



OBJETOS (JAVA)

Presentación – Objetos





El Paradigma de Objetos



Paradigmas

¿Qué es un paradigma?

Un Paradigma es un concepto abstracto, la definición de un modelo o patrón en cualquier disciplina física que debe seguirse en determinada situación para resolver problemas.

En programación, el paradigma, es una propuesta tecnológica para resolver problemas, que nos determina cómo van a ser los bloques de construcción de los programas, es decir, los elementos que los constituyen.



Tipos de paradigmas

Imperativo: Los programas se componen de un conjunto de sentencias que cambian su estado. Son secuencias de comandos que ordenan acciones a la computadora.

Declarativo: Opuesto al imperativo. Los programas describen los resultados esperados sin mostrar explícitamente los pasos a llevar a cabo para alcanzarlos.

Lógico: El problema se modela con enunciados de lógica de primer orden.



Tipos de paradigmas

Funcional: Los programas se componen de funciones, es decir, implementaciones de comportamiento que reciben un conjunto de datos de entrada y devuelven un valor de salida.

Orientado a objetos: El comportamiento del programa es llevado a cabo por objetos, entidades que representan elementos del problema a resolver y tienen atributos y comportamiento.



El paradigma de objetos

El paradigma de programación orientada a objetos plantea que todo sistema o proceso informático puede modelarse con Objetos que se encuentren vivos en algún tipo de Ambiente y se relacionan con otros Objetos enviando y recibiendo Mensajes.

Es decir, programar bajo este paradigma, implica que nuestros programas, deberán ser pensados sólo con objetos y mensajes.



El paradigma de objetos

Un objeto, en POO, es un ente completamente abstracto que sabe escuchar, responder y enviar ciertos mensajes.

Un Mensaje es una comunicación dirigida de un objeto a otro. Información enviada desde emisor, hasta el receptor.

El paradigma de objetos tiene otros conceptos que completan este esquema de Objeto-Mensaje, que se irán detallando a lo largo del curso.



El paradigma de objetos

Como todo concepto teórico o abstracto que quiera ser llevado a la vida real o a la práctica, requiere de algún tipo de implementación.

Un paradigma, en tanto que es un concepto teórico, tiene infinitas implementaciones. A la vez, por este mismo motivo, una implementación es una y sólo una instancia posible que responde al Paradigma.



El paradigma de objetos

Cuando hablamos de una “implementación” estamos hablando de un lenguaje de programación orientado a objetos (es decir, que contempla los conceptos teóricos del paradigma orientado a objetos). Cada uno de estos lenguajes implementará cada uno de estos conceptos de forma diferente.

Por eso es importante diferenciar los “conceptos del paradigma” de la “implementación del paradigma”. Es decir, por ejemplo, primero es necesario conocer la definición de un “objeto” según el paradigma, y luego, conocer la implementación de un objeto para determinado lenguaje de programación.



Programación Orientada a Objetos



Programación Orientada a Objetos

OOP (Object Oriented Programming, en inglés) ó POO (Programación orientada a objetos, en español) se basa en el concepto de agrupar código y datos juntos en una misma unidad llamada clase. Este proceso es usualmente referido como encapsulamiento o información oculta.

Estas clases son esencialmente la representación de un conjunto de funciones (llamadas métodos) y variables (llamadas propiedades) designadas para trabajar juntas y proveer una interfaz específica.



Programación Orientada a Objetos

Es importante entender que las clases son una estructura que no puede ser utilizada directamente, éstas deben ser instanciadas en objetos, los cuales podrán así interactuar con el resto de las aplicaciones.

Es posible pensar en clases como un plano para crear un auto, mientras el objeto es de hecho el auto en si mismo, ya que se puede crear en una línea de producción.

Podríamos decir que una clase es una fábrica de objetos.



¿ Por qué Objetos ?

La programación orientada a objetos tiende a ser una simple idea. Esta simple idea es: en lugar de crear estructuras de datos por un lado y código por el otro, la idea es asociar piezas de código junto con los datos.

El mercado de lenguajes de programación, tiende a utilizar este paradigma ya que trae innumerables ventajas con respecto a la programación estructurada.



¿ Por qué Objetos ?

Ventajas:

- **Escalabilidad**: Que un sistema sea escalable, significa que está preparado para poder crecer sin limitaciones. En casos de desarrollo corporativos como por ejemplo el de un sistema bancario, es necesario contar con un lenguaje de programación que permita fácilmente administrar estos tipos de desarrollos empresariales.
- **Mantenimiento**: Tiene una capacidad para realizar cambios en el programa a muy bajo costo (esfuerzo del programador). Gracias a una buena organización de las clases y sus relaciones es posible hacer pequeños pero importantes cambios que impacten luego en todas las partes del sistema.



¿ Por qué Objetos ?

- **Seguridad**: Este paradigma posee lo que se denomina como encapsulamiento y visibilidad, lo cual permite establecer diferentes niveles de seguridad dentro de un equipo de desarrolladores, organizando el trabajo en equipo y limitando la posible existencia de errores o incongruencias cometidos por los programadores.
- **Reutilización**: A través de la correcta utilización de jerarquía de clases, y la redefinición de funciones – polimorfismo, se puede reutilizar un importante porcentaje de código, evitando escribir subprogramas parecidos dentro de un mismo desarrollo.



¿ Por qué Objetos ?

- **Simplicidad**: Cuando el código de un programa desarrollado con lenguaje estructurado, crece indefinidamente, se llega a un punto donde el mismo se vuelve confuso y engorroso. El paradigma de objetos provee simplicidad al poder organizar la complejidad del mismo a través de su estructura de clases y objetos.



Características de la POO

- Abstracción
- Encapsulamiento
- Polimorfismo
- Herencia
- UML



Características de la POO

Abstracción

El proceso de abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real.

La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.

Presentación – Objetos



Características de la POO

Encapsulamiento

Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema.

Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.



Características de la POO

Principio de ocultación

Cada objeto está aislado del exterior, y cada tipo de objeto expone una interfaz a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del objeto pueden acceder a su estado.

Esto asegura que otros objetos no pueden cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas.

Presentación – Objetos



Características de la POO

Polimorfismo

Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando.

O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado.

Presentación – Objetos



Características de la POO

Herencia

Las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación.

Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen.

La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple.

Presentación – Objetos



Características de la POO

UML

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común, semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento.

En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.

UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML. UML guarda una relación directa con el análisis y el diseño orientados a objetos.

Presentación – Objetos



Cómo pensar en Objetos



Cómo pensar en objetos

Pensar en términos de objetos es muy parecido a cómo lo haríamos en la vida real. Por ejemplo vamos a pensar en un auto. Diríamos que el auto es el elemento principal que tiene una serie de características, como podrían ser el color, el modelo o la marca. Además tiene una serie de funcionalidades asociadas, como pueden ser ponerse en marcha, parar o detenerse.

Pues en un esquema OOP el auto sería el objeto, las propiedades serían las características como el color ó el modelo y los métodos serían las funcionalidades asociadas como ponerse en marcha o parar.



Cómo pensar en objetos

Estos objetos se podrán utilizar en los programas, por ejemplo en un programa de concesionarios. Los programas Orientados a objetos utilizan muchos objetos para realizar las acciones que se desean realizar y ellos mismos también son objetos. Es decir, el taller de autos será un objeto que utilizará objetos auto, herramientas, mecánicos, etc.

Los objetos son entidades que combinan estado (atributo), comportamiento (método) e identidad.



Cómo pensar en objetos

Estados

Está compuesto de datos, será uno o varios atributos a los que se habrán asignado unos valores concretos (datos).

Comportamiento

Está definido por los procedimientos o métodos con que puede operar dicho objeto, es decir, qué operaciones se pueden realizar con él.



Cómo pensar en objetos

Identidad

Es una propiedad de un objeto que lo diferencia del resto, dicho con otras palabras, es su identificador. Un objeto contiene toda la información que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases e incluso frente a objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos. A su vez, los objetos disponen de mecanismos de interacción llamados métodos, que favorecen la comunicación entre ellos. Esta comunicación favorece a su vez el cambio de estado en los propios objetos.



Modelado de Objetos



Modelado de objetos

¿Qué es un modelo?

Un modelo es una posibilidad de visualizar a escala o de una manera simulada algo que será construido en la realidad. Es posible decir que antes de construir un edificio se realizan maquetas a escala que representan modelos a seguir, así como también cuando se construye un auto se confeccionan distintos planos o modelos a escala para intentar simular o prever su comportamiento.



Modelado de objetos

¿Qué es un modelo?

Académicamente, es posible definirla como una abstracción de la realidad, es una simplificación de la realidad, con el objetivo final de pasar del modelo a producto real.

Los modelos pueden ser expresados en distintos niveles de precisión, desde algo muy genérico presentando una visión, hasta algo mucho más específico que representa un gran compromiso con el producto a construir.



Modelado de objetos

¿Qué es un objeto?

Un objeto en OOP es un ente completamente abstracto que sabe escuchar, responder, y enviar ciertos mensajes.

Bajando más a informática, podemos definir un objeto como un ente computacional, que está dentro del software, y puede representar o no un ente de la realidad que está fuera de la computadora.

Por ejemplo, un objeto Persona, o un Usuario, puede representar una persona real, en el manejo de usuarios de una aplicación.



Modelado de objetos

¿Qué es un objeto?

Si seguimos con el ejemplo de Persona, una persona puede comer, vestirse, trasladarse, leer, y muchas cosas más. Si revisamos la primera definición, dice que sabe “responder ciertos mensajes” entonces, estos son, los mensajes de los que hablábamos.



Modelado de objetos

¿Quién envía los mensajes?

La respuesta es fácil, otros objetos, recordemos, que en el paradigma de objetos, sólo tenemos objetos y mensajes, y nada más.

¿Qué pasa con los números? ¿También son objetos?

Va a depender del lenguaje. En principio, en un lenguaje orientado a objetos, todo debería ser un objeto, pero esto no es tan así, existen algunos lenguajes que eligen no respetar eso a rajatabla y modelan algunos conceptos muy básicos como tipos primitivos, como por ejemplo los números, los booleanos, los caracteres.



Modelado de objetos

¿Qué es el comportamiento?

Ya dijimos que un objeto era un ente abstracto, que sabía enviar y responder mensajes. Y que nosotros debemos definir cuáles son los mensajes que un objeto debe entender.

El comportamiento, es el conjunto de mensajes que un objeto entiende, y a los que puede o no responder. Lo importante, es que lo entiende.



Modelado de objetos

Con este nuevo concepto, podríamos entonces entender un objeto como: “Un ente computacional que exhibe comportamiento”.

De esta forma, dado un objeto, por ejemplo Persona, uno podría preguntar, ¿cuál es el comportamiento de una Persona? Y ahí uno debe responder con todo lo que el objeto persona sabe hacer. Esto, lo define el programador de ese objeto.

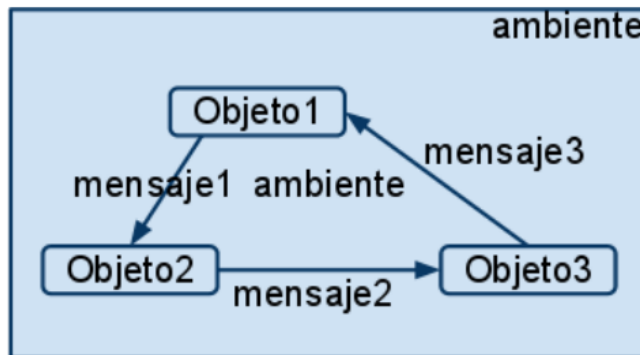
Dicho de otra manera, el “comportamiento” son las cosas que sabe hacer un objeto, el conjunto de mensajes que un objeto le va a poder enviar.



Modelado de objetos

¿Qué es un ambiente?

Los objetos nacen y mueren, o sea “viven”. Viven “dentro de la computadora”, en algún lugar determinado, que es el responsable de contenerlos, manejar su ciclo de vida, dar el contexto para la creación y el envío de mensajes.





El Software Orientado a Objetos



El software orientado a objetos

Con estos conceptos, podemos decir qué es el software. Es el resultado de la interacción de un conjunto de objetos que viven en un ambiente y que se comunican entre sí enviándose mensajes.

Es importante definir software desde el punto de vista de la programación orientada a objetos, ya que esta es la forma que voy a tener de pensar en una programa, en la construcción de un programa.



El software orientado a objetos

Entonces, de ahora en más, para pensar un programa, tenemos que pensar en:

- Qué objetos vamos a incluir en la aplicación
- Qué mensajes va a entender cada objeto
- Cómo van a interactuar los objetos entre sí



Clases

¿Qué es una clase?

Una clase es una plantilla, es un molde que permite construir objetos.

Presenta ideas del mundo real, en forma genérica.

Dentro de un sistema, las clases suelen detectarse como sustantivos en singular.

Poseen atributos y métodos.

Ejemplos de clases: Auto, Empleado, CajaDeAhorro, Alumno.



Detección de clases

Para pensar:

Identificar las clases iniciales necesarias para construir un sistema de gestión bancaria.

Recordar:

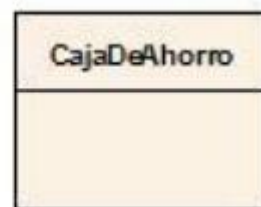
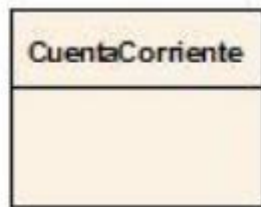
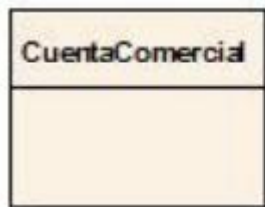
Las clases se detectan como sustantivos en singular.



Detección de clases

Clases iniciales necesarias para construir un sistema de gestión bancaria:

Banco – Sucursal – GrupoFinanciero – Servicio
ClientePyme – ClienteCorporativo – ClienteIndividuo
CuentaComercial – CajaDeAhorro – CuentaCorriente
DirectorGeneral – DirectorRegional - DirectorDeSucursal





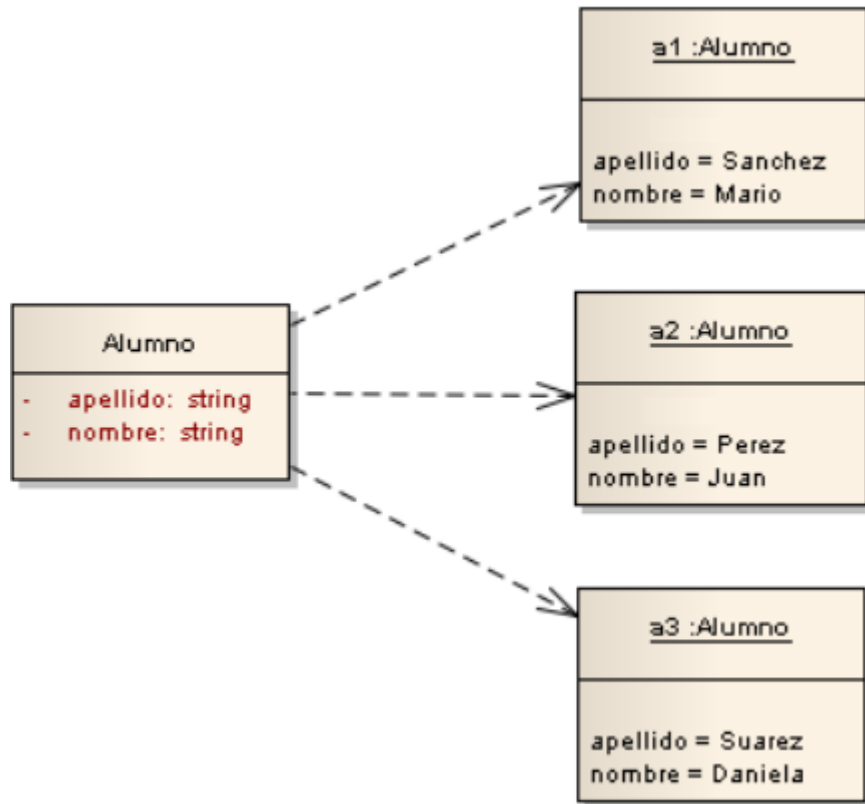
Objetos

¿Qué es un objeto?

Un objeto es una instancia de una clase, podemos decir que el objeto representa algo en particular.

Poseen un estado (de acuerdo a sus atributos).

Poseen un comportamiento (realizan operaciones de acuerdo a sus métodos).





Atributos

¿Qué son los atributos?

Son características que posee una clase.

Son variables contenidas y establecidas por los objetos, y normalmente cuentan con un tipo de dato asociado.

Los atributos de una clase definen las características de sus objetos.

Las clases definen los atributos, y los objetos “los completan”.

Automovil
color modelo precio usado



Detección de atributos

Para pensar:

A partir de las clases detectadas anteriormente, identificar los atributos que considere necesarios en cada clase.

Cada atributo deberá tener establecido un tipo de dato. Identificar al menos tres atributos por clase.

Recordar:

Los atributos son características de una clase.



DetECCIÓN DE ATRIBUTOS

Posible solución:

Sucursal: nombre, numero, direccion, cantidadDeEmpleados, cantidadDeClientes

ClientePyme: razonSocial, direccion, fechaDeAlta, cuentaCorriente

ClienteCorporativo: razonSocial, direccion, fechaDeAlta, cuentaCorriente

ClienteIndividuo: nombre, apellido, direccion, fechaDeAlta, cajaDeAhorro

Servicio: nombre, descripcion, fechaDeAlta

GrupoFinanciero: nombre, descripcion, fechaDeAlta

CajaDeAhorro: moneda, numero, cbu, saldo

CuentaCorriente: moneda, numero, cbu, saldo, giroEnDescubierto

Presentación – Objetos



Detección de atributos

Posible solución:

DirectorGeneral: nombre, apellido, fechaDeNacimiento, fechaDeIngreso

DirectorRegional: nombre, apellido, fechaDeNacimiento, fechaDeIngreso

DirectorDeSucursal: nombre, apellido, fechaDeNacimiento, fechaDeIngreso

Banco
<ul style="list-style-type: none">- nombre: String- cantidadDeEmpleados: int- cantidadDeSucursales: int- fechaDeConstitucion: Date



Operaciones

Las operaciones son acciones contenidas en una clase, y definen su comportamiento.

Dentro de un sistema, las operaciones suelen detectarse como verbos.

Desde la perspectiva de Diseño y Programación, se denominan Métodos.

Desde la perspectiva de Análisis, se denominan Operaciones.

Puede tener opcionalmente valores de entrada (Parámetros) y valores de salida (Valores de Retorno).

Procedimientos (no retornan un valor) vs. Funciones (retornan un valor).

Auto
color: string
encender() acelerar() frenar()



Operaciones

¿Qué es un valor de entrada o parámetro?

Los parámetros son valores enviados a una operación.

La operación toma los parámetros como valores de entrada, y así puede realizar las acciones necesarias.

Todos los parámetros deben tener un tipo de dato asociado.

- Método encender() → sin parámetros
- Método acelerar(int) → recibe como parámetro la cantidad de “Kms.” a acelerar
- Método frenar(int) → recibe como parámetro la cantidad de “Kms.” que debe bajar de velocidad.

Auto
color: string
encender() acelerar(int) frenar(int)



Operaciones

¿Qué es un valor de salida o retorno?

El valor de salida de una operación es un valor retornado por la operación luego de realizar cierto procesamiento.

Los valores de entrada son datos, y los valores de salida son considerados información.

Todos los valores de salida deben tener un tipo de dato asociado.

Es posible retornar un único valor de salida.

Presentación – Objetos



Detección de operaciones

Para pensar:

A partir de las clases y los atributos detectados en los ejercicios anteriores, identificar operaciones que podría realizar cada clase.

Recordar:

Las operaciones son verbos.



Detección de operaciones

Posible solución:

Banco: informarCantidadDeEmpleados(),informarCantidadDeSucursales(),
informarSucursales(), calcularFacturacionAnual(), calcularFacturacionMensual(),
informarClientes()

Sucursal: informarDireccion(),informarCantidadDeEmpleados(),
calcularFacturacionAnual(), calcularFacturacionMensual(), informarClientes(),
informarDatosDeCuenta(nroDeCuenta)

ClientePyme: informarMovimientosEnCuentas(), informarDatos()

ClienteCorporativo: informarMovimientosEnCuentas(), informarDatos()

ClienteIndividuo: informarMovimientosEnCuentas(), informarDatos()

Presentación – Objetos



Detección de operaciones

Posible solución:

CajaDeAhorro: informarSaldo(), extraerDinero(monto), depositarDinero(monto)

Servicio: informarNombre(), informarDescripcion()

GrupoFinanciero: informarFechaDeAlta(), informarDescripcion()

DirectorGeneral: informarDatos()

DirectorRegional: informarDatos()

DirectorDeSucursal: informarDatos()



Sobrecarga de operaciones

¿Qué es?

Es la aparición de métodos dentro de una misma clase que se llaman igual, pero realizan acciones (levemente) diferentes.

Normalmente varían en cantidad y/o tipo de parámetros.

Por ejemplo, en la clase Auto podríamos encontrar variantes del método `acelerar()` :

- **`acelerar()`** -> acelera 10km/h
- **`acelerar(int km)`** -> acelera de acuerdo al parametro “km”
- **`acelerar(int km, boolean tieneNitro)`** -> ídem caso anterior, pero si el parámetro “tieneNitro” es verdadero acelera el doble

Presentación – Objetos



Visibilidad

¿Qué es la visibilidad?

Es la posibilidad de “ver” un atributo o método.

Si un atributo o método es privado (-) -> solo puede verse dentro de la clase.

Si un atributo o método es publico (+) -> puede verse desde otras clases.

La visibilidad es establecida por los modificadores de visibilidad: **private** y **public**.



Encapsulamiento

¿Qué es el encapsulamiento?

Es el ocultamiento del estado de un objeto.

El estado (atributos) podrá accederse únicamente a través de sus operaciones (métodos).

En la clase, los atributos deben ser privados y los métodos para acceder a los atributos deben ser públicos.



Setters y getters

¿Qué son los setters y getters?

Son métodos de acceso públicos a atributos privados.

Representan la única forma de acceder a los atributos.

setter -> método utilizado para setear un valor a un atributo

getter -> método utilizado para obtener un valor de un atributo

Los IDEs generalmente permiten generar los setters y los getters de forma automática.

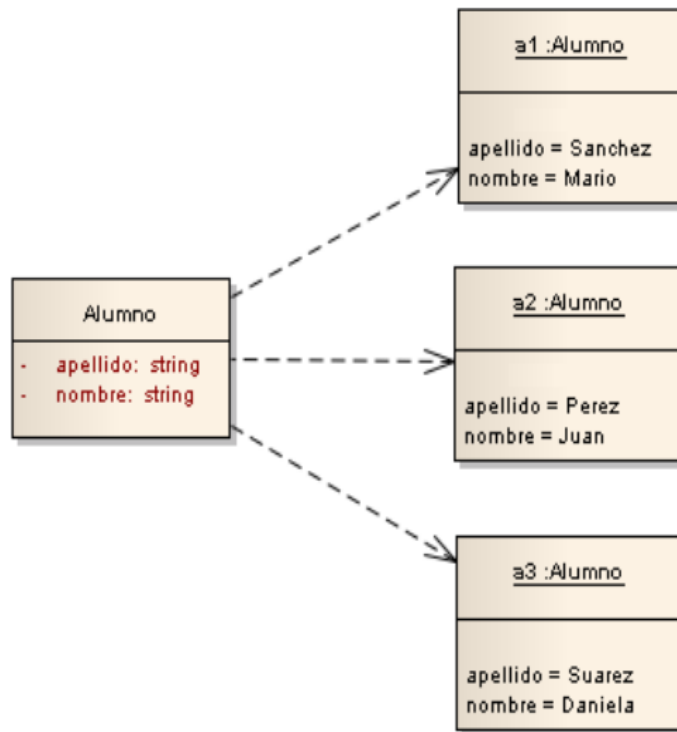


Constructores

Objetos vs. clases

La **clase** representa un concepto, es un molde, una plantilla.

Los **objetos** representan instancias de una clase. Sería como tomar una plantilla (una clase) y personalizarla (completar sus atributos).





Constructores

Constructores para construir objetos

Los objetos son construidos a partir de una clase. Todos los objetos dependen de una clase.

Para construir un objeto es necesario utilizar un constructor.

El constructor es “un método” de la clase que se invoca al construir un objeto, y en su interior tiene un conjunto de acciones a realizar.



Constructores

Conceptos avanzados

¿Cómo se llama cuando una clase tiene más de un constructor? **Sobrecarga de constructores.**

Una clase debe tener al menos un constructor. Si no se agrega un constructor, normalmente se asume que posee el constructor vacío.

Una clase puede tener todos los constructores que sean necesarios.

Presentación – Objetos



Constructores

Conceptos avanzados

¿Cómo se llama cuando una clase tiene más de un constructor? **Sobrecarga de constructores.**

Una clase debe tener al menos un constructor. Si no se agrega un constructor, normalmente se asume que posee el constructor vacío.

Una clase puede tener todos los constructores que sean necesarios.

Presentación – Objetos