantidad() { return cantidad; }
    public void setCantidad(int cantidad) { this.cantidad = cantidad; }
    public double getPrecioUnitario() { return precioUnitario; }
    public void setPrecioUnitario(double precioUnitario) { this.precioUnitario = precioUnitario; }
    public double getSubtotal() { return subtotal; }
}
```

Clase VentaForm.java

```
//Clase VentaForm.java

package seminario;

import java.awt.EventQueue;

public class VentaForm extends JFrame {
    private JTextField txtClienteId, txtTotal;
    private JButton btnAgregarVenta, btnListarVentas;
    private VentaDAO ventaDAO = new VentaDAO();

    public VentaForm() {
        setTitle("Gestión de Ventas");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(new GridLayout(4, 2));

        JLabel lblClienteId = new JLabel("Cliente ID:");
        txtClienteId = new JTextField();
        JLabel lblTotal = new JLabel("Total:");
        txtTotal = new JTextField();

        btnAgregarVenta = new JButton("Agregar Venta");
        btnAgregarVenta.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                agregarVenta();
            }
        });

        btnListarVentas = new JButton("Listar Ventas");
        btnListarVentas.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                listarVentas();
            }
        });

        add(lblClienteId); add(txtClienteId);
        add(lblTotal); add(txtTotal);
        add(btnAgregarVenta); add(btnListarVentas);
    }

    private void agregarVenta() {
        int clienteId = Integer.parseInt(txtClienteId.getText());
        double total = Double.parseDouble(txtTotal.getText());
        Venta venta = new Venta(0, clienteId, "2024-10-27", total);
        if (ventaDAO.addVenta(venta)) {
            JOptionPane.showMessageDialog(this, "Venta agregada correctamente");
        } else {
            JOptionPane.showMessageDialog(this, "Error al agregar venta");
        }
    }
}
```

Continuación:

```
    }

    private void listarVentas() {
        List<Venta> ventas = ventaDAO.getAllVentas();
        StringBuilder listado = new StringBuilder("Ventas:\n");
        for (Venta venta : ventas) {
            listado.append("ID Venta: ").append(venta.getId())
                .append(", Cliente ID: ").append(venta.getClienteId())
                .append(", Total: ").append(venta.getTotal()).append("\n");
        }
        JOptionPane.showMessageDialog(this, listado.toString());
    }
}
```

Clase Producto.java

```
//CLASE PRODUCTO.java

package seminario;

public class Producto {
    private int id;
    private String nombre;
    private String descripcion;
    private double precio;
    private int stock;
    private String marca;
    private String modelo;

    public Producto(int id, String nombre, String descripcion, double precio, int stock, String marca, String modelo) {
        this.id = id;
        this.nombre = nombre;
        this.descripcion = descripcion;
        this.precio = precio;
        this.stock = stock;
        this.marca = marca;
        this.modelo = modelo;
    }

    // Getters y Setters
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }
    public String getNombre() { return nombre; }
    public void setNombre(String nombre) { this.nombre = nombre; }
    public String getDescripcion() { return descripcion; }
    public void setDescripcion(String descripcion) { this.descripcion = descripcion; }
    public double getPrecio() { return precio; }
    public void setPrecio(double precio) { this.precio = precio; }
    public int getStock() { return stock; }
    public void setStock(int stock) { this.stock = stock; }
    public String getMarca() { return marca; }
    public void setMarca(String marca) { this.marca = marca; }
    public String getModelo() { return modelo; }
    public void setModelo(String modelo) { this.modelo = modelo; }
}
```

Clase ProductoForm.java

```
//CLASE ProductoForm.java

package seminario;

import java.awt.EventQueue;

public class ProductoForm extends JFrame {
    private ProductoDAO productoDAO = new ProductoDAO();
    private JTextField txtNombre, txtDescripcion, txtPrecio, txtStock, txtMarca, txtModelo;

    public ProductoForm() {
        setTitle("Gestión de Productos");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(7, 2));

        // Componentes del formulario
        JLabel lblNombre = new JLabel("Nombre:");
        txtNombre = new JTextField();
        JLabel lblDescripcion = new JLabel("Descripción:");
        txtDescripcion = new JTextField();
        JLabel lblPrecio = new JLabel("Precio:");
        txtPrecio = new JTextField();
        JLabel lblStock = new JLabel("Stock:");
        txtStock = new JTextField();
        JLabel lblMarca = new JLabel("Marca:");
        txtMarca = new JTextField();
        JLabel lblModelo = new JLabel("Modelo:");
        txtModelo = new JTextField();

        JButton btnAgregar = new JButton("Agregar");
        btnAgregar.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                agregarProducto();
            }
        });

        JButton btnListar = new JButton("Listar");
        btnListar.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                listarProductos();
            }
        });

        add(lblNombre); add(txtNombre);
        add(lblDescripcion); add(txtDescripcion);
        add(lblPrecio); add(txtPrecio);
        add(lblStock); add(txtStock);
        add(lblMarca); add(txtMarca);
        add(lblModelo); add(txtModelo);
        add(btnAgregar); add(btnListar);
    }

    private void agregarProducto() {
        Producto producto = new Producto(0, txtNombre.getText(), txtDescripcion.getText(), Double.parseDouble(txtPrecio.getText()),
            Integer.parseInt(txtStock.getText()), txtMarca.getText(), txtModelo.getText());
        if (productoDAO.addProducto(producto)) {
            JOptionPane.showMessageDialog(this, "Producto agregado correctamente");
        } else {
            JOptionPane.showMessageDialog(this, "Error al agregar producto");
        }
    }

    private void listarProductos() {
        List<Producto> productos = productoDAO.getAllProductos();
        StringBuilder listado = new StringBuilder("Productos:\n");
        for (Producto producto : productos) {
            listado.append(producto.getNombre()).append(" - ").append(producto.getDescripcion()).append("\n");
        }
        JOptionPane.showMessageDialog(this, listado.toString());
    }
}
```

Clase Cliente.java

```
//CLASE CLIENTE.java

package seminario;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.List;

public class ClienteForm extends JFrame {
    private ClienteDAO clienteDAO = new ClienteDAO();
    private JTextField txtNombre, txtDireccion, txtTelefono, txtEmail;

    public ClienteForm() {
        setTitle("Gestión de Clientes");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(5, 2));

        // Componentes del formulario
        JLabel lblNombre = new JLabel("Nombre:");
        txtNombre = new JTextField();
        JLabel lblDireccion = new JLabel("Dirección:");
        txtDireccion = new JTextField();
        JLabel lblTelefono = new JLabel("Teléfono:");
        txtTelefono = new JTextField();
        JLabel lblEmail = new JLabel("Email:");
        txtEmail = new JTextField();

        JButton btnAgregar = new JButton("Agregar");
        btnAgregar.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                agregarCliente();
            }
        });

        JButton btnListar = new JButton("Listar");
        btnListar.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                listarClientes();
            }
        });

        add(lblNombre); add(txtNombre);
        add(lblDireccion); add(txtDireccion);
        add(lblTelefono); add(txtTelefono);
        add(lblEmail); add(txtEmail);
        add(btnAgregar); add(btnListar);
    }

    private void agregarCliente() {
        Cliente cliente = new Cliente(0, txtNombre.getText(), txtDireccion.getText(), txtTelefono.getText(), txtEmail.getText());
        if (clienteDAO.addCliente(cliente)) {
            JOptionPane.showMessageDialog(this, "Cliente agregado correctamente");
        } else {
            JOptionPane.showMessageDialog(this, "Error al agregar cliente");
        }
    }
}
```

```
private void listarClientes() {
    List<Cliente> clientes = clienteDAO.getAllClientes();
    StringBuilder listado = new StringBuilder("Clientes:\n");
    for (Cliente cliente : clientes) {
        listado.append(cliente.getNombre()).append(" - ").append(cliente.getDireccion()).append("\n");
    }
    JOptionPane.showMessageDialog(this, listado.toString());
}
}
```