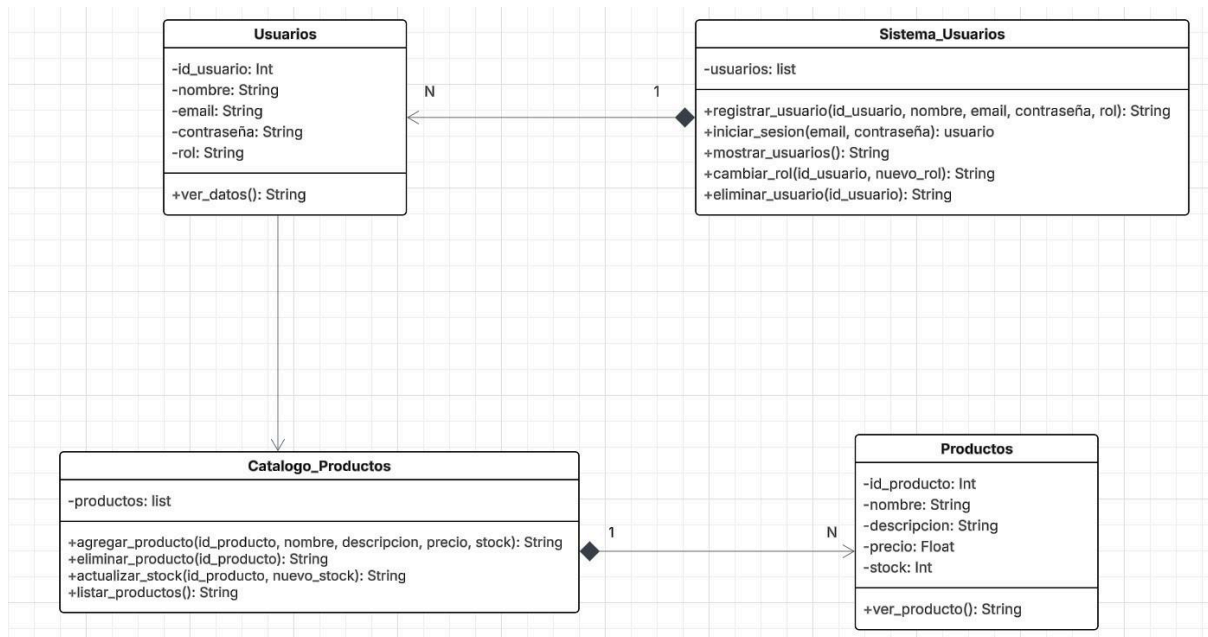


DISEÑO DEL DIAGRAMA DE CLASES



Descripción del diagrama:

El diagrama de clases se compone de 4 clases:

Clase Usuarios:

La cual contiene los siguientes **atributos**:

- id_usuario: Int -> Identificación única del cliente.
- nombre: String -> Nombre del cliente.
- email: String -> Correo electrónico del usuario.
- contraseña -> Contraseña de acceso.
- rol: String -> Rol que va a ocupar cada usuario ("Administrador" o "Estándar").

Y contiene el siguiente **método**:

- ver_datos(): String -> Retorna la información del usuario en formato de texto.

Clase SistemaUsuarios:

La cual contiene el siguiente **atributo**:

- usuarios: Lista -> Lista de objetos Usuario registrado en el sistema.

Y contiene los siguientes **métodos** :

- registrar_usuario(id_usuario, nombre, email, contraseña, rol) -> Registra un nuevo usuario en el sistema y devuelve un mensaje de confirmación.
- iniciar_sesion(email, contraseña) -> Verifica si los datos ingresados son válidos. Devuelve el objeto usuario si es correcto.
- mostrar_usuarios() -> Devuelve la lista de usuarios registrados en forma de texto.
- cambiar_rol(id_usuario, nuevo_rol): String -> Cambia el rol de un usuario en específico y devuelve un mensaje de confirmación.
- eliminar_usuario(id_usuario): String -> Elimina un usuario del sistema a partir de su ID y devuelve un mensaje de confirmación.

Clase Productos:

La cual contiene los siguientes **atributos**:

- id_producto: Int -> Identificación única del producto.
- nombre: String -> Nombre del producto.
- descripción: String -> Descripción del producto.
- precio: Float -> Precio del producto.
- stock: Int -> Cantidad física del producto.

Y contiene el siguiente **método**:

- ver_producto(): String -> Retorna la información del producto en formato de texto.

Clase Catalogo_Productos:

La cual contiene el siguiente **atributo**:

- productos: Lista -> Lista de objetos Productos registrado en el sistema.

Y contiene los siguientes **métodos** :

- agregar_producto(id_producto, nombre, descripción, precio, stock) -> Agrega un nuevo producto en el sistema y devuelve un mensaje de confirmación.
- eliminar_producto(id_producto) -> Elimina un producto del sistema a partir de su ID y devuelve un mensaje de confirmación.
- listar_productos(): String -> Devuelve la lista de productos registrados en forma de texto.
- actualizar_stock(id_producto, nuevo_stock): String -> Actualiza el stock de un producto en específico y devuelve un mensaje de confirmación.

Clases y su justificación:

- ❖ Clase Clientes: Representan a los productores/clientes que entran al sistema.
- ❖ Clase Sistema_Usuarios: Administra el login, registro y gestión de roles.
- ❖ Clase Productos: Representa a los sensores de temperatura y otros insumos agro que ofrece la empresa.
- ❖ Clase Catalogo_Productos: Agrupa y organiza todos los listado de productos que se muestran en la web.

Relaciones entre clases:

- ❖ **Sistema_Usuarios a Usuario(1 a N):** Un usuario puede estar en un solo sistema de usuarios, pero un sistema de usuarios puede albergar a muchos usuarios.
- ❖ **Catalogo_Productos a Producto(1 a N):** Un producto puede pertenecer a un solo catálogo, pero un catálogo puede contener varios productos.
- ❖ **Usuario a Catalogo_Productos(1 a N):** Un usuario puede interactuar con un solo catálogo, pero un catálogo puede ser visto por muchos usuarios.