



Universidad Católica del Norte.  
Facultad de Ingeniería y Ciencias Geológicas.  
Departamento de Ingeniería de Sistemas y Computación.

---

# CONTRATOS E INFORME

## TALLER 3

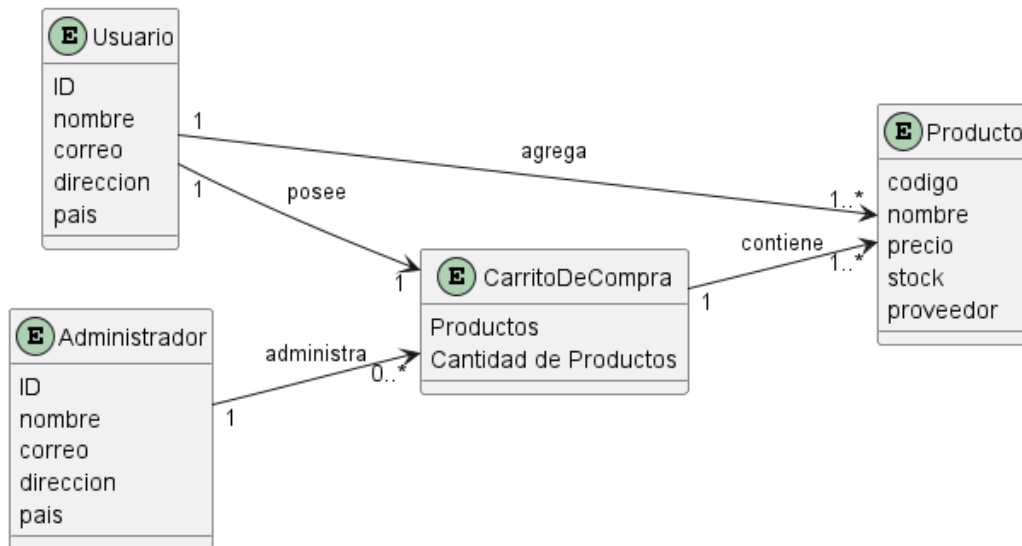
---

Programación Avanzada

Franco Olivos Macaya  
[Franco.olivos@alumnos.ucn.cl](mailto:Franco.olivos@alumnos.ucn.cl)  
21.645.360-3  
Paralelo C2

Profesor: Tomás Reimann  
Ayudante: Edgardo Ortiz

# Modelo de Dominio



## Explicación dominio:

### Usuario → Carrito de compras:

- Cada usuario puede tener un carrito de compras, suponiendo que puede tener solo un carrito de compras a la vez.
- El usuario puede realizar mas de una compra en el mismo carrito.
- Cada usuario que se registre podrá tener su propio carrito de compras en el sistema.

### Usuario → Productos:

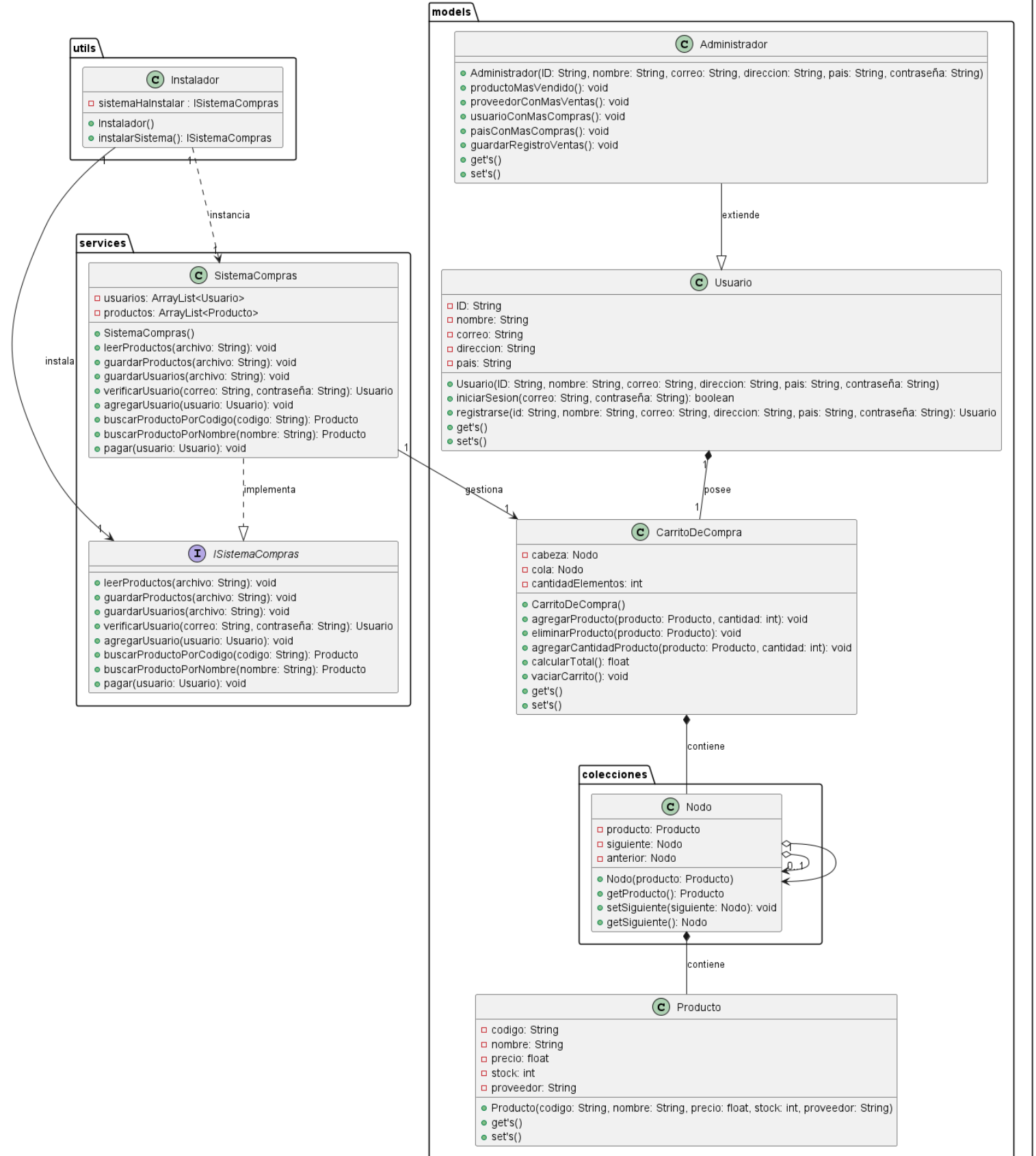
- Un usuario puede agregar o eliminar uno o mas productos a su carrito de compras.
- Por cada compra realizada por el usuario, se lleva el registro de los productos que compró en dicha compra.

### Administrador → Carrito de compras:

- Se entiende que el administrador es quien puede gestionar y administrar además del usuario. El administrador tiene la posibilidad de ver y supervisar todas las estadísticas de las ventas.
- El administrador tiene visibilidad sobre todas las compras en el sistema.

# Diagrama de clases

cl\_uen\_disc\_pa\_taller3



## Explicación Diagrama de Clases:

- **Usuario:** Representa a los usuarios del sistema de compras. Cada usuario tiene un ID único, nombre, correo electrónico, dirección y país. Los usuarios pueden iniciar sesión y registrarse en el sistema.
- **Producto:** Representa los productos disponibles en la tienda. Cada producto tiene un código único, nombre, precio, stock y proveedor.
- **Administrador:** Hereda los atributos de usuario y tiene privilegios adicionales en el sistema. Además de las características comunes de un usuario, el administrador puede realizar acciones como determinar el producto más vendido, el proveedor con más ventas, etc.
- **CarritoDeCompra:** Representa el carrito de compra de un usuario. Contiene una lista de productos que el usuario ha agregado al carrito, así como métodos para agregar, eliminar, calcular el total y vaciar el carrito.
- **ISistemaCompras:** Es la interface que define los métodos que debe proporcionar el sistema de compras, como leer y guardar productos y usuarios, verificar usuarios, agregar usuarios, y buscar productos por código o nombre.
- **SistemaCompras:** Es la implementación de la interface ISistemaCompras. Administra la lista de usuarios y productos, y proporciona métodos para realizar las operaciones definidas en la interface, como leer y guardar productos y usuarios, verificar usuarios, agregar usuarios, y buscar productos por código o nombre.
- **Instalador:** Clase encargada de instalar el sistema de compras. Contiene una referencia al sistema que se va a instalar (ISistemaCompras) y proporciona un método para instalar el sistema.

## Listas utilizadas:

- **Lista con nexos dobles:** Se decidió usar este tipo de lista para el Carrito de Compra ya que permitirá agregar un nuevo nodo al final del carrito de forma más eficiente, sin embargo, también tiene desventajas como el eliminar algún producto que esté en medio de la lista que puede llegar a ser menos eficiente.
- **ArrayList:** En este caso se utilizarán ArrayList para las listas de usuarios y productos en la clase SistemaCompras ya que nos permite buscar o verificar usuarios y productos más rápidamente, además de que se pide el uso de ellas.

## Explicación del código:

Primero, se instancia un objeto SistemaCompras, que es responsable de gestionar las operaciones del sistema, además se carga la lista de productos desde el archivo "productos.txt" utilizando el método leerProductos.

Se crea un objeto Usuario con los detalles proporcionados como ID, nombre, correo electrónico, dirección y país. Luego se muestra por pantalla un menú inicial con las opciones de “inicio de sesión” y “salir”.

- **Inicio de sesión:** Se solicita al usuario que ingrese su correo electrónico y contraseña. Se llama al método iniciarSesion del SistemaCompras para verificar las credenciales. Si las credenciales son correctas, se devuelve un objeto Usuario (puede ser un Administrador o un usuario normal). Se muestra un mensaje de confirmación y se redirige al menú correspondiente (menuAdministrador o menuUsuario).
- **Menú del Administrador:** Si se inicia sesión como administrador, se muestra un menú con las siguientes opciones: buscar un producto, gestionar carrito de compras, estadísticas y salir.
- **Menú del Usuario:** Si se inicia sesión como usuario normal, se muestra un menú con las siguientes opciones: Buscar un producto, gestionar carrito de compras y salir.
- **Gestión de Carrito:** En ambos menús de administrador y usuario se permite al usuario buscar productos por código o nombre, además se muestra la información del producto si se encuentra, incluyendo precio, stock y proveedor y se ofrece la opción de agregar el producto al carrito de compras del usuario.
- **Especificaciones del Administrador:** El administrador puede realizar operaciones como gestionar el carrito de compras de cualquier usuario, eliminar productos del carrito, aumentar la cantidad de productos, etc. También puede acceder a estadísticas del sistema, como el producto más vendido y el total de ventas, por ejemplo.
- **Pasar a pagar:** Cuando se decide pasar a pagar desde el carrito de compras se muestra el total a pagar y se confirma el pago y se realiza la transacción utilizando el método pagar del SistemaCompras.
- **Finalización del Programa:** El usuario puede elegir la opción Salir en cualquier momento para finalizar el programa o volver al menú anterior.

## Contratos

OPERACIONES	agregarUsuario(Usuario usuario)
DESCRIPCION	Agrega un nuevo usuario al sistema.
PRECONDICIONES	El usuario por agregar no debe ser nulo. El correo del usuario no debe estar registrado previamente en el sistema.
POSTCONDICIONES	El usuario es agregado correctamente al sistema.

OPERACIÓN	buscarProductoPorCodigo(String codigo)
DESCRIPCION	Busca un producto por su código único.
PRECONDICIONES	El código del producto no debe ser nulo ni vacío. Debe existir el producto.
POSTCONDICIONES	Si se encuentra un producto con el código especificado, se devuelve ese producto.

OPERACIÓN	buscarProductoPorNombre(String nombre)
DESCRIPCION	Busca un producto por su nombre.
PRECONDICIONES	El nombre del producto no debe ser nulo ni vacío. Debe existir el producto.
POSTCONDICIONES	Si se encuentra un producto con el nombre especificado, se devuelve ese producto.

OPERACIÓN	pagar(Usuario usuario)
DESCRIPCION	Permite al usuario realizar el pago de los productos en su carrito.
PRECONDICIONES	El usuario debe haber iniciado sesión correctamente. El carrito de compras del usuario no debe estar vacío.
POSTCONDICIONES	Se realiza el proceso de pago de los productos en el carrito del usuario. Se actualiza el stock de los productos en el sistema. Se genera una boleta electrónica con los detalles de la compra.

OPERACIÓN	eliminarProducto(Producto producto)
DESCRIPCION	Elimina un producto del carrito de compras del usuario.
PRECONDICIONES	El producto para eliminar no debe ser nulo. El producto debe existir en el carrito de compras del usuario.

POSTCONDICIONES	El producto es eliminado correctamente del carrito de compras del usuario y se actualiza el total de la compra en el carrito.
-----------------	---

OPERACIÓN	leerProductos(String archivo)
DESCRIPCION	Lee los productos desde un archivo y los carga en el sistema.
PRECONDICIONES	El archivo especificado debe existir y ser accesible para su lectura.
POSTCONDICIONES	Los productos son cargados correctamente en el sistema.

OPERACIÓN	guardarProductos(String archivo)
DESCRIPCION	Guarda los productos del sistema en un archivo.
PRECONDICIONES	El sistema debe contener al menos un producto. El archivo especificado debe poder sobrescribirse.
POSTCONDICIONES	Los productos son guardados correctamente en el archivo.

OPERACIÓN	verificarUsuario(String correo, String contraseña)
DESCRIPCION	Verifica las credenciales de un usuario para permitir el inicio de sesión.
PRECONDICIONES	El correo y la contraseña no deben ser nulos ni vacíos. El usuario correspondiente debe estar registrado en el sistema.
POSTCONDICIONES	Si las credenciales son válidas, el usuario correspondiente inicia sesión.

OPERACIÓN	guardarUsuarios(archivo: String)
DESCRIPCION	Guarda la información de los usuarios en un archivo.
PRECONDICIONES	El sistema debe haber creado al menos un usuario durante la ejecución.
POSTCONDICIONES	La información de los usuarios creados se guarda en el archivo especificado.

## **Tiempo dedicado (Aproximado)**

- Creación modelo de dominio: 00:40 horas/minutos
- Creación Diagrama de clases: 03:30 horas/minutos
- Creación Informe y contratos, primera entrega: 01:50 horas/minutos
  
- Creación del Código: 9:00 horas/minutos
- Entrega final del Informe: 1:00 horas/minutos