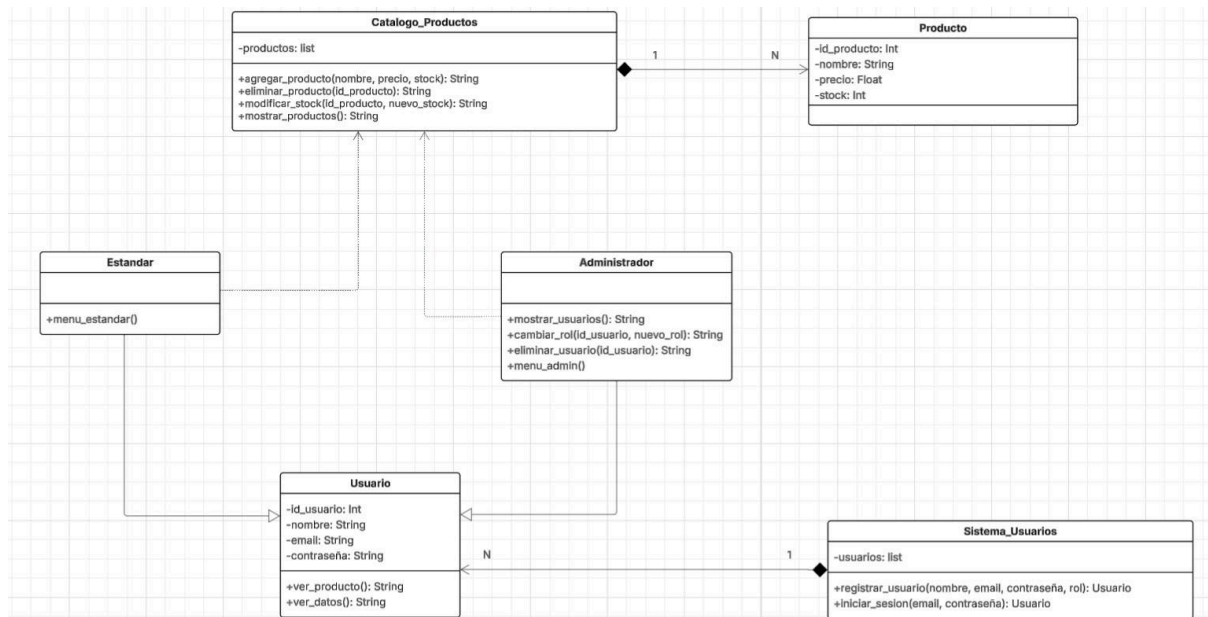


# DISEÑO DEL DIAGRAMA DE CLASES



## Descripción del diagrama:

El diagrama de clases se compone de 4 clases principales y 2 controladores que coordinan operaciones entre objetos (**Sistema\_Usuarios** y **Catalogo\_Productos**):

### Clase Usuario:

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.

Y contiene el siguiente **método**:

- **ver\_datos(): String** -> Retorna la información personal del usuario.
- **ver\_producto(): String** -> Muestra información de un producto disponible.

### Clase Administrador:

Hereda los **atributos** de la clase Usuario

Y contiene los siguientes **métodos**:

- **mostrar\_usuarios(): String** -> Devuelve la lista de usuarios registrados.
- **cambiar\_rol(id\_usuario, nuevo\_rol): String** -> Modifica el rol de un usuario.

- **eliminar\_usuario(id\_usuario): String** -> Elimina un usuario del sistema.
- **menu\_admin():** Despliega el menú con las opciones exclusivas del administrador.

## Clase Estándar:

Hereda los **atributos** de la clase Usuario

Y contiene el siguiente **método**:

- **menu\_estandar():** Muestra el menú con las opciones disponibles para el usuario estándar.

## Controlador Sistema\_Usuarios:

El mismo manipula los siguientes **datos**:

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

Y contiene las siguientes **funciones principales** :

- **registrar\_usuario(id\_usuario, nombre, email, contraseña, rol)** -> Registra un nuevo usuario en el sistema y devuelve el objeto creado.
- **iniciar\_sesion(email, contraseña)** -> Verifica si los datos ingresados son válidos. Devuelve el objeto usuario si es correcto.

## Clase Producto:

La cual contiene los siguientes **atributos**:

- **id\_producto: Int** -> Identificación única del producto.
- **nombre: String** -> Nombre del producto.
- **precio: Float** -> Precio del producto.
- **stock: Int** -> Cantidad física del producto.

## Controlador Catalogo\_Productos:

El mismo manipula los siguientes **datos**:

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

Y contiene las siguientes **funciones principales** :

- **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.

- **mostrar\_productos(): String** -> Devuelve la lista de productos registrados en forma de texto.
- **modificar\_stock(id\_producto, nuevo\_stock): String** -> Actualiza el stock de un producto en específico y devuelve un mensaje de confirmación.

### Clases/Controladores y su justificación:

- ❖ **Clase Usuario / Administrador / Estándar:** Representan los distintos tipos de usuarios que pueden interactuar con el sistema, cada uno con permisos y menús diferenciados.
- ❖ **Controlador Sistema\_Usuarios:** Centraliza la gestión de usuarios, incluyendo registro, autenticación e inicio de sesión.
- ❖ **Clase Productos:** Define las características de los productos.
- ❖ **Controlador Catalogo\_Productos:** Administra y organiza el conjunto de productos disponibles para su consulta o modificación.

### Relaciones entre clases y controladores:

- ❖ **Sistema\_Usuarios a Usuario (1 a N):** Un sistema de usuarios puede contener múltiples usuarios, pero cada usuario pertenece a un solo sistema.
- ❖ **Catalogo\_Productos a Producto (1 a N):** Un catálogo puede incluir múltiples productos, mientras que cada producto pertenece a un solo catálogo.
- ❖ **Administrador y Estándar heredan de Usuario:** Mantienen los mismos atributos, pero difieren en los métodos y permisos asociados.
- ❖ **Estándar a Catalogo\_Productos (Dependencia de uso):** El usuario estándar depende del catálogo de productos para consultar la información disponible. Utiliza los métodos de la clase Catalogo\_Productos—como `mostrar_productos()`— para visualizar los productos, sin tener permisos de modificación sobre ellos.
- ❖ **Administrador a Catalogo\_Productos (Dependencia de uso):** El administrador depende del catálogo de productos para gestionar su contenido. Emplea los métodos de la clase Catalogo\_Productos—como `agregar_producto()`, `eliminar_producto()` y `modificar_stock()`— con el fin de administrar la información de los productos registrados.