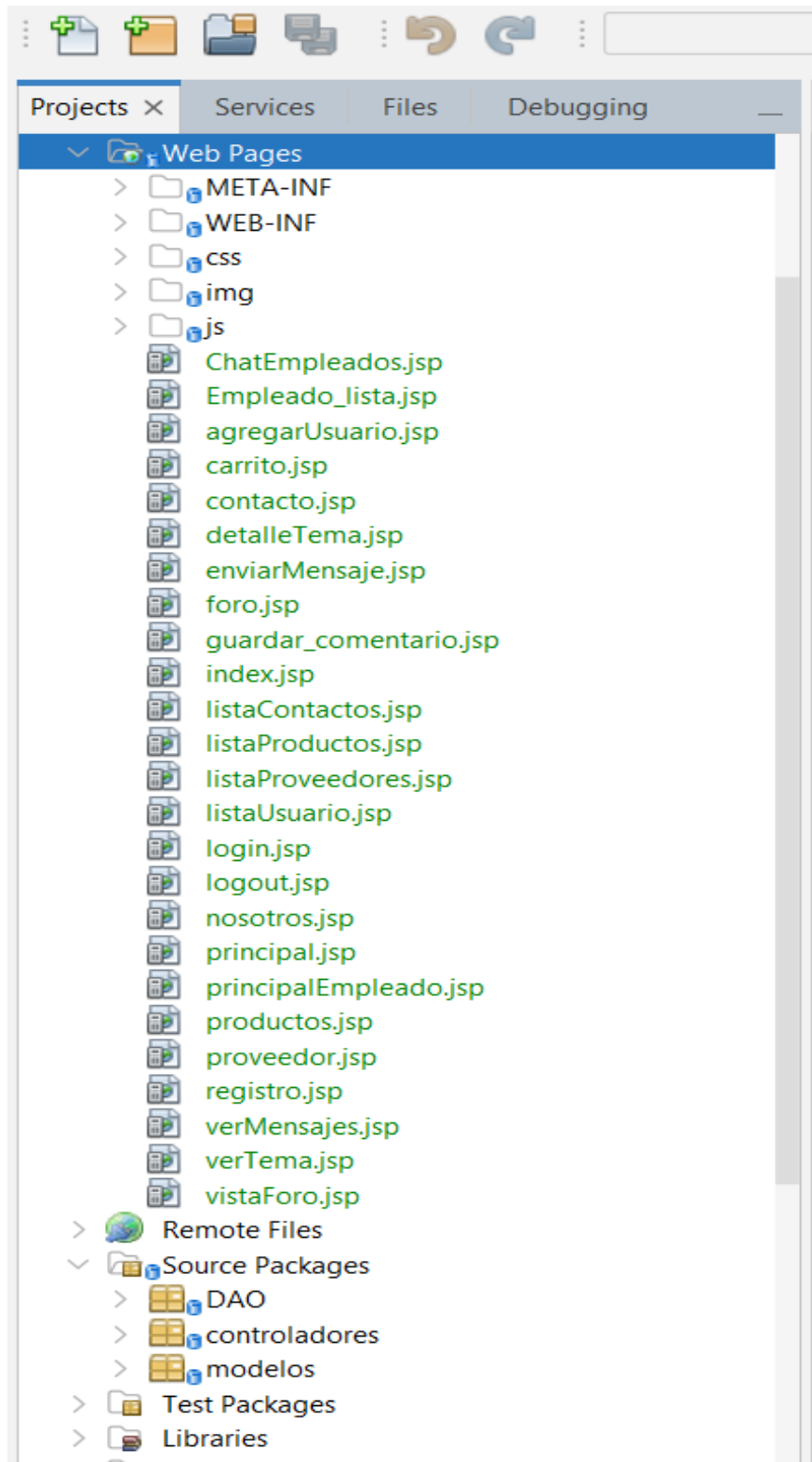


1. Introducción

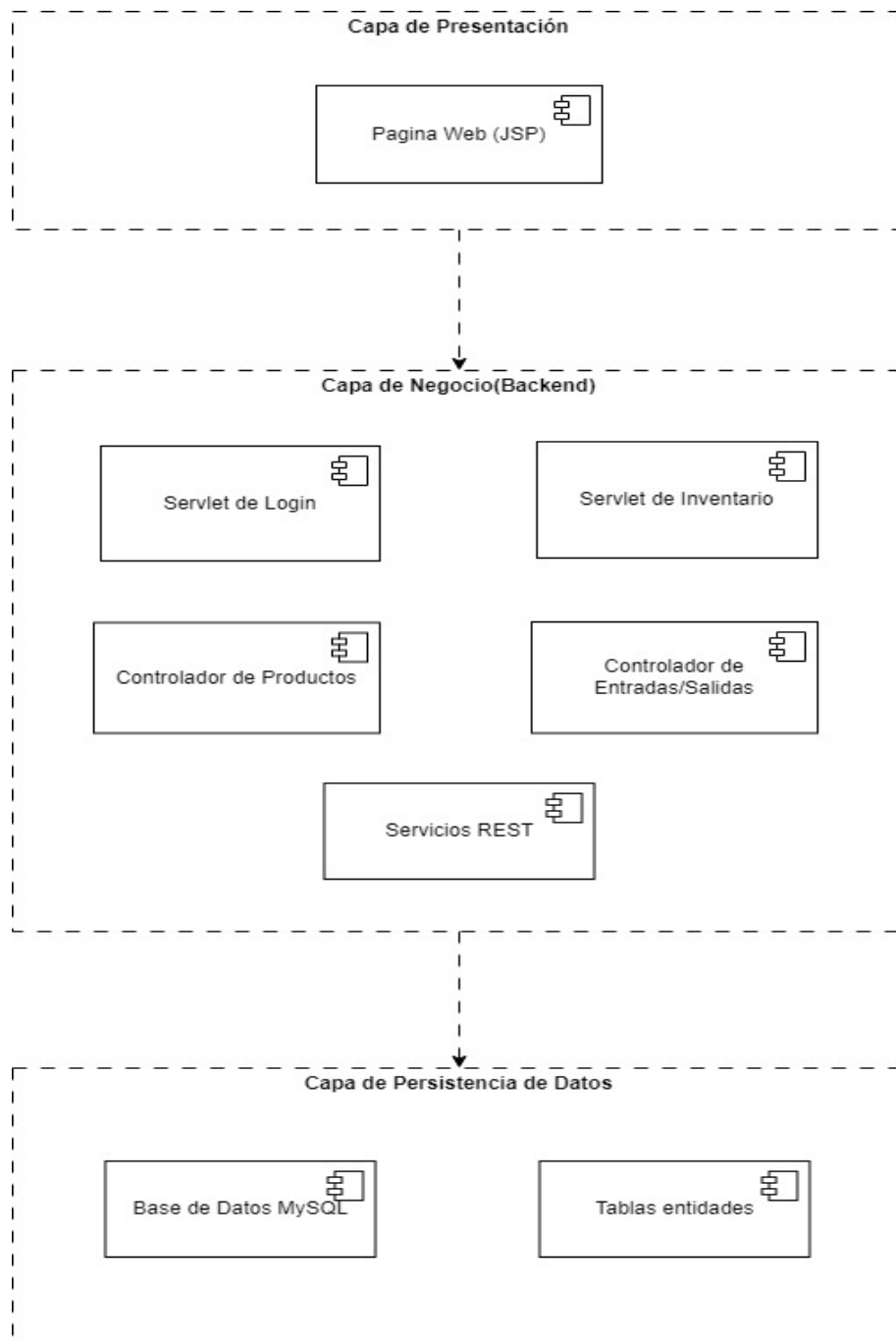
- **Propósito del Documento:** Este documento técnico proporciona una guía detallada para el desarrollo, implementación y mantenimiento de un aplicativo y página web para la gestión de inventario de productos agrícolas y ganaderos, de la Agropecuaria Campo Verde.
- **Propósito del Software:** Se establecen las funciones que hace el software (gestiona los inventarios de los productos agrícolas y ganadero), permitiendo registrar entradas y salidas de productos agrícolas y ganaderos, así como realizar un control de las existencias disponibles en la agropecuaria.
- **Audiencia:** Lista los interesados (desarrolladores, mantenedores, administradores, clientes) dirigida a ellos ya que son los que van a tener una interacción con el sistema de la empresa.

2. Arquitectura del Sistema

- **Visión General de la Arquitectura:** El sistema está estructurado con un patrón de diseño Modelo, Vista, Controlador (MVC), también usando el patrón singleton.



- **Diagrama de Arquitectura:** Incluye diagramas como diagramas de bloques, diagramas de componentes o diagramas de flujo para visualizar la arquitectura del sistema.



- **Componentes Principales:** Lista y describe detalladamente los componentes principales del sistema y cómo se relacionan entre sí.

MODELO - Base de Datos (Persistencia de Datos)

- **Descripción:** Dentro del sistema se guardan todos los datos relacionados con productos, entradas, salidas, usuarios y empleados.
- **Tecnologías Utilizadas:** Base de datos MySQL
- **Estructura de Datos:**
 - **Tabla de Productos:** Almacena información sobre cada producto (nombre, tipo, cantidad, etc.).
 - **Tabla de Empleado:** Registra la información de los empleados de la empresa (dni, nombre, teléfono, correo, cargo, área, contraseña y estado)
 - **Tabla de Categoría:** Registra las categorías de los productos (idCategoría y nombre)
 - **Tabla de Usuarios:** Almacena información de los usuarios del sistema (nombreCompleto, apellido, correo, contraseña).
 - **Tabla de Temas:** Almacena información de los temas del foro de la empresa (idTema, usuario, mensaje, fecha y hora).
 - **Tabla de Contacto:** Registra información de los usuarios del sistema (usuario, nombre, correo, teléfono, ciudad, producto, mensaje y fecha).
 - **Tabla de Inventario:** Almacena información del stock de la empresa y existencia de productos (productoID, proveedorID, tipo, cantidad, registro).
 - **Tabla de Proveedor:** Almacena información de los proveedores (nombre, nombreEmpresa, RUC, telefono).

- **Interacción:** El servidor de aplicaciones realiza consultas y actualizaciones a la base de datos registrada utilizando JDBC (Java Database Connectivity).

VISTA – Aplicación WEB

- **Descripción:** La interfaz de usuario del sistema accesible a través de un navegador web.
- **Tecnologías Utilizadas:** HTML, CSS, JavaScript, JSP (JavaServer Pages)
- **Principal funcionalidad:**
 - **Gestión de Productos:** Permite agregar, editar y eliminar productos agrícolas y ganaderos.
 - **Registro de Entradas y Salidas:** Formularios para registrar la entrada y salida de productos del inventario.
 - **Consultas y Reportes:** Páginas para consultar el inventario actual y generar reportes detallados.
- **Interacción:** Se comunica con el backend a través de solicitudes HTTP (GET, POST, PUT, DELETE) para enviar y recibir datos, dentro del sistema.

CONTROLADOR - Servidores de Aplicación / Servicios Web - API

- **Descripción:** El servidor gestiona la lógica de negocio del sistema, procesando solicitudes de la aplicación web y accediendo a toda la data de la empresa, así mismo permiten la comunicación entre la aplicación web y el servidor de aplicaciones, y potencialmente con otros sistemas externos.

- **Tecnologías Utilizadas:** Java, Servlets. RESTful APIs, JSON
- **Funcionalidades Principales:**
 - **Autenticación y Autorización:** Maneja el inicio de sesión de empleados y controla el acceso a diferentes funcionalidades según el rol del empleado.
 - **Operaciones CRUD (Crear, Leer, Actualizar, Eliminar):** Servicios para manejar las operaciones de productos, entradas, y salidas.
 - **Gestión de Inventario:** Procesa las solicitudes relacionadas con el registro de entradas y salidas de productos, y la consulta del estado del inventario.
 - **Generación de Reportes:** Crea reportes basados en los datos de inventario y los entrega a la aplicación web.
- **Interacción:** Se comunica con la vista (front-end) mediante servicios RESTful y con la base de datos para almacenar y recuperar información.

3. Requisitos del Sistema

Requisitos Funcionales:

- Registro de entradas de productos agrícolas y ganaderos.
- Registro de salidas de productos.
- Consulta de existencias disponibles.
- Generación de reportes de inventario.

Productos Registrados

ITEM	Nombres	Precio	Stock	Acción
1	Zanahoria	3.0	12	Eliminar
2	Cebada	5.0	19	Eliminar
5	Beterraga	5.0	12	Eliminar
6	Huevos	10.0	23	Eliminar

Agregar Producto

Nombre del Producto:

Categoría:

Foto:

Elegir archivo No se ha seleccionado ningún archivo

Descripción:

Precio:


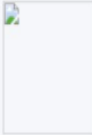


Stock:

Agregar Producto

Requisitos No Funcionales:

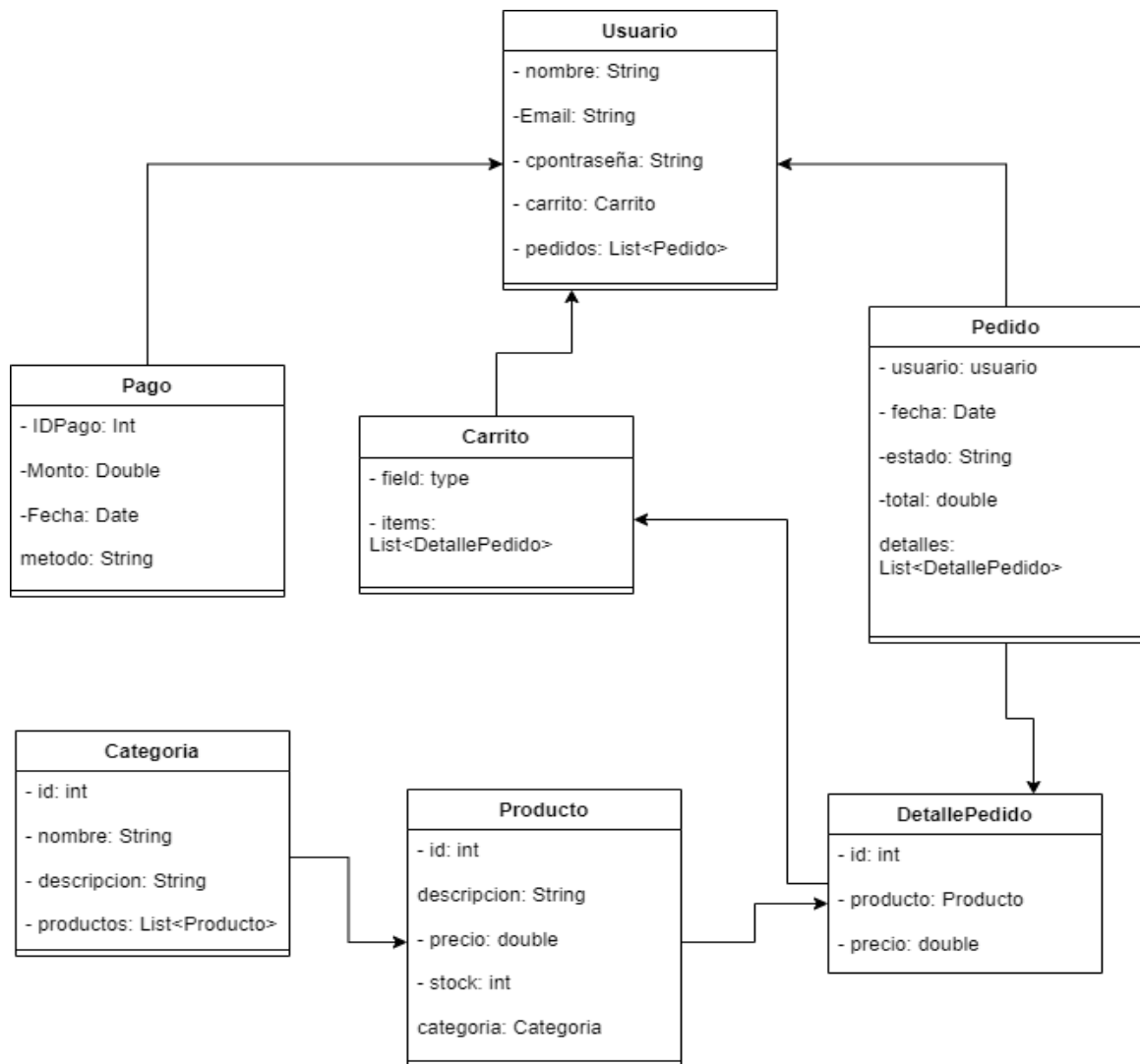
- Rendimiento: El sistema debe ser capaz de manejar grandes volúmenes de datos.
- Seguridad: Implementación de autenticación y autorización para proteger el acceso a datos sensibles.
- Usabilidad: Interfaz intuitiva para facilitar el uso por parte de los usuarios finales.

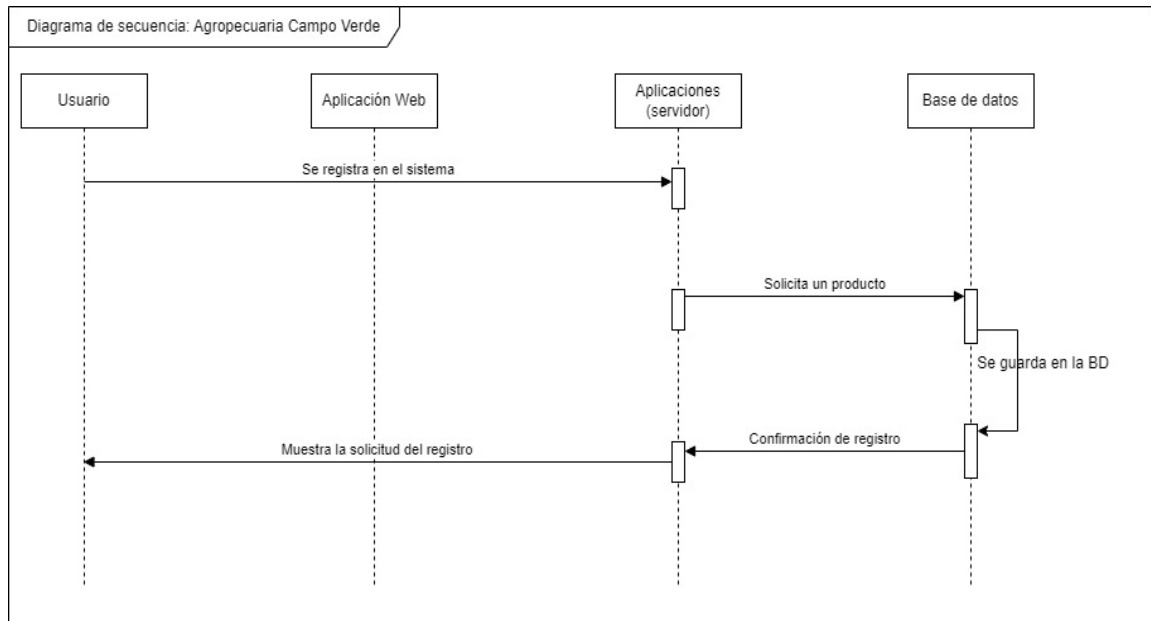
Campo Verde							
Principal							
Perfil							
Lista de clientes							
Productos							
Chat - Campo Verde							

Productos Registrados							
ITEM	Nombres	Foto	Categoría	Descripción	Precio	Stock	Acción
1	Zanahoria		1	zanahoria orange	3.0	12	Eliminar
2	Cebada		2	cebada tostada	5.0	19	Eliminar
5	Beterraga		1	De mesa	5.0	12	Eliminar
6	Huevos		2	huevos de granja	10.0	23	Eliminar

4. Diseño Detallado

- **Diagramas de Diseño:** Incluye diagramas como diagramas de clases, diagramas de secuencia u otros relevantes para ilustrar el diseño del software.

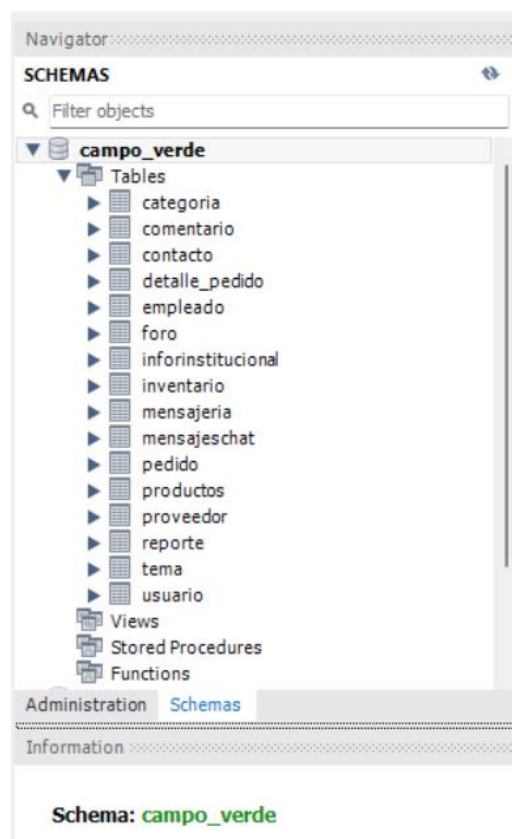




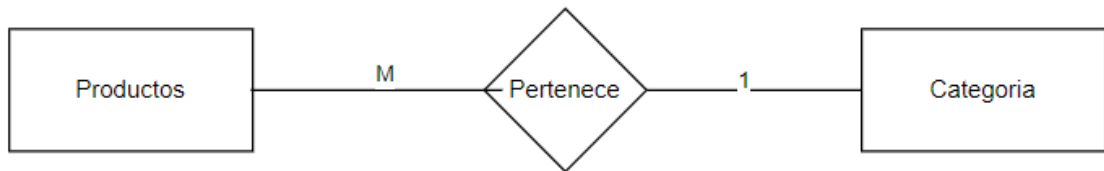
- **Decisiones de Diseño:** Las decisiones fueron trabajadas en equipo, en la cual todos los integrantes daban su opinión y se compartía en una lluvia de ideas, proponiendo diferentes interfaces y logrando llegar a un buen objetivo.

5. Modelo de Datos de la Base de Datos

- **Diagrama de Modelo de Datos:** Se presentará un diagrama ER que muestre la estructura de la base de datos.

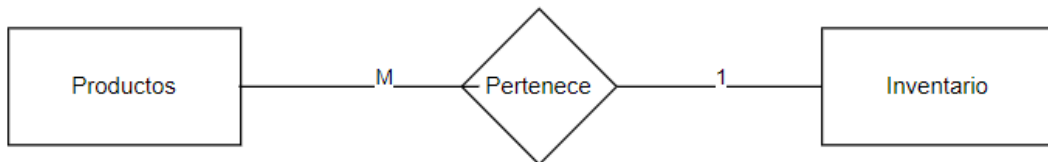


- **Descripción de Entidades y Relaciones:** Se detallan las entidades y sus atributos, así mismo las relaciones entre ellas.



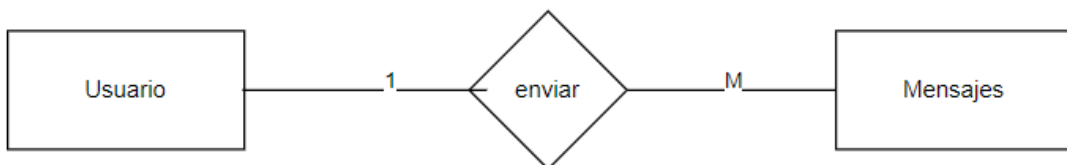
Varios productos pertenecen a una categoría.

Una categoría puede contener varios productos.



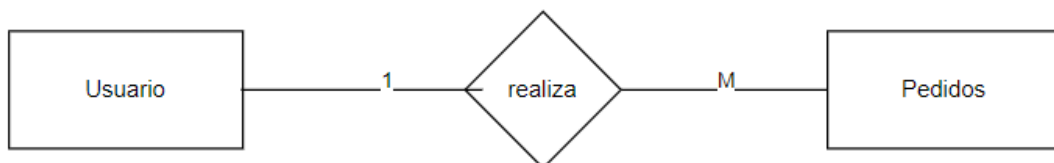
Varios productos pueden almacenarse en un inventario.

En un inventario se pueden almacenar varios productos.



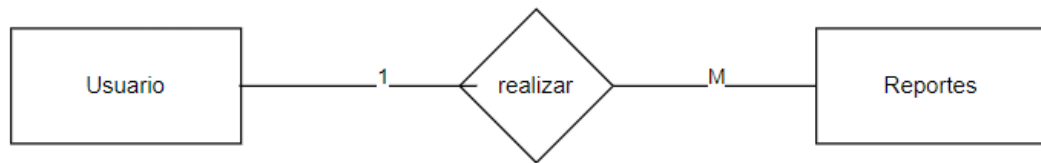
Un usuario puede enviar uno o varios mensajes.

Varios mensajes pueden ser enviados por un usuario.



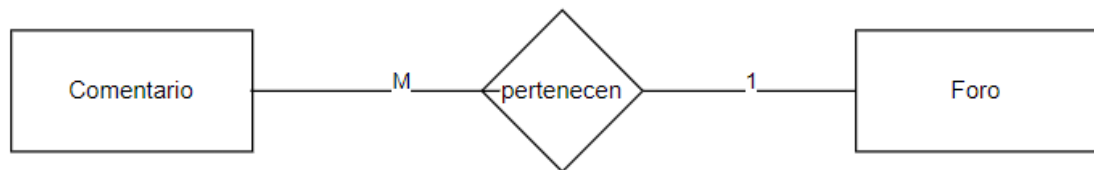
Un usuario puede realizar uno o varios pedidos.

Varios pedidos pueden ser realizados por un usuario.



Un usuario puede realizar uno o varios reportes.

Varios reportes pueden ser realizados por un usuario.



Varios comentarios pueden enviarse en un foro.

En un foro pueden enviarse varios mensajes.

Tabla de la base de datos

Tabla Producto

Campo	Tipo	Descripción
ProductoID	int (11)	Identificador de producto
Nombre	Varchar (30)	Nombre de producto
Categoría	Int (10)	Categoría de producto
Descripción	Text	Descripción del producto
Precio	Decimal (10,0)	Precio del producto
Foto	longblob	Imagen del producto
Disponibilidad	Int (11)	Stcok en almacen

Tabla Inventario

Campo	Tipo	Descripción
InventarioID	int (11)	Identificador del inventario
ProductoID	Int (11)	Identificador del producto
ProveedorID	Int (11)	Identificador del proveedor
Categoria	Int (11)	Categoría del producto
Cantidad	Int (11)	Adquirir productos
Registro	Date	Fecha de compra

Tabla proveedor

Campo	Tipo	Descripción
IDproveedor	Int (11)	Identificador del proveedor
Nombre	varchar (255)	Nombre del proveedor
NombreEmpresa	Varchar (255)	Nombre de la empresa
RUC	Varchar (255)	RUC de la empresa
Teléfono	Varchar (255)	Teléfono de la empresa

Tabla Categoría

Campo	Tipo	Descripción
IDCategoría	Int (11)	Identificador de la categoría.
Nombre	varchar (45)	Nombre de la categoría

Tabla detalle_pedido

Campo	Tipo	Descripción
idPedido_ped	Int (11)	Identificador del detalle de pedido.
IDPedido	Int(11)	Identificador del pedido
IDProducto	Int(11)	Identificador del producto
precio	Decimal(8,2)	Precio total del producto
cantidad	Int(11)	Cantidad adquirida por cada producto

Tabla Pedido

Campo	Tipo	Descripción
IDPedido	Int(11)	Identificador del pedido realizado
IDUsuario	Int(11)	Identificador del usuario
Fecha_ped	date	Fecha en que se realizó el pedido
Total	Decimal(8,2)	Precio total de la compra
estado	Varchar(60)	Estado de la compra

Tabla Empleado

Campo	Tipo	Descripción
IdEmpleado	Int(11)	Identificador del empleado
DNI	Varchar(8)	DNI del empleado
Nombres	Varchar(255)	Nombre del empleado
Telefono	Varchar(9)	Teléfono del empleado
Correo	varchar(255)	Correo del empleado
Cargo	Varchar(100)	Cargo del empleado en la empresa
User	Varchar(10)	Usuario con el que el usuario se registro

Tabla Usuario

Campo	Tipo	Descripción
UserID	Int(11)	Identificador del usuario
nombreCompleto	Varchar(30)	Nombre del usuario
apellido	Varhcar(30)	Apellido del usuario
correo	Varchar(30)	Correo del usuario
contraseña	Varchar(30)	Contraseña del usuario

Tabla Reporte

Campo	Tipo	Descripción
ReporteID	Int(11)	Indentificador del reporte
IDUsuario	Int(11)	Identificador del usuario
Tipo	Varchar(100)	Reporte del stock de producto
Contenido	Text	Contenido del reporte
FechaGenerado	date	Fecha actualizada del stock

Tabla mensajesChat

Campo	Tipo	Descripción
idMensaje	Int (11)	Identificador de mensaje
idTeam	Int(11)	Tema de conversación colectiva
Usuario	Varchar(50)	Usuario que envia el mensaje
Mensaje	Text	Mensaje que se envio
Timestamp	Timestamp	Hora y fecha de envio

Tabla mensajería

Campo	Tipo	Descripción
MensajeID	Int(11)	Identificador del chat
EmisorID	Int(11)	Empleado que envía el mensaje
ReceptorID	Int(11)	Empleado que recibe el mensaje
mensaje	Text	Contenido del mensaje
timestamp	timestamp	Hora y fecha de envío

Tabla tema

Campo	Tipo	Descripción
id	Int(11)	Identificador del tema conversado
titulo	Varchar(255)	Tema de conversación

6. Módulos o Componentes

- **Lista de Módulos o Componentes:** Enumera todos los módulos o componentes del software.
 - Módulo de Gestión de Productos
 - Módulo de Gestión de Entradas y Salidas
 - Módulo de Registro de Empleados
 - Módulo de Registro de Usuarios
 - Módulo de Foros
- **Descripción de Cada Módulo o Componente:** Detalla el propósito, funciones, interfaces, dependencias y relaciones de cada módulo con otros.

1. Módulo de Gestión de Productos

Funciones:

- Permite registrar nuevos productos en el sistema.
- Permite actualizar la información de los productos existentes.
- Facilita la eliminación de productos obsoletos o fuera de stock.

Interfaces:

- Interfaz de usuario para el registro y mantenimiento de productos.
- Conexión con la base de datos para almacenar y recuperar información de productos.

Dependencias:

- Depende del módulo de autenticación de usuarios para gestionar permisos de acceso.
- Interactúa con el módulo de gestión de entradas y salidas para actualizar el inventario según las transacciones realizadas.

2. Módulo de Gestión de Entradas y Salidas

Funciones:

- Permite registrar nuevas entradas de productos al inventario.
- Permite registrar las salidas de productos vendidos o retirados del inventario.
- Actualiza automáticamente la cantidad de productos disponibles en el inventario.

Interfaces:

- Interfaz para registrar entradas y salidas de productos.
- Integración con el módulo de gestión de productos para verificar disponibilidad y detalles de los productos.

Dependencias:

- Depende del módulo de gestión de productos para acceder a la información detallada de cada producto.
- Interactúa con el módulo de reportes de inventario para generar informes actualizados sobre las transacciones.

3. Módulo de Registro de Empleados

Funcionalidades

1. Registro de Nuevo Empleado

- Permite ingresar los datos básicos de un nuevo empleado, como nombre, apellido, correo electrónico, teléfono, cargo, área, usuario y contraseña.
- Valida la información ingresada para asegurar la integridad de los datos.

2. Validación de Datos

- Verifica que todos los campos obligatorios estén completos antes de permitir el registro.
- Implementa validaciones específicas, como formato de correo electrónico válido y longitud adecuada de contraseña.

3. Almacenamiento en la Base de Datos

- Guarda la información del empleado registrado en la base de datos, asegurando que cada empleado tenga un identificador único (ID).
- Utiliza el DAO (Data Access Object) correspondiente para interactuar con la base de datos de manera segura y eficiente.

Interfaces y Componentes

1. Interfaz de Usuario

- **Formulario de Registro:** Presenta un formulario amigable donde se ingresan los datos del empleado.

- **Validaciones en Tiempo Real:** Implementa validaciones en tiempo real para dar retroalimentación inmediata sobre errores de formato o campos faltantes.
- **Confirmación de Registro:** Muestra una confirmación visual cuando el registro se ha completado correctamente.

2. Componentes del Backend

- **Clase de Servicio (Service Layer):** Contiene la lógica de negocio para validar los datos del formulario antes de pasarlos al DAO.
- **Clase DAO:** Gestiona la interacción directa con la base de datos, ejecutando consultas SQL preparadas para insertar los datos del nuevo empleado.

Seguridad y Privacidad

1. Seguridad en Contraseñas

- Implementa políticas de contraseñas seguras, como requerimientos de longitud mínima y complejidad.
- Almacena las contraseñas utilizando técnicas de hash para proteger la información sensible.

2. Acceso Controlado

- Limita el acceso al módulo de registro de empleados a usuarios autorizados, como administradores o personal de recursos humanos.

- Utiliza sesiones y roles de usuario para gestionar permisos y asegurar que solo personal autorizado pueda realizar registros.

4 Módulo de Registro de Usuarios

Funcionalidades

1. Registro de Nuevo Usuario

- Permite a los nuevos usuarios registrarse proporcionando información como nombre completo, correo electrónico, contraseña, y cualquier otro dato necesario según tus requerimientos.
- Valida la información ingresada para asegurar la integridad de los datos y evitar errores de formato.

2. Validación de Datos

- Implementa validaciones en tiempo real para verificar que los campos obligatorios estén completos y que los datos ingresados cumplan con los requisitos especificados (por ejemplo, formato de correo electrónico válido, longitud de contraseña).

3. Almacenamiento en la Base de Datos

- Guarda la información del nuevo usuario en la base de datos utilizando un DAO (Data Access Object) para interactuar de manera segura y eficiente con la base de datos.

4. Confirmación de Registro

- Muestra un mensaje o pantalla de confirmación una vez que el registro se haya completado exitosamente.
- Proporciona instrucciones adicionales, como verificar el correo electrónico o completar un perfil adicional si es necesario.

Interfaces y Componentes

1. Interfaz de Usuario

- **Formulario de Registro:** Presenta un formulario claro y fácil de entender donde los usuarios puedan ingresar su información personal.
- **Validaciones en Tiempo Real:** Implementa validaciones que se activan a medida que los usuarios completan cada campo, proporcionando retroalimentación inmediata sobre errores o campos faltantes.
- **Mensajes de Confirmación:** Muestra mensajes visuales o de texto que indiquen el éxito del registro y los pasos siguientes.

2. Componentes del Backend

- **Clase de Servicio (Service Layer):** Contiene la lógica de negocio para validar los datos del formulario antes de enviarlos al DAO.
- **Clase DAO:** Gestiona la conexión y las consultas a la base de datos, utilizando consultas preparadas para insertar los datos del nuevo usuario de manera segura.

Seguridad y Privacidad

1. Seguridad en Contraseñas

- Implementa políticas de contraseñas seguras, como longitud mínima y complejidad de caracteres.
- Utiliza técnicas de hash para almacenar las contraseñas de forma segura en la base de datos.

2. Acceso Controlado

- Limita el acceso al módulo de registro de usuarios a usuarios específicos o roles autorizados dentro del sistema.
- Utiliza sesiones y roles de usuario para gestionar permisos y asegurar que solo usuarios autorizados puedan registrar nuevos usuarios.

Consideraciones Adicionales

1. Integración con Otros Módulos

- Asegura que la información del nuevo usuario se sincronice correctamente con otros módulos relacionados, como gestión de permisos y roles.

2. Auditoría y Registro de Actividades

- Implementa un sistema de registro de actividades para registrar cada acción realizada en el módulo de registro de usuarios.
- Facilita la auditoría y el seguimiento de cambios para garantizar la transparencia y la trazabilidad de las operaciones realizadas.

5. Módulo de Foros

Funcionalidades

1. Creación de Temas

- Permite a los usuarios registrados crear nuevos temas de discusión en el foro.
- Cada tema puede incluir un título y contenido inicial.

2. Listado de Temas

- Muestra una lista de todos los temas disponibles en el foro, ordenados por fecha de creación o relevancia.
- Permite a los usuarios filtrar o buscar temas específicos según palabras clave o categorías.

3. Visualización de Detalles de Temas

- Permite a los usuarios ver detalles completos de un tema específico, incluyendo el contenido inicial y los comentarios asociados.

4. Comentarios en Temas

- Permite a los usuarios agregar comentarios y respuestas a los temas existentes.
- Los comentarios pueden incluir texto, imágenes u otros archivos multimedia según los requisitos del proyecto.

5. Gestión de Comentarios

- Proporciona opciones para editar y eliminar comentarios, con permisos adecuados basados en roles de usuario (por ejemplo, solo el creador del comentario o un administrador puede editar o eliminar).

6. Interacción y Notificaciones

- Implementa funciones de interacción como likes o votos positivos en comentarios para fomentar la participación.
- Configura notificaciones para usuarios cuando se agreguen nuevos comentarios o respuestas a los temas que siguen.

Interfaces y Componentes

1. Interfaz de Usuario

- **Lista de Temas:** Muestra una lista visualmente atractiva de los temas disponibles, con opciones de filtro y búsqueda.
- **Detalles del Tema:** Presenta el contenido completo de un tema seleccionado, incluidos comentarios y respuestas.
- **Formulario de Creación/Edición:** Proporciona un formulario claro y accesible para crear o editar temas y comentarios.

2. Componentes del Backend

- **Clase de Servicio (Service Layer):** Contiene la lógica de negocio para validar y procesar la creación, edición y eliminación de temas y comentarios.

- **Clase DAO:** Gestiona las consultas a la base de datos para almacenar y recuperar información de temas y comentarios de manera eficiente y segura.

Seguridad y Privacidad

1. Control de Acceso

- Implementa roles y permisos para controlar quién puede crear, editar o eliminar temas y comentarios.
- Garantiza que solo usuarios autenticados puedan participar en el foro y realizar acciones relacionadas.

2. Protección de Datos Sensibles

- Asegura que la información confidencial de los usuarios, como contraseñas o información personal, esté protegida mediante técnicas de encriptación y almacenamiento seguro.

Consideraciones Adicionales

1. Moderación y Gestión de Contenido

- Implementa herramientas para la moderación de contenido, permitiendo a los administradores eliminar o ocultar comentarios inapropiados o spam.
- Facilita la gestión eficiente del contenido del foro para mantener un entorno de discusión saludable y relevante.

2. Integración con Otros Módulos

- Asegura la integración adecuada con otros componentes del sistema, como el módulo de usuarios y autenticación, para proporcionar una experiencia cohesiva para los usuarios.

7. Interfaces de Programación (APIs)

API de PayPal

Script que se debe agregar en el archivo jsp

```
<script src = "https://www.paypal.com/sdk/js?client-id=AZLmPXld2V2idGJuUv7l_ulbXjlBihFgpHGZwVDZsAkdeyCCuj2DfxqIpPI1Teoc2NnWVP_SH6VTCza7" currency="USD"></script>
```

Las siguientes lineade código nos van a permitir mostrar los botones de paypal, y al mismo tiempo darle estilos. Además agregando la variable `${totalPagar}`, permite ver el precio total que el cliente debe pagar en la ventana de paypal.

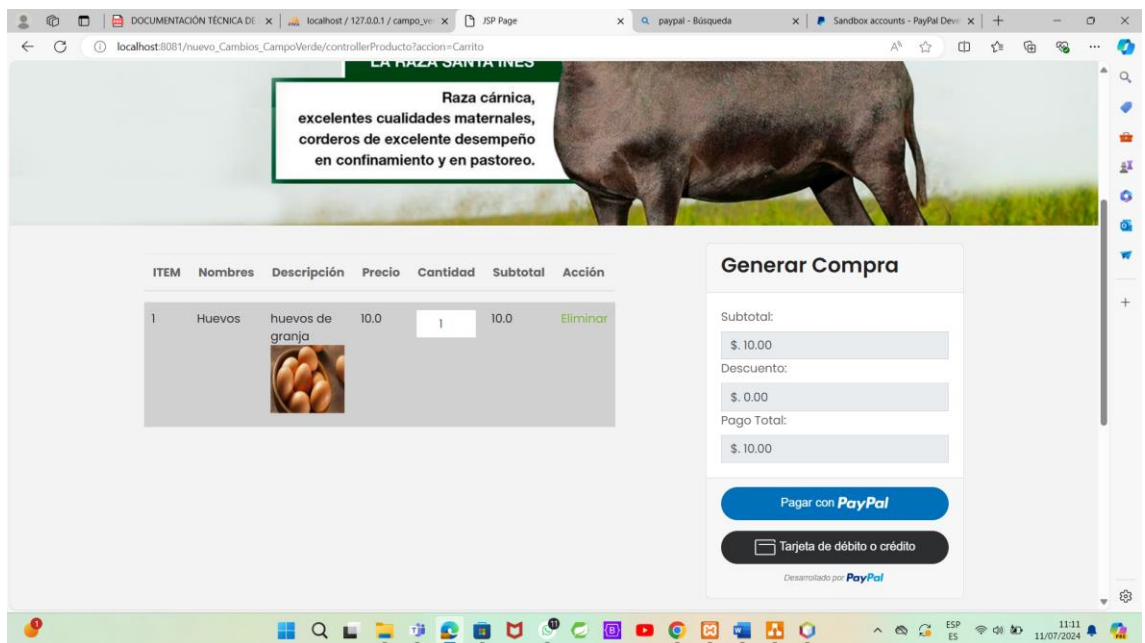
```
<div class="card-footer">
    <div id="paypal-button-container"></div>
    <script>
        var totalPagar = ${totalPagar};
        paypal.Buttons({
            style: {
                color: 'blue',
                shape: 'pill',
                label: 'pay'
            },
            createOrder: function(data, actions){
                return actions.order.create({
                    purchase_units:[{
```

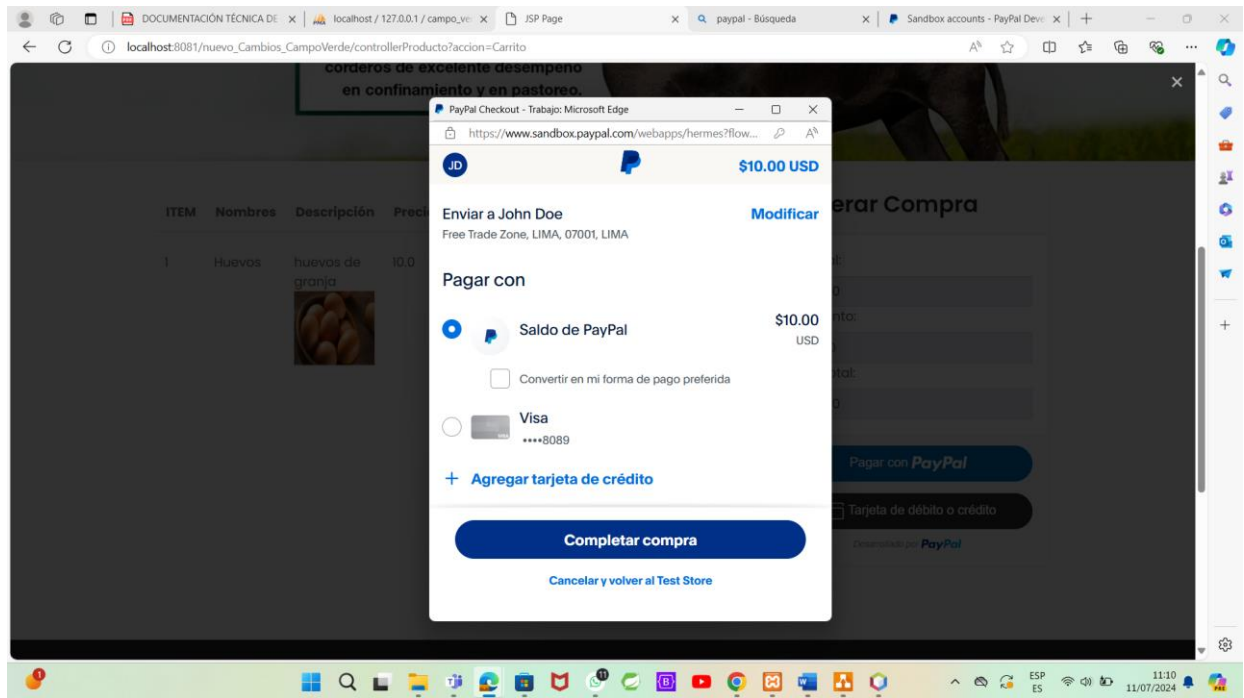
```

        amount: {
            value: totalPagar.toFixed(2)
        }
    }
}

onApprove: function(data, actions){
    actions.order.capture().then(function(detalles){
        console.log(detalles);
    });
},
onCancel: function(data){
    alert("Pago Cancelado");
    console.log(data);
}
}).render('#paypal-button-container');
</script>
</div>

```





API de Google Maps

En el archivo jsp se tiene que agregar la siguiente etiqueta:

```
<link rel="preconnect" href="https://fonts.gstatic.com">
```

```
<link
```

```
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap" rel="stylesheet">
```

```
<script
```

```
src="https://maps.googleapis.com/maps/api/js?key=TU_API_KEY"></script>
```

```
<!-- Script para inicializar el mapa -->
```

```
<script>
```

```
function initMap() {
```

```
    // Coordenadas del centro del mapa
```

```
    var myLatLng = {lat: -12.046374, lng: -77.042793};
```

```
    // Opciones del mapa
```

```
    var mapOptions = {
```

```

        zoom: 15, // Nivel de zoom

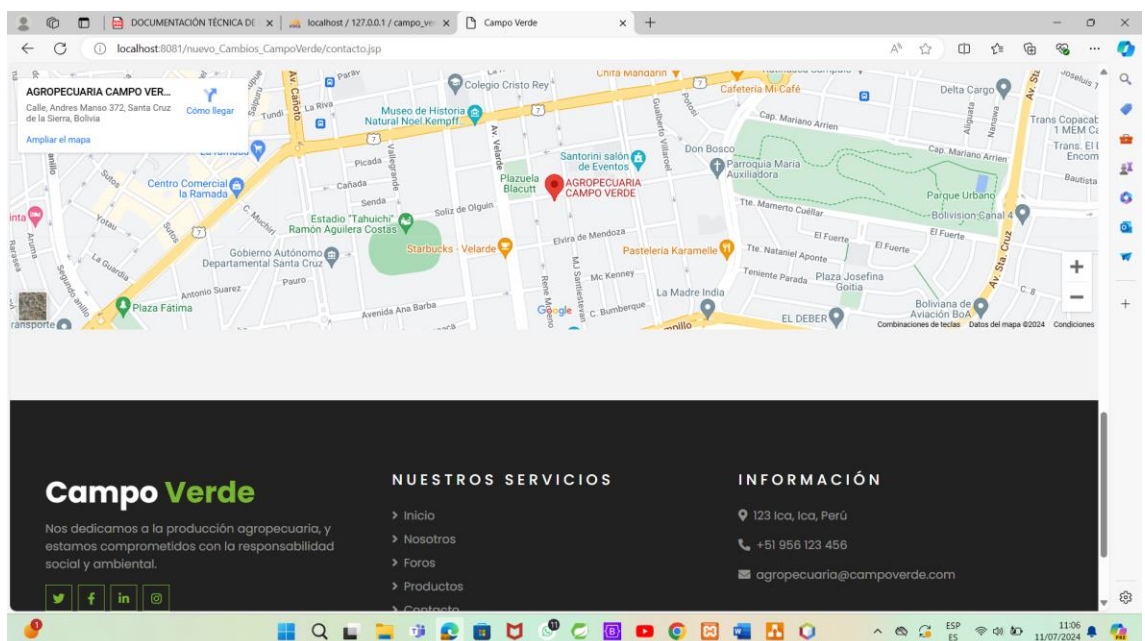
        center: myLatLng // Centro del mapa
    };

    // Crear el mapa
    var map = new google.maps.Map(document.getElementById('map'),
    mapOptions);

    // Marcador en el centro del mapa
    var marker = new google.maps.Marker({
        position: myLatLng,
        map: map,
        title: 'Ubicación de Campo Verde' // Título del marcador
    });
}

// Llamar a initMap cuando la página esté cargada
document.addEventListener('DOMContentLoaded', initMap);
</script>

```



8. Guía de Uso y Operación

Funcionalidades Principales: Guía detallada sobre cómo utilizar las funciones principales del aplicativo y la página web, incluyendo el registro de productos, gestión de entradas y salidas, y consulta de inventario.

1. Inicio de Sesión

- Accede al aplicativo utilizando tu nombre de usuario y contraseña proporcionados por el administrador.

2. Registro de Productos

- Para añadir un nuevo producto al inventario:
 - Dirígete a la sección de "Registro de Productos".
 - Completa los campos requeridos como nombre, tipo y cantidad.
 - Confirma el registro para guardar el producto en la base de datos.

3. Gestión de Entradas

- Registra las nuevas entradas de productos al inventario:
 - Selecciona la opción de "Registrar Entrada".
 - Ingresa los detalles como el producto, cantidad y fecha de entrada.
 - Guarda la entrada para actualizar la disponibilidad del producto en el inventario.

4. Gestión de Salidas

- Registra las salidas de productos del inventario:
 - Accede a la opción de "Registrar Salida".

- Indica el producto a retirar, la cantidad y la fecha de salida.
- Confirma la salida para actualizar el inventario y mantener la trazabilidad de productos.

5. Consulta de Inventario

- Revisa el estado actual del inventario en tiempo real:
 - Navega a la sección de "Consulta de Inventario".
 - Visualiza la lista de productos disponibles junto con su cantidad actual.

6. Seguridad y Acceso

- Para mantener la seguridad de los datos:
 - Cierra sesión después de cada uso.
 - Utiliza contraseñas seguras y no compartas tus credenciales de acceso.

9. Consideraciones de Seguridad

- **Consideraciones de Seguridad:** Se implementan medidas de seguridad, como la autenticación de usuarios y empleados, así mismo como el control de acceso al aplicativo por parte de los empleados, como acceso por el cargo que ellos tienen.

```
@WebServlet("/inicioSesion")
public class inicioSesion extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String correo = request.getParameter(string: "usuario");
        String contrasena = request.getParameter(string: "contrasena");

        // Validar las credenciales del usuario normal
        if (validarCredencialesUsuario(correo, contrasena)) {
            // Obtener y establecer el nombre completo del usuario en la sesión
            String nombreUsuario = obtenerNombreUsuario(correo);
            request.getSession().setAttribute(string: "usuario", o: correo);
            request.getSession().setAttribute(string: "nombreUsuario", o: nombreUsuario);
            response.sendRedirect(string: "index.jsp");
        } else {
            // Validar las credenciales del empleado
            String cargo = obtenerCargoEmpleado(usuario:correo, contrasena);
            int IdEmpleado = obtenerIdEmpleado(usuario:correo, contrasena);
            if (cargo != null) {
                // Verificar el estado del empleado
                String estadoEmpleado = obtenerEstadoEmpleado(usuario:correo, contrasena);
                if ("habilitado".equals(anObject: estadoEmpleado)) {
                    request.getSession().setAttribute(string: "empleado", o: correo);
                    String nombre = obtenerNombreEmpleado(usuario:correo, contrasena);
                    request.getSession().setAttribute(string: "IdEmpleado", o: IdEmpleado);
                    request.getSession().setAttribute(string: "nombre", o: nombre);
                    if ("Administrador".equals(anObject: cargo)) {
                        response.sendRedirect(string: "principal.jsp");
                    } else {
                        response.sendRedirect(string: "principalEmpleado.jsp");
                    }
                } else {
                    request.setAttribute(string: "error", o: "El empleado está inhabilitado para iniciar sesión.");
                    request.getRequestDispatcher(string: "login.jsp").forward(sx: request, srl:response);
                }
            } else {
                request.setAttribute(string: "error", o: "Usuario o contraseña incorrectos");
                request.getRequestDispatcher(string: "login.jsp").forward(sx: request, srl:response);
            }
        }
    }
}
```


```
private String obtenerCargoEmpleado(String usuario, String contrasena) {
    try (Connection conn = ConexionDB.getInstancia().getConnection();
        PreparedStatement stmt = conn.prepareStatement(string: "SELECT Cargo FROM empleado WHERE User = ? AND Contraseña = ?")) {

        stmt.setString(i: 1, string: usuario);
        stmt.setString(i: 2, string: contrasena);


        try (ResultSet rs = stmt.executeQuery()) {
            if (rs.next()) {
                return rs.getString(string: "Cargo");
            }
        }
    } catch (SQLException | ClassNotFoundException e) {
        e.printStackTrace();
    }



    return null;
}
```


- **Vulnerabilidades Conocidas:** Entre las vulnerabilidades que encontramos es que se tiene que trabajar con un repositorio donde este almacenado el proyecto y actualizarlo correspondientemente, para mitigar riesgos de perdida de proyecto y mitigar la seguridad en el sistema.

 Campo-Verde---WEB Public Pin Unwatch

main 1 Branch 0 Tags Add file Code

 JAlonsoSC Campo Verde ebfd516 · 7 minutes ago 1 Commit

 nuevo_Cambios_CampoVerde	Campo Verde	7 minutes ago
 campo_verde.sql	Campo Verde	7 minutes ago

10. Mantenimiento de la Documentación

- **Plan de Mantenimiento:** El plan de mantenimiento busca asegurar que la documentación del sistema de gestión de inventario de productos agrícolas y ganaderos esté siempre actualizada y refleje con precisión el estado y las funcionalidades del sistema a lo largo de todo su ciclo de vida. Además, este plan contempla estrategias para continuar mejorando tanto el proyecto como el sistema de manera constante.

Algunas estrategias de mantenimiento

- Revisar las documentaciones de manera continua
- Utilizar un sistema de control de versiones como Git para actualizar los cambios de documentación.
- Integración continua, donde el flujo de integración se realice automáticamente junto con nuevas versiones de software.
- Recoger feedback de los empleados sobre la utilidad y claridad de la documentación.
- Capacitar a los empleados sobre como utilizar el sistema, como acceder y entender la documentación.

Estrategias de mejora del sistema

- Actualizar las tecnologías con las últimas tecnologías y tendencias en desarrollo de software.
- Implementar pruebas de integración periódicas para identificar y corregir vulnerabilidades.
- Diseñar la arquitectura del sistema para soportar un crecimiento futuro en el número de usuarios y volumen de datos.

- Establecer métricas de éxito y evaluar periódicamente el impacto de las actividades de mantenimiento y mejora.
- Definir un calendario de actividades de mantenimiento, incluyendo revisiones trimestrales, auditorías semestrales, y actividades de mejora continua.