

<

Paradigmas de Programación

/>



Clase 1
UTN – FRLP

1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 0 1

</Desarrollo de la clase 1

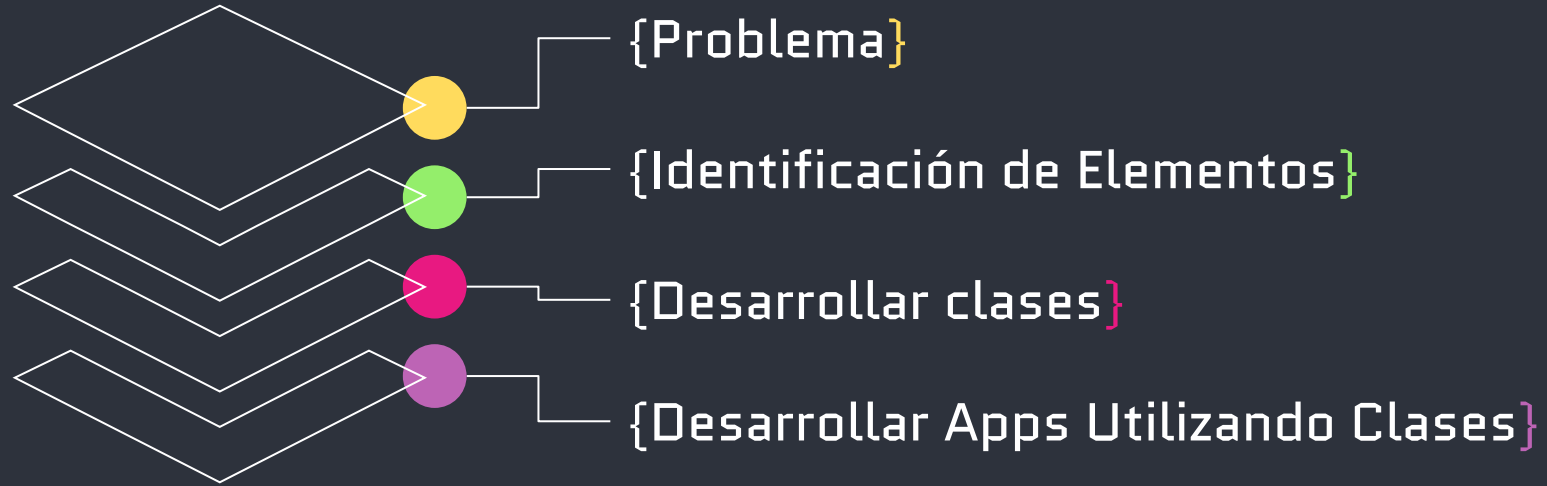
- Nuevo paradigma para resolver problemas.
- Definición de POO.
- Definición de Objetos.
- Definición de Clases.
- Definición de Instancia.
- 4 pilares y Binding Dinámico.
- Ejercicio para resolver.

</Nuevo paradigma para resolver problemas

Dado un problema debemos:



</Nuevo paradigma para resolver problemas



Si llegáramos a necesitar una clase que ya tenemos creada durante el desarrollo de una nueva aplicación, simplemente tendríamos que importarla e implementarla conociendo los datos y comportamiento que maneja

</¿Qué es el POO?

La Programación Orientada a Objetos se define como una manera de programar específica, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones.

Con la POO se resuelven los problemas escribiendo programas en términos de clases, objetos, propiedades, métodos, diagrama de clases entre otros conceptos; ya no se escribe como se solía hacer en la programación estructurada.

</¿Con qué trabajamos?



</¿Qué es una clase?

Son la definición de un objeto. Es un modelo para definir las características y comportamiento de un objeto.



Cada modelo [clase] define un conjunto de variables [estado], y métodos para operarlos [comportamiento].



Esto permite abstraer los datos y sus operaciones asociadas.



</¿Qué es un objeto?

Es la representación de una entidad del sistema. Se conforma de



- Estado: representado por los atributos. Son las propiedades que caracterizan la entidad.
- Comportamiento: es el conjunto de funciones o métodos que operan el estado del objeto.



</¿Qué es una instancia?

Al crear un objeto, estamos *instanciando* una clase.



Una instancia es un objeto particular de la clase. Todas las instancias de una clase responden al mismo comportamiento.



{Clase}

{Instancia[s]}

Características
(atributos)

Auto
<ul style="list-style-type: none">- anio: int- color: int- marca: int- modelo: int- tipo: int
<p>«property get»</p> <ul style="list-style-type: none">+ getAnio(): int+ getColor(): int+ getMarca(): int+ getModelo(): int+ getTipo(): int <p>«property set»</p> <ul style="list-style-type: none">+ setAnio(int): void+ setColor(int): void+ setMarca(int): void+ setModelo(int): void+ setTipo(int): void

Auto1: Auto

anio = 2014
color = blanco
marca = Peugeot
modelo = 208
tipo = Sedan 4P

Auto3: Auto

anio = 2010
color = Negro
marca = Toyota
modelo = Hilux
tipo = Utilitario

Auto2: Auto

anio = 2017
color = Rojo
marca = Chevrolet
modelo = Camaro
tipo = Sedan 2P

Acciones
(métodos)

1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

</ Los cuatro pilares de la POO

{01}

Herencia



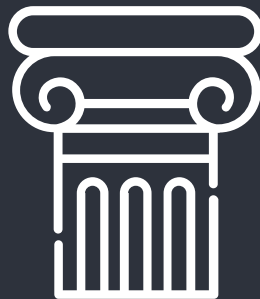
{02}

Polimorfismo



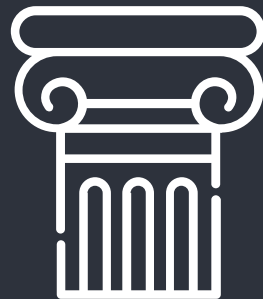
{03}

Abstracción



{04}

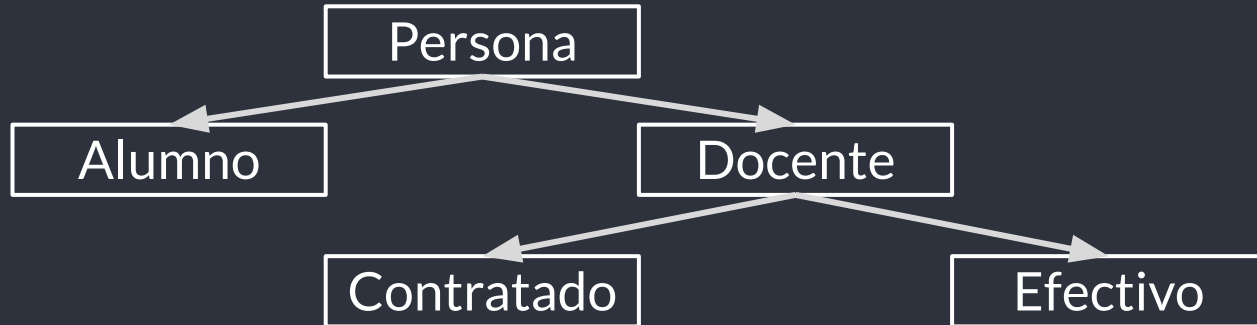
Encapsulamiento



</Herencia

Las clases están organizadas en una jerarquía de clases, donde las subclases heredan atributos y/o comportamiento de las clases padres.

Las subclases pueden agregar nuevos atributos, o solo cambiar el comportamiento de la clase padre.



</Polimorfismo

Polimorfismo: Es la capacidad que tienen los objetos de tener métodos con el mismo nombre y comportamientos diferentes. La interpretación del método será según el objeto receptor.

Binding dinámico: significa que la ligadura entre el objeto receptor y el método que se va a ejecutar se realiza en tiempo de ejecución



</Abstracción

La abstracción viene de la etapa donde analizamos los atributos y el comportamiento que las entidades deben tener.

Es la propiedad por la cual las entidades quedan representadas únicamente por los datos de interés para el problema, y no por todos los atributos que una entidad pueda tener en la vida real.

Por ejemplo: omitir color de ojos, altura, peso y talla de zapatillas de una persona para un sistema de gestión universitario.



</Encapsulamiento

Toda entidad queda comprendida dentro de un paquete compuesto de estado y comportamiento.

Este paquete debe poder protegerse por lo cual tiene propiedades de ocultamiento. Tenemos datos públicos, protegidos o privados y métodos por medio de los cuales operarlos en base a la necesidad.



</ Ejercicios a resolver

Resolver ejercicios 1 y 2 del trabajo práctico N° 1