ENTERPRISE JAVA DEVELOPER

JAVA ORIENTADO A OBJETOS

INTRODUCCIÓN

Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

Temas

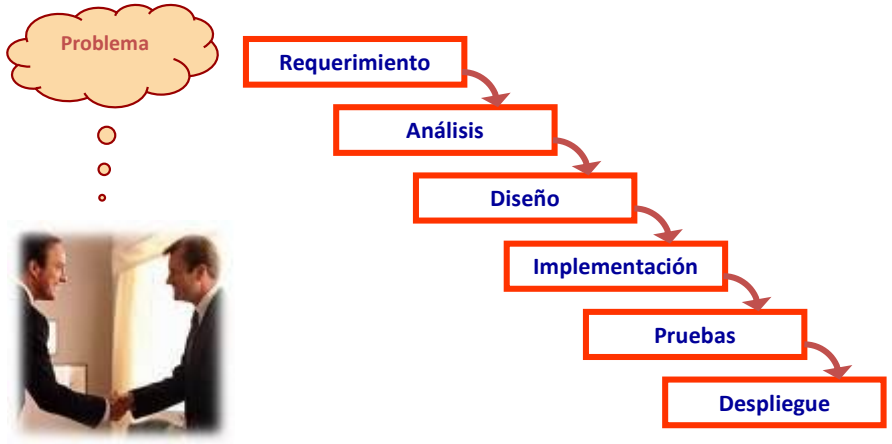
- Historia
- Ciclo de Vida del Software
- Metodologías
- Guerra de Métodos
- Metodologías Ágiles
- Abstracción
- Clase
- Objeto
- Encapsulamiento
- Herencia
- Polimorfismo
- Mensaje
- Conclusiones

Historia

- **Simula 1964:** Lenguaje de simulación, ha influido en el desarrollo de otros LPOO.
- **SmallTalk 1972:** LPOO puro, todo es un objeto, los objetos se comunican mediante mensajes, todo objeto se instancia de una clase, las clases se organizan en jerarquía (herencia).
- **C++ 1985:** LPOO híbrido nace del lenguaje estructurado C.
- **Object Pascal (Delphi) 1985:** LPOO híbrido nace del lenguaje estructurado Pascal.
- **PHP 1994:** Exclusivamente desarrollo Web.
- **Java 1995:** Centrado en la Web.
- **C# 2000:** Nuevo lenguaje basado en la tecnología .NET
- **VB.NET 2000:** LPOO híbrido nace del lenguaje VB.

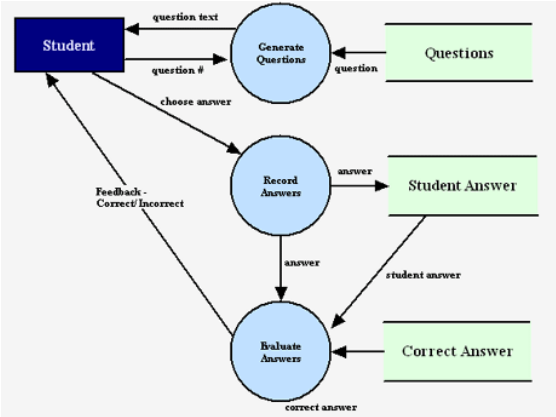
Ciclo de Vida del Software

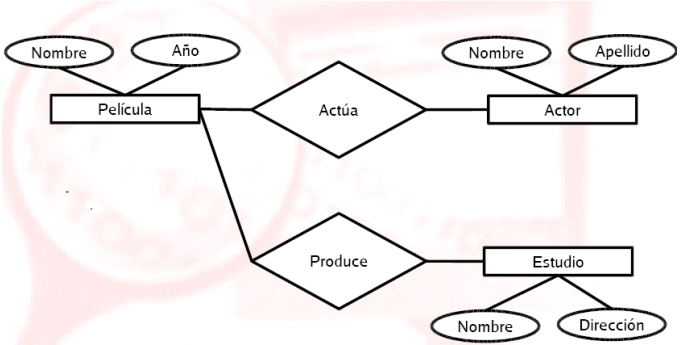
- **Requerimiento:** Concepción de un problema a resolver
- **Análisis:** ¿Qué? (entender el problema / cuestionarios y preguntar)
- **Diseño:** ¿Como? (Resolver el problema / prototipos y modelos)
- **Implementación:** ¿Hacerlo? (Codificación / Programarlo)
- **Pruebas:** ¿Funciona? (Verificar / Comprobar)
- **Despliegue:** ¿Instalar? (Distribuir el software)



Metodologías

- **Metodología Estructurada:** Orientada a los procesos, Metodología antigua, usa lenguajes estructurados (Pascal, Foxpro, C, etc.), exponentes de esta metodología (Yourdon, DeMarco, Gane y Sarson), Notación utilizada (DFD, ER, etc.)





Metodologías

- **Metodología Orientada a Objetos:** Orientada a los objeto, Metodología moderna, usa lenguajes de programación orientados a objetos (C#, Java, PHP, etc), exponentes de esta metodología (Booch (OOAD), Jacobson (OOSE), Rumbaugh (OMT), nace RUP, MSF, XP, Notación estándar UML (estandarizado por OMG en 1997)

Diagrama de Caso de Uso

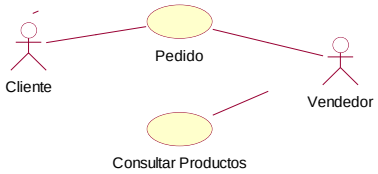


Diagrama de Clases

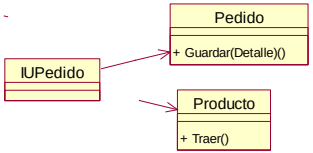
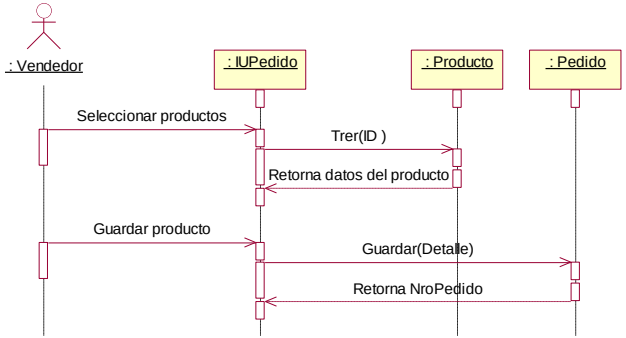


Diagrama de Secuencia



Guerra de Métodos

- Entre los años 80 e inicio de los 90, las metodología orientada a objetos comenzaba a madurar como un enfoque de desarrollo de software.

- Empezaron a surgir diferentes métodos de análisis y diseño orientado a objetos, entre los que destacan los métodos Booch, OOSE (Object Oriented Software Engineering) , OMT (Object Modeling Technique), Taylor, Coad /Yourdon entre otros.

- En julio de 1993, Jacobson criticó en lo que en él denominaba “La Guerra de los métodos” y planteo la necesidad de llegar a una notación estándar de modelado, para evitar la confusión reinante y favorecer el uso de los métodos de software.

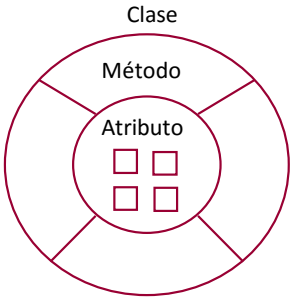
- Para poner fin a “La Guerra de métodos" que se presentó en ese momento, se creó el Lenguaje Unificado de Modelado (UML).

Guerra de Métodos

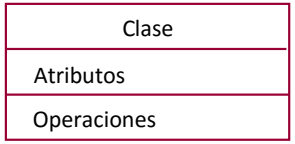
Notaciones OO

BOOCH'93

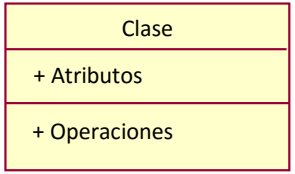
TAYLOR



OMT (Rumbaugh)



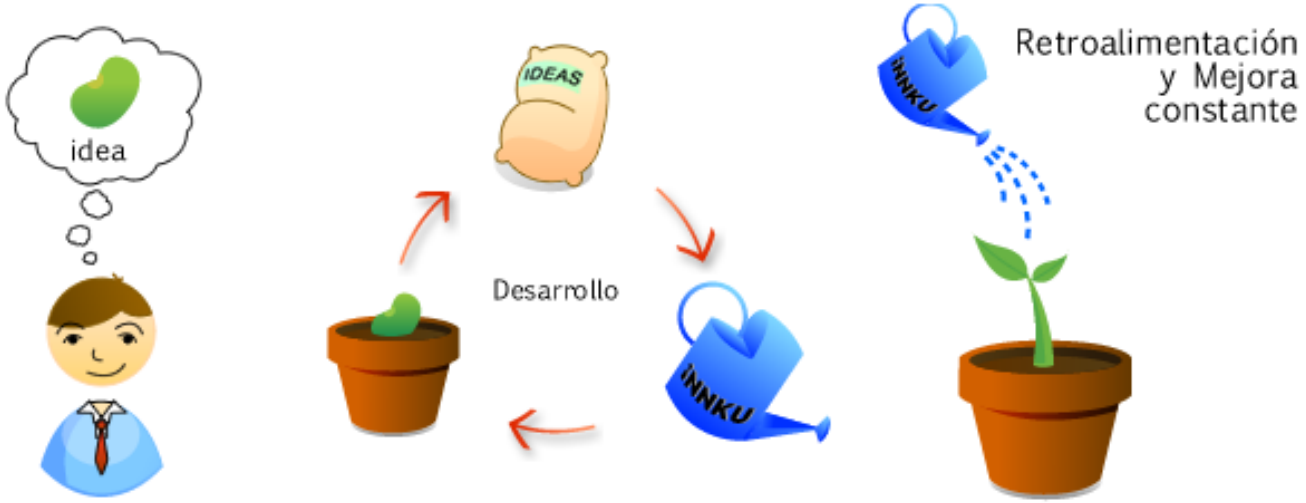
Notación estándar UML



Eric Gustavo Coronel Castillo gcoronelc@gmail.com gcoronelc.blogspot.com

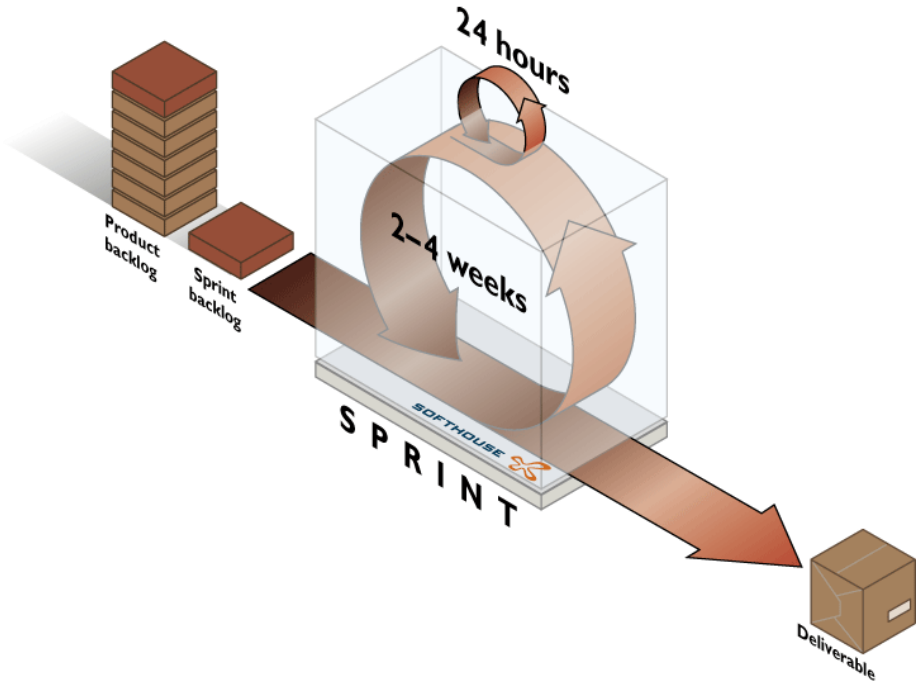
Metodología Ágil

- Está preparada para adecuarse rápidamente a cambios y mejoras.
- Significa trabajar muy cerca del cliente, realizar entregas que el cliente pueda utilizar en cortos periodos de tiempo y recibir retroalimentación constante.



Metodología Ágil

SCRUM



Metodología Ágil

KANBAN

Abstracción

- Consiste en capturar, percibir y clasificar las características (datos-atributos) y comportamientos (operaciones) necesarias (relevantes) del mundo real (proceso a sistematizar) para dar solución al problema.

Notación UML

Abstracción

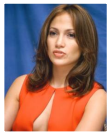
Persona
+ Nombre : String + Edad : Integer + Profesion : String
+ Caminar() + Correr() + Cantar() : String

Animal
+ Raza : String + Genero : String
+ Comer()

Transporte
+ Tipo : String + Marca : String + Año : Integer
+ Encender() : Boolean + Acelerar(Velocidad : Integer)

Clase

- Es la clasificación de las características y comportamientos comunes de objetos del mismo tipo.
- En la POO se dice que es la plantilla genérica para un conjunto de objetos con las mismas características.



Clasificación

Notación UML

CLASE

Persona
+ Nombre : String + Edad : Integer + Profesion : String
+ Caminar() + Correr() + Cantar() : String

← Nombre de la clase

← Atributos

← Operaciones

Objeto

- Es la representación concreta y detallada de algo en particular, tal representación determina su identidad (nombre único para distinguir un objeto de otro), su estado (conjunto de valores que caracterizan al objeto en un momento dado) y su comportamiento (conjunto de funciones que el objeto puede llevar a cabo).
- Los objetos son instancias de clases (una instancia es un objeto)

Notación UML OBJETO

objPersona1 : Persona

+ Nombre : Jennifer
+ Edad : 32
+ Profesion : Cantante

+ Caminar()
+ Correr()
+ Cantar():String

objPersona2 : Persona

+ Nombre : Zidane
+ Edad : 36
+ Profesion : Futbolista

+ Caminar()
+ Correr()
+ Cantar():String

Instancia

Instancia

Notación UML CLASE

Persona

+ Nombre : String
+ Edad : Integer
+ Profesion : String

+ Caminar()
+ Correr()
+ Cantar() : String

Encapsulamiento

- Oculta la complejidad, es considerada como la caja negra, solo se conoce el comportamiento pero no su detalle interno.
- En la POO nos interesa saber que hace un objeto y no como lo hace.

Persona
+ Nombre : String + Edad : Integer + Profesion : String
+ Caminar() + Correr() + Cantar() : String

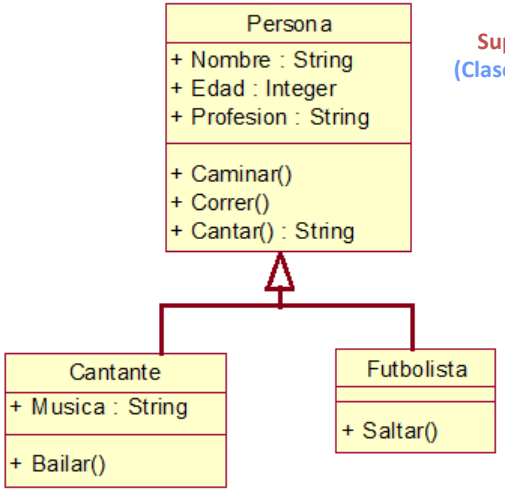
Correr
Implementa su propia técnica para correr y se desconoce como lo hace

Herencia

- Es la característica mas representativa de la POO, y permite reutilizar objetos para construir nuevos objetos.
- En la herencia se observa que existen clases genéricas (padre/superclase) que agrupan características y comportamientos similares para un conjunto de objetos y clases derivadas (hija/subclase) que extienden o redefinen la clase genérica.

Notación UML

GEENERAZALIZACIÓN o ESPECIALIZACIÓN (HERENCIA)



SuperClase (Clase genérica)

SubClase (Clase especializada)

SubClase (Clase especializada)

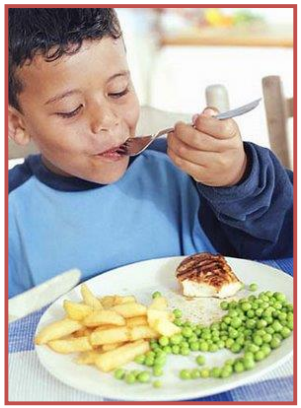
Polimorfismo

- Es la capacidad que tienen los objetos de responder al mismo mensaje de diferente formas.
- Operaciones con el mismo nombre pero con implementación diferente.

Persona
comer() descansar()

Animal
comer() descansar()

comer ( )



comer ( )

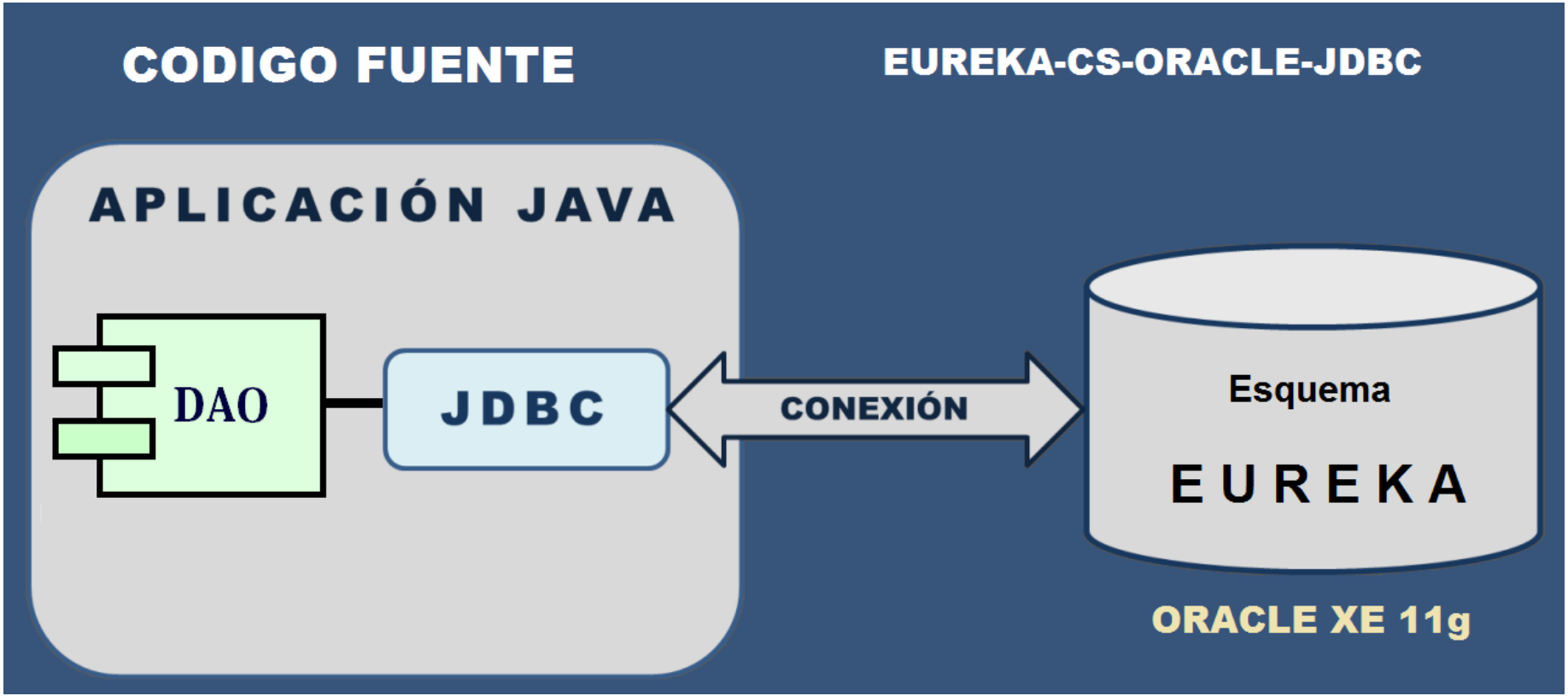
Mensaje

- Es la invocación de un comportamiento (operación) que incorpora el objeto.
- Se dice que el objeto es inútil si esta aislado.
- El medio para que un objeto interactué con otro, es el envió de mensajes.

Conclusiones

- La POO permite reutilizar funcionalidades (código) y hace más fácil el mantenimiento del código.
- Una aplicación usará varios objetos para cumplir su objetivo, es decir es un rompecabezas de objetos.
- Los objetos pueden reutilizarse en muchas aplicaciones.
- Para modelar, construir, documentar se utiliza una notación estándar llamada UML, que todo desarrollador debe entender (Leer e Interpretar) para programar en un lenguaje de Programación Orientado a Objetos.

Eric Gustavo Coronel Castillo gcoronelc@gmail.com gcoronelc.blogspot.com



Dirección de descarga: https://goo.gl/TDgc5R

JAVA ORIENTADO A OBJETOS

Gracias

Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com