

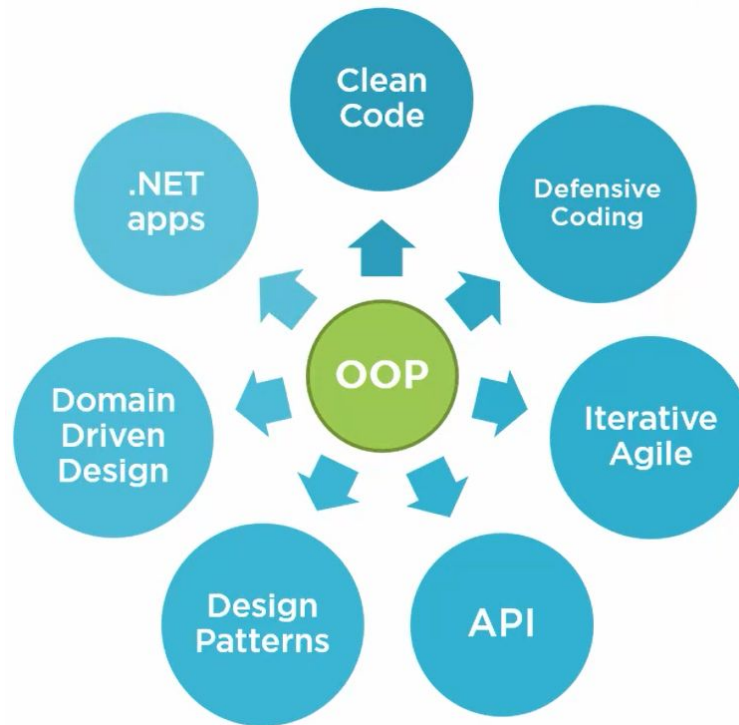


# PОО (Programación Orientada a Objetos)

# Objetivos

- Identificar clases desde los requerimientos
- Separar responsabilidades
- Definir relaciones entre clases
- Aprovechar la reutilización
- Pilares: Encapsulación, Abstracción, Herencia, Polimorfismo.

# Introducción



# Términos

Object

!=

Class

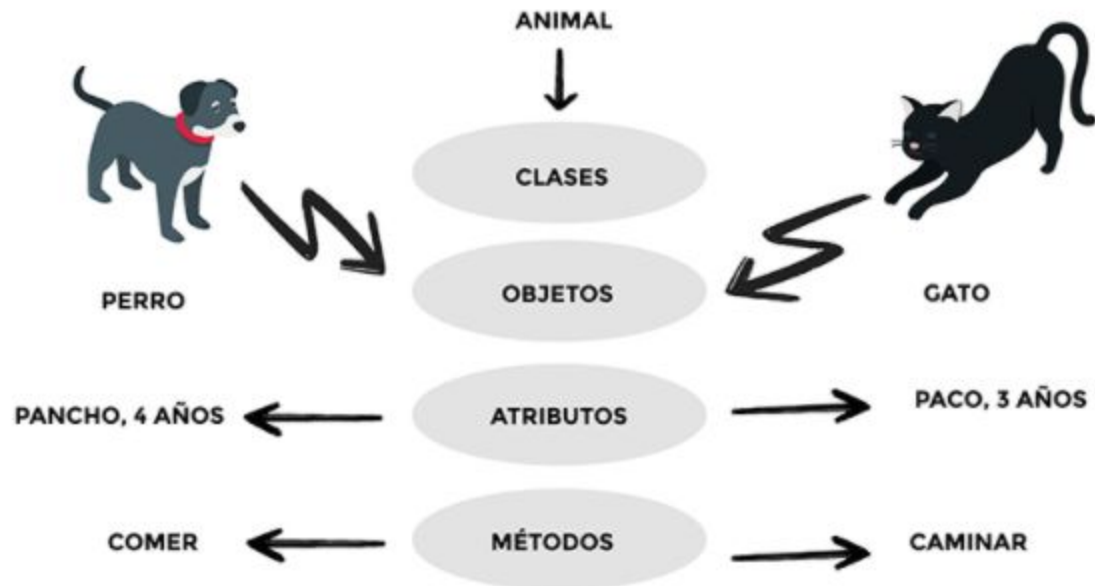
# Object

!=

# Class

```
var customer = new Customer();  
customer.FirstName = "Frodo";  
customer.Validate();
```

```
public class Customer  
{  
    public int CustomerId { get; set; }  
    public string EmailAddress { get; set; }  
    public string FirstName { get; set; }  
    public string LastName { get; set; }  
    public bool Validate(){ ... }  
}
```



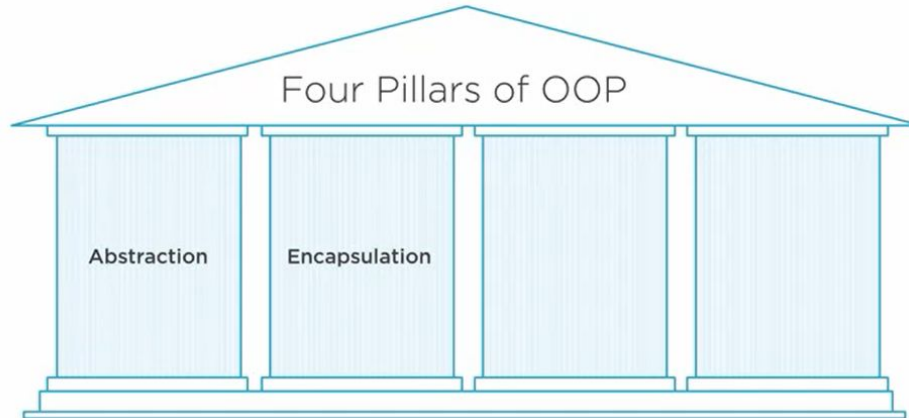
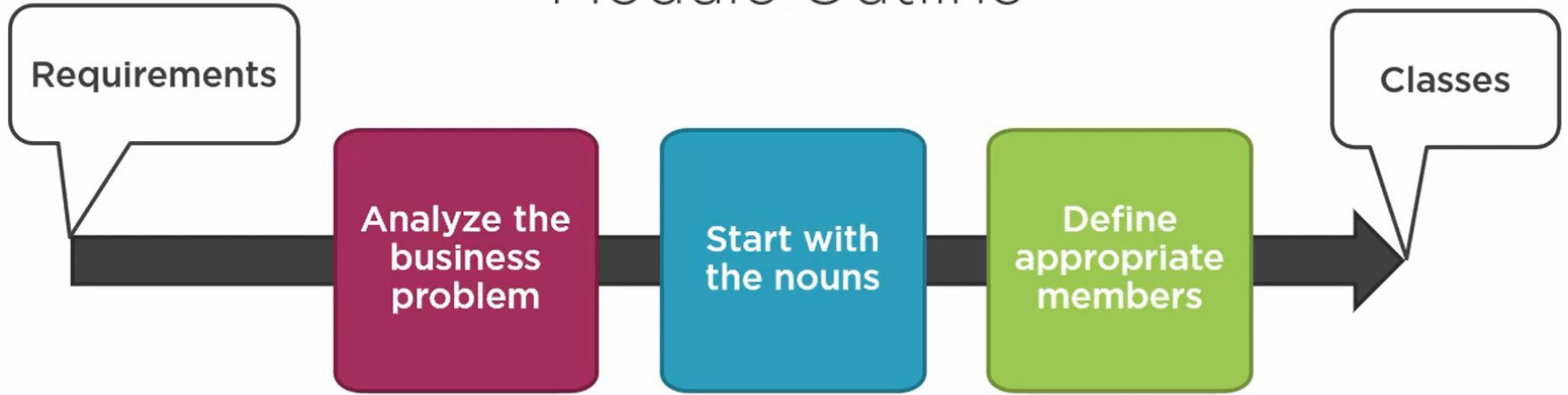
# Programación orientada a objetos

- Un conjunto de técnicas para diseñar y construir aplicaciones que son:
  - flexibles,
  - bien definidas y
  - testeables

enfocándose en cómo los objetos interactúan unos con otros.

- Identificando clases, separando responsabilidades, estableciendo relaciones y pensando en la reutilización.

# Module Outline





# Requerimientos



Manage business,  
residential, government,  
and educator types of  
customers



Manage our products



Accept orders from  
customers online or  
through our call center



Manage business,  
residential, government,  
and educator types of  
customers

**Customer**



Manage our products

**Product**



Accept orders from  
customers online or  
through our call center

**Order**



- Customer's name (Last name, first name)
- Email address
- Home and work addresses



- Product name
- Description
- Current price



- Customer
- Order date
- Shipping address
- Products and quantities ordered

## Define Appropriate Members

### Customer

- Name
- Email address
- Home address
- Work address

### Product

- Product name
- Description
- Current price

### Order

- Customer
- Order date
- Shipping address
- Product
- Quantity

# Define Appropriate Members

## Customer

- Name
- Email address
- Home address
- Work address

## Product

- Product name
- Description
- Current price

## Order

- Customer
- Order date
- Shipping address
- Order items

## Order Item

- Product
- Quantity

## Define Appropriate Members

### Customer

- Name
- Email address
- Home address
- Work address
- Validate()
- Retrieve()
- Save()

### Product

- Product name
- Description
- Current price
- Validate()
- Retrieve()
- Save()

### Order

- Customer
- Order date
- Shipping address
- Order items
- Validate()
- Retrieve()
- Save()

### Order Item

- Product
- Quantity
- Validate()
- Retrieve()
- Save()

# Define Appropriate Members

## Customer

- Name
- Email address
- Home address
- Work address
- Validate()
- Retrieve()
- Save()

## Product

- Product name
- Description
- Current price
- Validate()
- Retrieve()
- Save()

## Order

- Customer
- Order date
- Shipping address
- Order items
- Validate()
- Retrieve()
- Save()

## Order Item

- Product
- Quantity
- Purchase price
- Validate()
- Retrieve()
- Save()

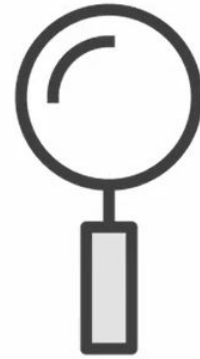
# Abstraction



Simplifying reality



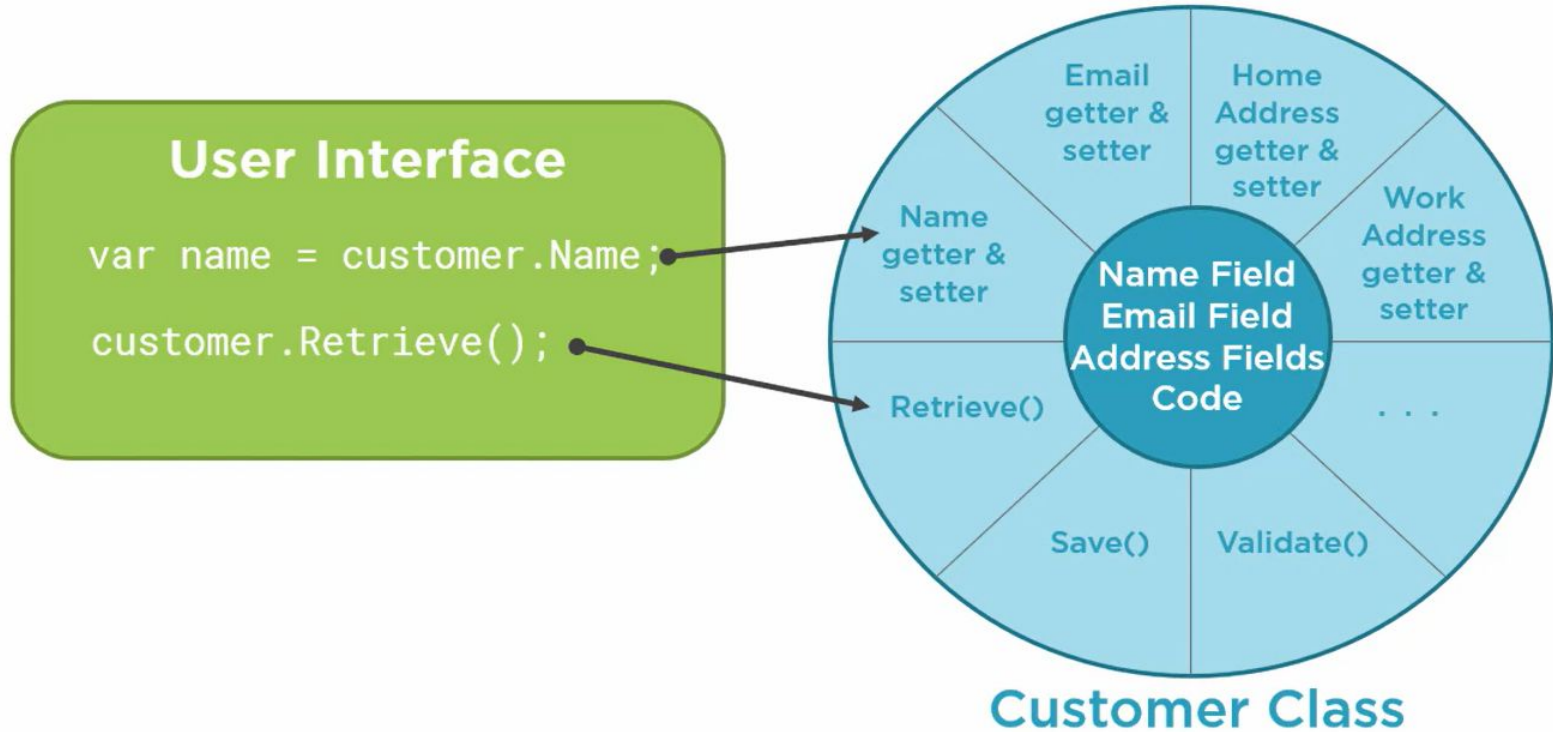
Ignoring extraneous  
details



Focusing on what is  
important for a  
purpose



# Encapsulation



# Caso de Estudio

Se desea diseñar el software necesario para una red bancaria provista de cajeros automáticos (ATM, *automatic teller machines*), que serán compartidos por un consorcio de bancos. Cada banco dispone de su propio ordenador, provisto de software propio, que lleva la información sobre sus cuentas y procesa las transacciones que actúan sobre dichas cuentas. A este ordenador están conectadas las estaciones de cajero, que son propiedad del banco y en las que operan cajeros humanos, que pueden crear cuentas e introducir transacciones sobre ellas.

Los cajeros automáticos aceptan tarjetas de crédito, interaccionan con el usuario, se comunican con un ordenador central para llevar a cabo las transacciones, entregan dinero en efectivo al usuario e imprimen recibos. El sistema llevará correctamente el registro de las transacciones efectuadas, cumplirá características aceptables de seguridad y manejará correctamente accesos concurrentes a la misma cuenta.

El coste de desarrollo de la parte compartida del sistema se dividirá entre los bancos que forman parte del consorcio en función del número de clientes provistos de tarjetas de crédito.

# Casos de Uso

Escenario Normal

- El cajero automático pide al cliente que inserte la tarjeta de crédito.
- El cliente inserta la tarjeta de crédito.
- El cajero automático acepta la tarjeta de crédito y lee el número de tarjeta y el código del banco.
- El cajero automático pide la contraseña al cliente.
- El cliente teclea "1234".
- El cajero automático envía el número de tarjeta, el código del banco y la contraseña al consorcio.
- El consorcio envía el número de tarjeta y la contraseña al banco.
- El banco notifica la aceptación al consorcio.
- El consorcio notifica la aceptación al cajero automático.
- El cajero automático pide al cliente que elija el tipo de transacción: retirada de fondos, depósito, transferencia, información.
- El cliente selecciona retirada de fondos.
- El cajero automático pide al cliente que teclee la cantidad.
- El cliente teclea 25000.
- El cajero automático comprueba que la cantidad está dentro de los límites generales.
- El cajero automático genera una transacción y la envía al consorcio.
- El consorcio pasa la transacción al banco.
- El banco aprueba la transacción.
- El banco actualiza la cuenta.
- El banco envía al consorcio la notificación de aceptación y el nuevo saldo de la cuenta.
- El consorcio envía al cajero automático la notificación de aceptación y el nuevo saldo de la cuenta.
- El cajero automático entrega el dinero al cliente.
- El cliente toma el dinero.
- El cajero automático pregunta al cliente si quiere un recibo.
- El cliente contesta SI.
- El cajero automático imprime un recibo y pide al cliente que lo tome.
- El cliente toma el recibo.
- El cajero automático pregunta al cliente si quiere hacer otra operación.
- El cliente contesta NO.
- El cajero automático expulsa la tarjeta de crédito e indica al cliente que la tome.
- El cliente toma la tarjeta de crédito.
- El cajero automático vuelve a la situación inicial.

# Casos de Uso

Escenario con Problemas

- El cajero automático pide al cliente que inserte la tarjeta de crédito.
- El cliente inserta la tarjeta de crédito.
- El cajero automático acepta la tarjeta de crédito y lee el número de tarjeta y el código del banco.
- El cajero automático pide la contraseña al cliente.
- El cliente teclea "9999".
- El cajero automático envía el número de tarjeta, el código del banco y la contraseña al consorcio.
- El consorcio envía el número de tarjeta y la contraseña al banco.
- El banco notifica el rechazo al consorcio.
- El consorcio notifica el rechazo al cajero automático.
- El cajero automático notifica el rechazo al cliente y pide que teclee de nuevo la contraseña.
- El cliente teclea "1234".
- El cajero automático envía el número de tarjeta, el código del banco y la contraseña al consorcio.
- El consorcio envía el número de tarjeta y la contraseña al banco.
- El banco notifica la aceptación al consorcio.
- El consorcio notifica la aceptación al cajero automático.
- El cajero automático pide al cliente que elija el tipo de transacción: retirada de fondos, depósito, transferencia, información.
- El cliente selecciona retirada de fondos.
- El cajero automático pide al cliente que teclee la cantidad.
- El cliente teclea CANCELAR.
- El cajero automático expulsa la tarjeta de crédito e indica al cliente que la tome.
- El cliente toma la tarjeta de crédito.
- El cajero automático vuelve a la situación inicial.

# Ejercicios

1. Identifica y diseña las clases para un Sistema de Gestión de Biblioteca (prestamos, devolución)
2. Identifica y diseña las clases para un Taller de Vehiculos (agendar citas, recepción de vehículos, tareas a realizar, retiro de vehículo).