

# Orientación a Objetos I

## 2025

Explicación de práctica #8  
correspondiente a los ejercicios de la  
semana del 27 de Octubre



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Actividades de la semana anterior

- Ejercicio 19: Alquiler de propiedades
- Ejercicio 20: Políticas de cancelación
- Ejercicio 21: Envío de paquetes

## **Esta semana:**

- Ejercicio 22. Sistema de pedidos
- Ejercicio 23. Poolcar
- Ejercicio 24. GreenHOOme
- Ejercicio 25. Bag



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Ejercicio 22 - Sistema de pedidos

**Tarea:** Realice un diagrama de clases UML donde identifique todas las clases del dominio, sus atributos, roles, cardinalidades y asociaciones. No debe incluir comportamiento.



FACI

Se desea diseñar un sistema para una empresa que ofrece viandas preparadas y artículos de supermercado. De las viandas se conoce su nombre, precio, si es apta para celíacos y disponibilidad. De los artículos de supermercado se conoce su nombre, categoría, precio, peso en kg y stock actual.

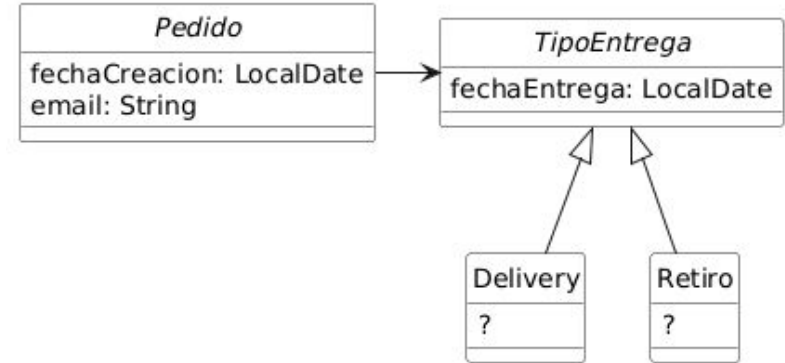
Para realizar un pedido, no es necesario registrarse en la aplicación, alcanza con indicar el email al momento de solicitar los productos. Por cuestiones logísticas, cada pedido deberá ser de uno de los dos rubros: supermercado o viandas. Para coordinar la entrega, se puede optar por envío por delivery, en ese caso se debe indicar la dirección y fecha/hora en la que se recibirá el pedido; o bien retiro en sucursal, para lo cual deberá indicar el DNI de la persona autorizada a retirar y la fecha/hora estimadas en la que retira.

La empresa dispone de cocina sin TACC, por lo que es muy importante al momento de realizar un pedido que se indique si es apto celíaco o no. Los pedidos de supermercado pueden obtener un 10% de descuento en caso de que se utilice un cupón de descuento al solicitarlos.

**Tarea:** Realice un diagrama de clases UML donde identifique todas las clases del dominio, sus atributos, roles, cardinalidades y asociaciones. No debe incluir comportamiento.

Recordar los pasos del proceso de modelado:

- Identificar los conceptos
- Identificar los atributos
- Descartar lo que no usamos
- Identificar relaciones
- Realizar modelo
- El entregable es el modelo, pero lo anterior nos ayuda a construirlo



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Ejercicio 23 - Poolcar

**Tarea:** Realice un diagrama de clases UML donde identifique todas las clases del dominio, sus atributos, roles, cardinalidades y asociaciones. No debe incluir comportamiento.

Poolcar es una aplicación para organizar viajes de larga distancia, que permite compartir gastos y abaratar el costo del viaje.

De cada usuario se conoce su nombre y dirección. Además se mantiene un estado de cuenta, que se usa para el pago de sus viajes. Al momento de su alta, los usuarios se registran como conductor o pasajero, lo cual es una característica que no se puede modificar.

Los conductores deben registrar al menos un vehículo. De cada uno se conoce una descripción, su capacidad (incluyendo al conductor), el año de fabricación y el valor de mercado. Un vehículo pertenece siempre a un único conductor, y cada conductor sólo podrá viajar en calidad de chofer de su propio vehículo.

Los viajes son creados por los conductores y quedan asociados a uno de sus vehículos. De cada viaje se registra el costo total, la localidad de origen, la localidad de destino y la fecha de salida.

Los pasajeros pueden inscribirse en los viajes creados por conductores, siempre y cuando haya lugares disponibles en el vehículo y su saldo de cuenta sea positivo. La inscripción sólo está permitida hasta dos días antes de la fecha de salida del viaje. Cuando un viaje se lleva a cabo, el costo total del mismo se divide en partes iguales entre todos los participantes, tanto pasajeros como conductor, descontándose de sus respectivos saldos de cuenta. Además, se aplican descuentos:



# Ejercicio 24 - GreenHOOme

## Tareas:

1. Instancie los elementos correspondientes ...
2. Diseñe los casos de prueba para el método *calcularCosto* de una orden, teniendo en cuenta los conceptos de valores de borde y particiones equivalentes vistos en la teoría.

GreenHome es una empresa que promueve prácticas sustentables mediante la venta de productos ecológicos y la realización de servicios de mantenimiento en los hogares.

El sistema gestiona a los usuarios del servicio, registrando para cada uno las órdenes de compra de productos y las órdenes de servicio que ha solicitado para el mantenimiento de su hogar. Las órdenes de servicio pueden requerir la intervención de técnicos especializados y el uso de determinados productos, mientras que las órdenes de compra solo registran los productos adquiridos.

El costo de una orden de compra se calcula sumando los valores de los productos adquiridos, mientras que el costo de una orden de servicio se obtiene sumando el costo de los productos utilizados y el valor correspondiente a las horas de trabajo de los técnicos involucrados.

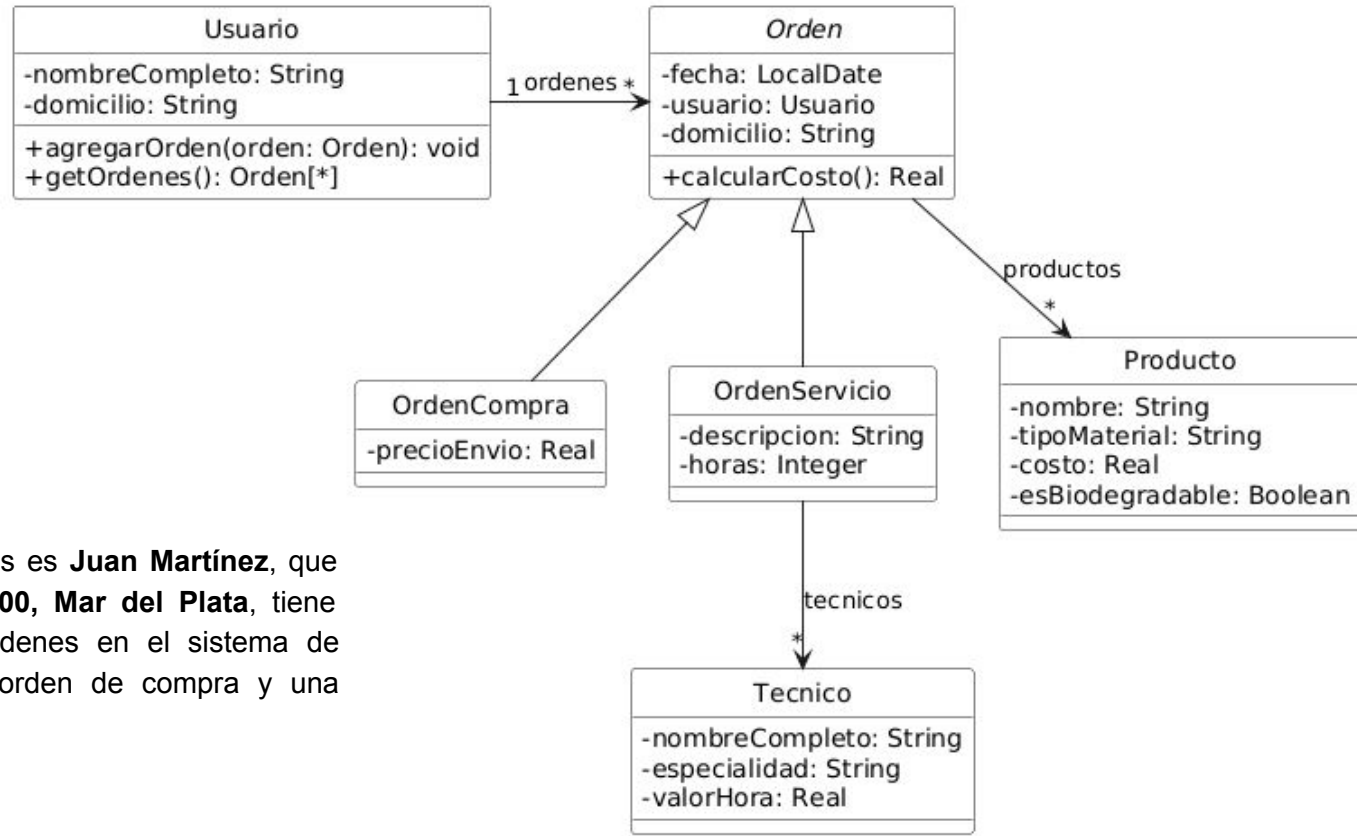
El siguiente diagrama muestra el diseño general de la aplicación.



FACI



# Ejercicio 24 - GreenHOOme

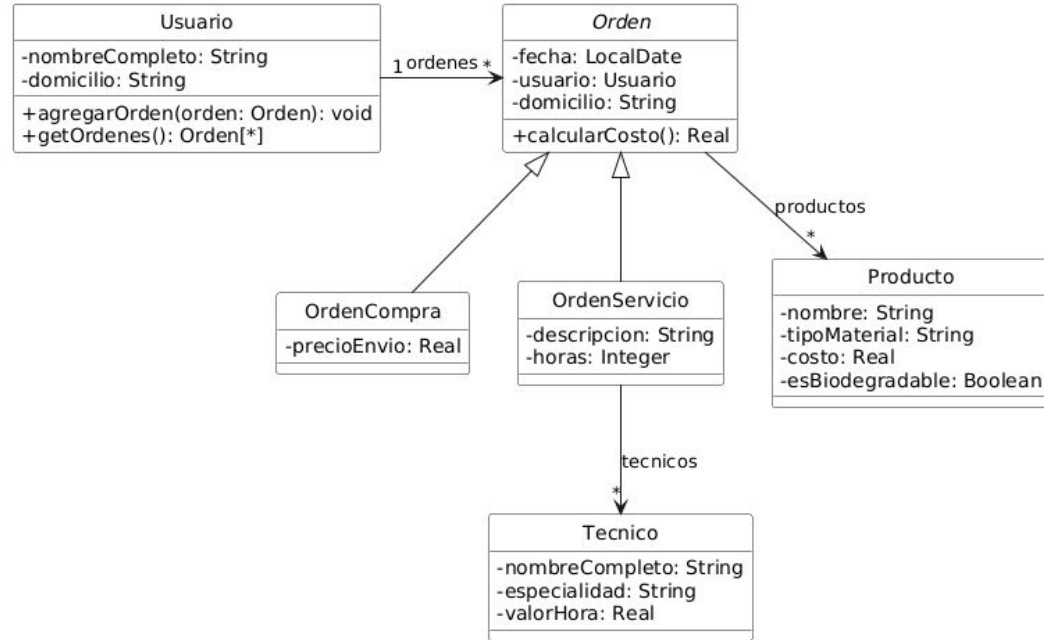


Uno de los usuarios es **Juan Martínez**, que vive en **Larrea 5800, Mar del Plata**, tiene registradas dos órdenes en el sistema de GreenHome: una orden de compra y una orden de servicio.

# Ejercicio 24 - GreenHOOme

Uno de los usuarios es **Juan Martínez**, que vive en **Larrea 5800, Mar del Plata**, tiene registradas dos órdenes en el sistema de GreenHome: una orden de compra y una orden de servicio.

La orden de compra corresponde a la adquisición de productos ecológicos. En este caso, Juan compró un panel solar y una compostera. El panel solar está hecho con materiales reciclables y tiene un costo de \$35.000 pesos, mientras que la compostera es un producto biodegradable que cuesta \$8.000 pesos. Esta orden se registró con la fecha actual, tiene asociado el domicilio de Juan y no interviene ningún técnico, ya que se trata de una compra directa. El costo total de la orden se obtiene sumando el valor de ambos productos, lo que da un total de \$43.000 pesos.





# Ejercicio 25 - Bag

```
Bag<String> bag = new ...  
bag.add("hola");  
bag.add("hola");  
bag.add("mundo");
```

```
bag.ocurrencesOf("hola") -> 2  
bag.size() -> 3
```

"hola"	2
"mundo"	1



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Ejercicio 25 - Bag

```
public interface Bag<T> extends Collection<T> {  
    @Override  
    boolean add(T element);  
    int occurrencesOf(T element);  
    void removeOccurrence(T element);  
    void removeAll(T element);  
    @Override  
    int size();  
}
```



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Foros de consulta

## Cómo preguntar en el foro

Breve guía para poder sacar el mejor provecho al foro y a la convivencia a través de las preguntas y respuestas.

### [Cómo preguntar en el foro](#)

[Antes de Preguntar: Busca una respuesta por tus propios medios](#)

[Elegí el foro específico](#)

[Elegí un título apropiado para la pregunta](#)

[No envíes una solución para que la corrijan](#)

[Describir qué estás intentando hacer](#)

[Describir el problema y lo que has intentado para resolverlo](#)

[Escribir claro](#)

[No solicites respuestas a tu correo](#)

[Si no entendés la respuesta](#)

[Terminá con una breve nota de conclusión.](#)

[Evitá el "Me sumo al pedido"](#)



Foro: Ejercicio 12 - Job Scheduler



Foro: Ejercicio 13 - ¡A implementar Inversores!



Foro: Ejercicio 14 - Volumen y superficie de sólidos



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA