

Patrones



Maestría en Ingeniería de Software



Agenda

- Model Driven Architecture (MDA)
- Unified Model Language (UML)
- Object Constraint Language (OCL)
- **Patrones**
- Conclusiones

Contenido

Patrones

- Introducción
 - Definición
 - Características
 - Elementos para su descripción
 - Categorías
- Patrones arquitectónicos
 - Características
 - Ejemplo

Contenido

Patrones

- Patrones de Diseño
 - Definición
 - Clasificación Gof
 - Algunos patrones Gof
- Patrones de Análisis
 - Características
 - Ejemplo

Contenido

Patrones

- Idiomas
 - Características
 - Ejemplo
- Conceptos asociados
 - Antipatrones
 - Refactoring
- Patrones en MDA

Introducción a Patrones

“ Un patrón describe primero un problema que ocurre una y otra vez en nuestro entorno, y a continuación describe el núcleo de la solución al problema, de tal manera que puede usar la solución millones de veces sin repetir la solución específica una sola vez ”

[Fuente: Christopher Alexander en “Timeless Way of Building,”1979]

Introducción a Patrones

Definición

Problema de diseño particular que surge en contextos de diseños específicos y presenta un esquema genérico bien probado por su solución. El esquema de la solución es especificado describiendo sus componentes constituyentes, sus responsabilidades y relaciones, y la forma en la cual ellos colaboran.

[Fuente: Buschmann et al. En “*Pattern-Oriented Software Architecture*,”]

Introducción a Patrones

Características

- Es una solución reusable de un problema recurrente
- Testeada
- Considera la solución como un template
- Puede ser adaptada y personalizada para un dominio específico

Introducción a Patrones

Elementos para Descripción

- Nombre
- Problema
- Contexto
- Efecto
- Solución
- Ejemplos
- Contexto resultante
- Patrones relacionados
- Casos conocidos
- Representación gráfica

Introducción a Patrones

Idiomas: patrones de bajo nivel específico a un lenguaje de programación; tales como Java y C++ (Coplien, 1992).

Patrones de Diseño: se enfoca en un pequeño grupo de de objetos que colaboran entre sí. Gamma, Helm, Johnson, Vlissides (1995)

Patrones Arquitectónicos: se enfoca en la estructura de subsistemas de un sistema. Buschmann Meunier, Rohnert, Sommerlad, Stal (1996)

Patrones de Análisis: se enfoca en patrones encontrados durante el análisis OO. Fowler (2002)

Patrones Arquitectónicos

Idiomas: patrones de bajo nivel específico a un lenguaje de programación; tales como Java y C++

Patrones de Diseño: se enfoca en un pequeño grupo de de objetos que colaboran entre sí. Gamma, Helm, Johnson, Vlissides (1995)

Patrones Arquitectónicos: se enfoca en la estructura de subsistemas de un sistema. Buschmann, Meunier, Rohnert, Sommerlad, Stal (1996)

Patrones de Análisis: se enfoca en patrones encontrados durante el análisis OO. Fowler (2002)

.

Patrones Arquitectónicos

Características

- Estructuras de alto nivel para sistemas de software
- Fija la arquitectura global de una aplicación
- Contiene un cjto. de subsistemas predefinidos
- Define las responsabilidades de c/u de los subsistemas
- Detalla en las relaciones entre los subsistemas
- Posteriormente se hará uso de patrones de diseño para problemas específicos.



Patrones Arquitectónicos

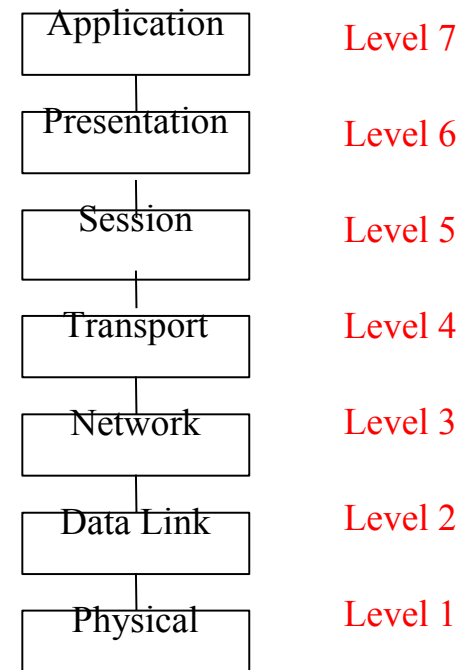
Patrones arquitectónicos

- **Layer**
- **MVC**
- Presentation-Abstraction-Control (PAC)
- Microkernel
- Reflection
- Pipes y Filters
- Backboard
- Broker

Patrones Arquitectónicos

Layers

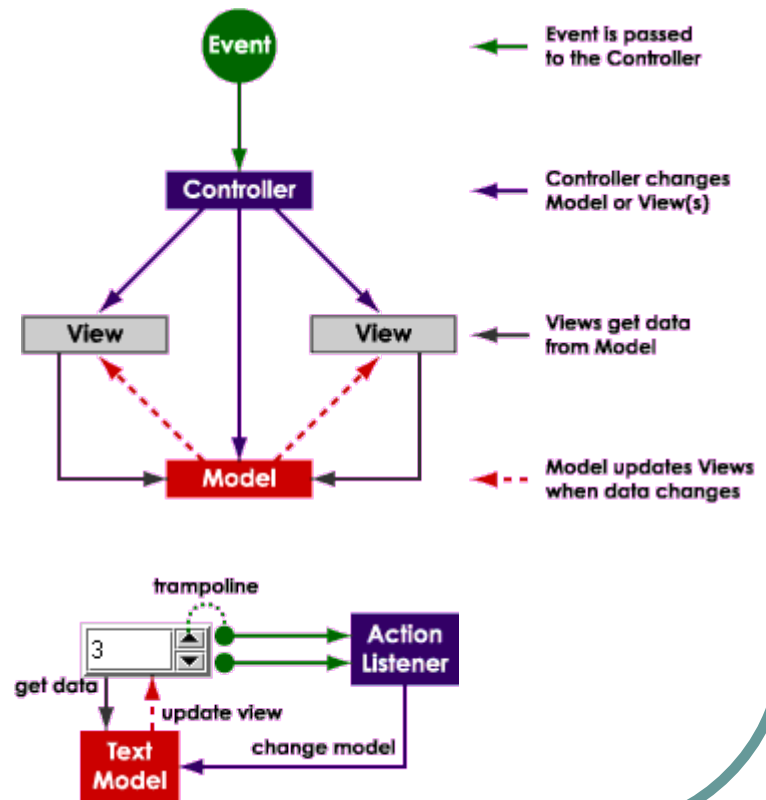
- ayuda a estructurar aplicaciones que pueden ser descompuestas en grupos de subtareas
- subtareas agrupadas en distintos niveles de abstracción



Patrones Arquitectónicos

Model-View-Controller (MVC)

- divide una aplicación interactiva en 3 componentes: modelos, vistas y componentes.
- el modelo contiene la funcionalidad central y datos
- las vistas muestran la información al usuario
- los controladores manejan la entrada de usuario
- las vistas y los controladores se encargan de la interfase de usuario



Patrones Arquitectónicos

Idiomas: patrones de bajo nivel específico a un lenguaje de programación; tales como Java y C++

Patrones de Diseño: se enfoca en un pequeño grupo de de objetos que colaboran entre sí. Gamma, Helm, Johnson, Vlissides (1995)

Patrones Arquitectónicos: se enfoca en la estructura de subsistemas de un sistema. Buschmann, Meunier, Rohnert, Sommerlad, Stal (1996)

Patrones de Análisis: se enfoca en patrones encontrados durante el análisis OO. Fowler (2002)

.

Patrones de Diseño

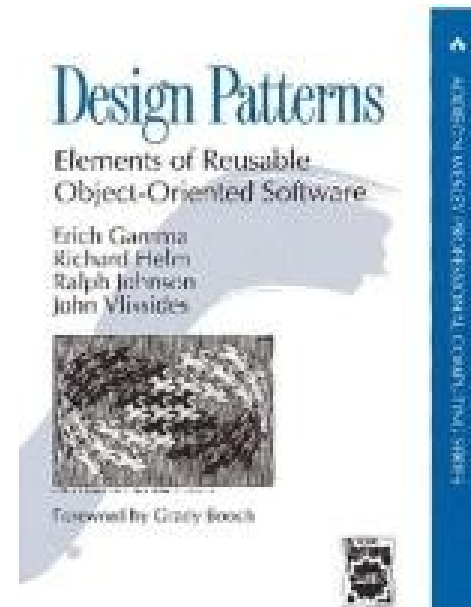
Ayudan a los desarrolladores de software a resolver problemas de diseño OO que repetidamente ocurren. Se reutilizan diseños exitosos basándose en la experiencia previa.

Características

- Nivel de abstracción medio independiente de la implementación
- Diseñado para “microarquitecturas”
- Sencillos, flexibles
- Extensibles y reusables

Permiten

- Incrementar vocabulario de diseño
- Mejorar la documentación y mantenimiento de sistemas
- Acelerar proceso de diseño
- Facilitar aprendizaje de buenas prácticas



Patrones de Diseño

Definición

“ descripciones de objetos que se comunican y clases que son personalizadas para resolver un problema de diseño general en un contexto particular ”

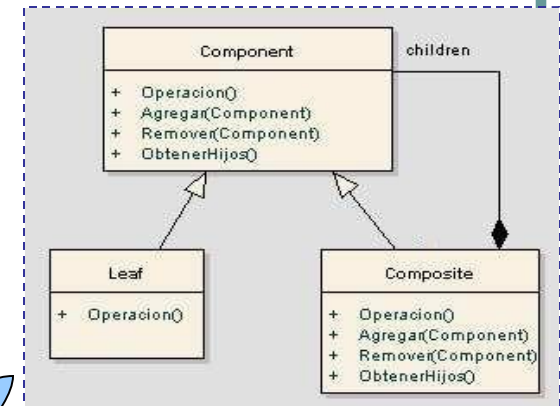
(Gamma y otros, 1995)

Patrones de Diseño

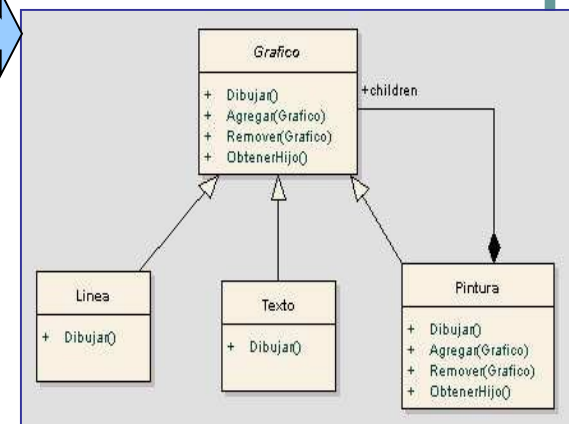
Clasificación Gof

		PROPOSITO		
		Creacional	Estructural	Comportamiento
ALCANCE	Clase	- Factory Method	- Adapter	- Interpreter - Template Method
	Objeto	- Abstract Factory - Builder - Prototype - Singleton	- Bridge - Composite - Decorator - Façade - Flyweight - Proxy	- Chain of Responsibility - Command - Iterator - Mediator - Memento - Observer - State - Strategy - Visitor

Patrón

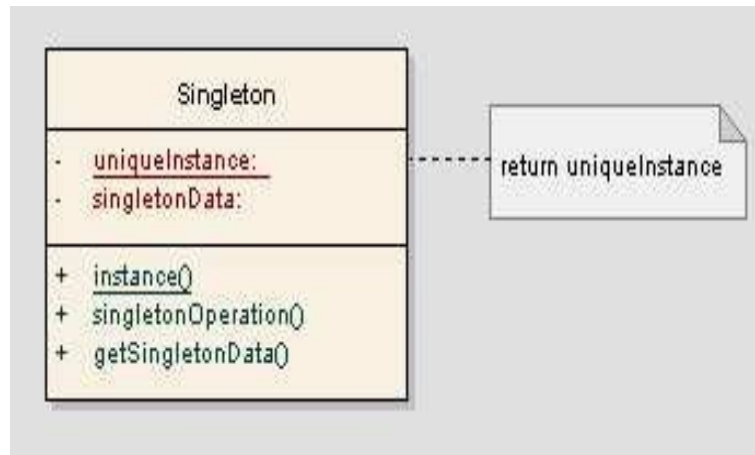


Diseño particular



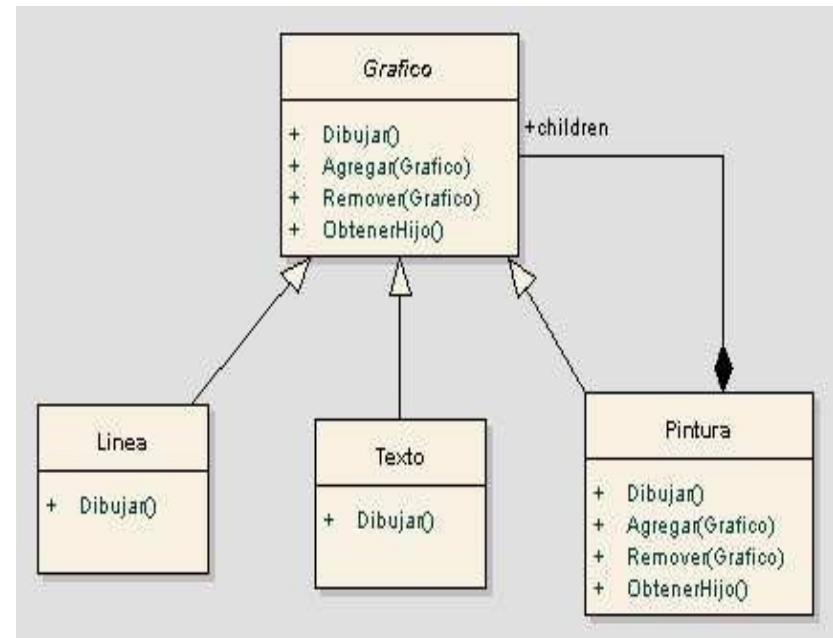
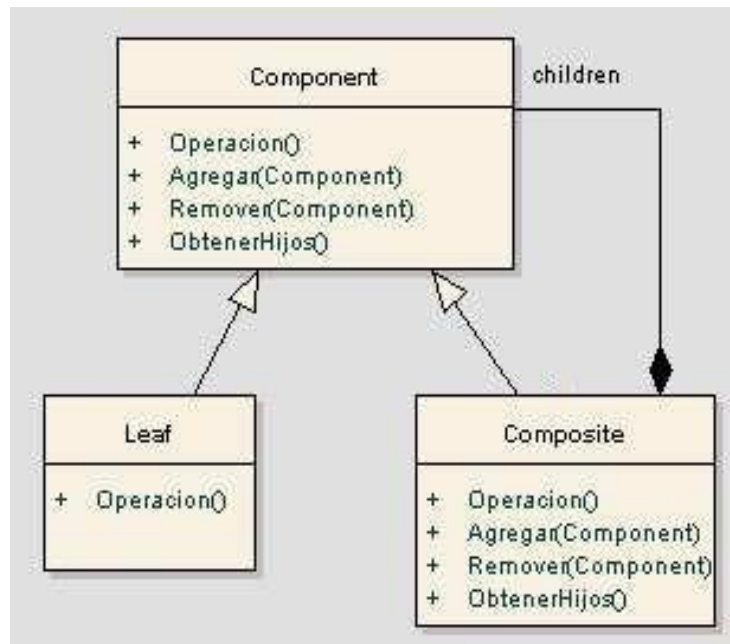
Patrones de Diseño

Singleton



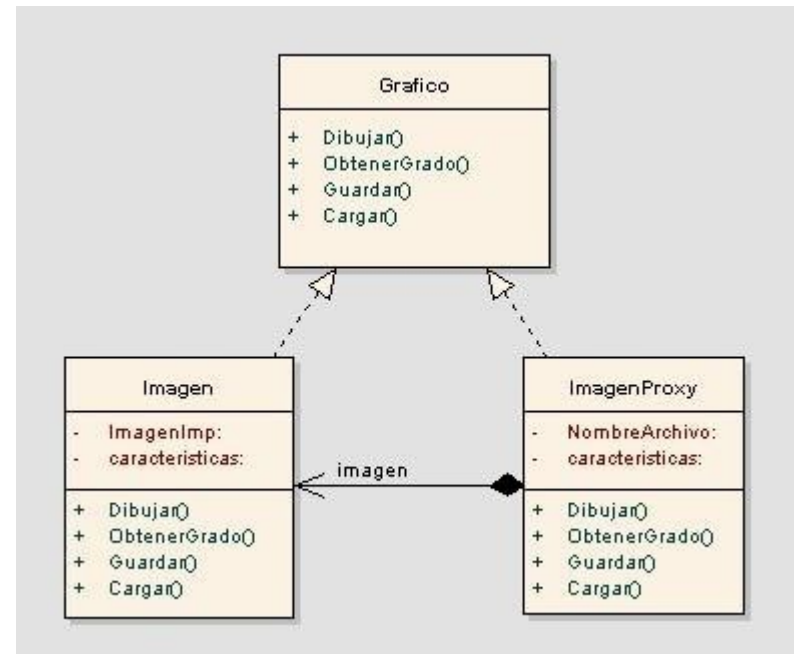
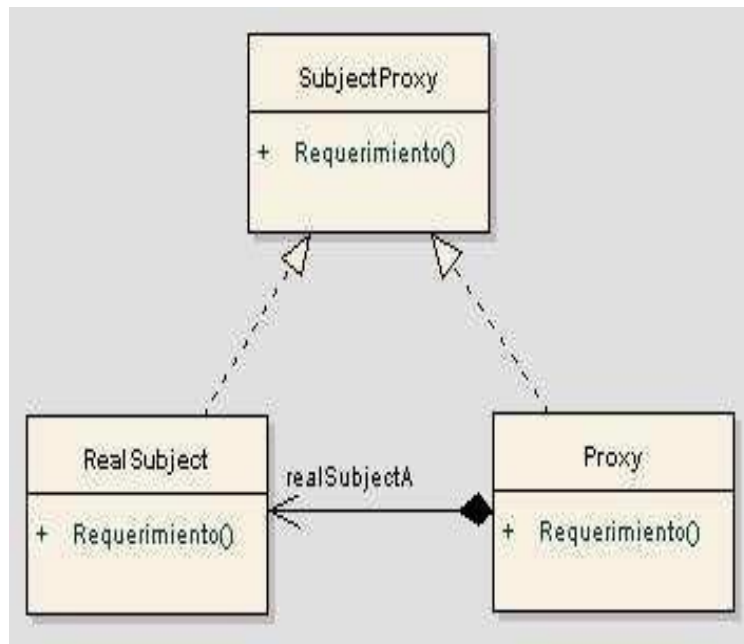
Patrones de Diseño

Composite



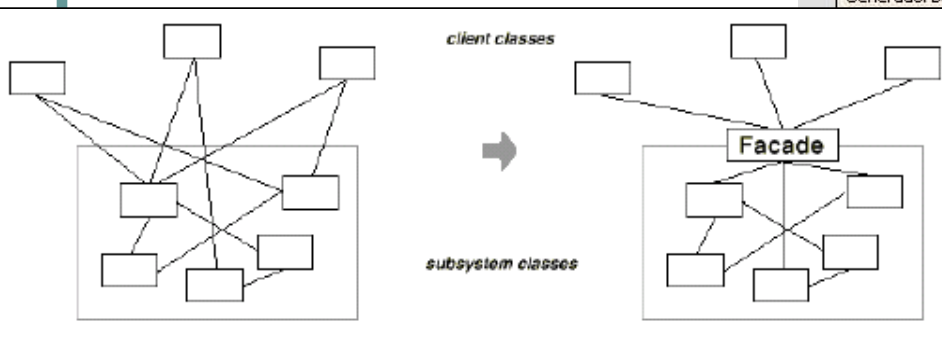
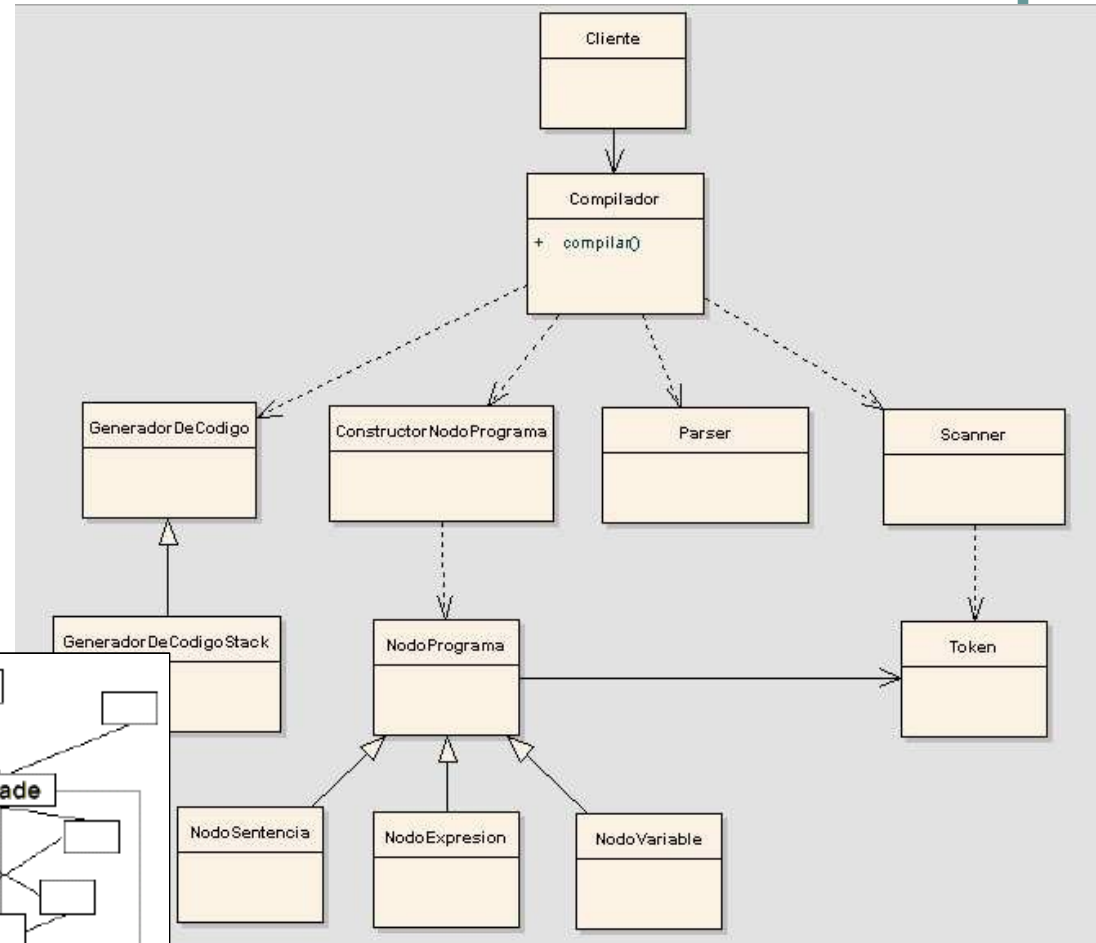
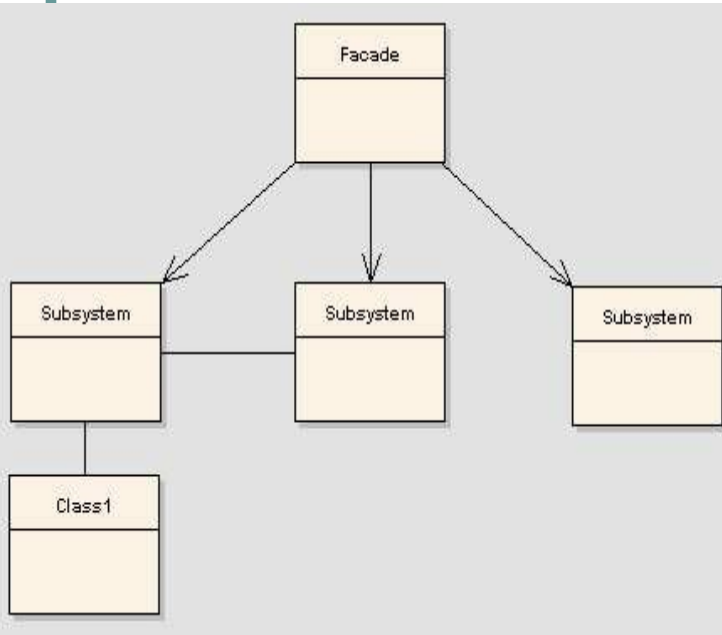
Patrones de Diseño

Proxy



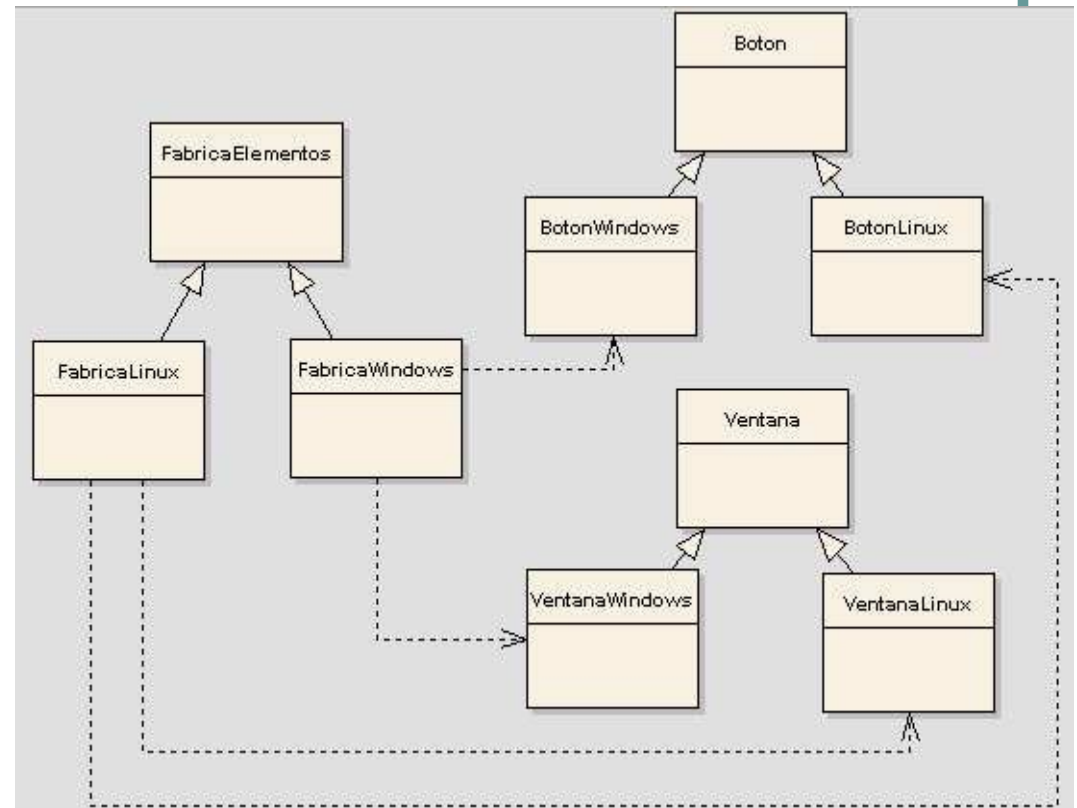
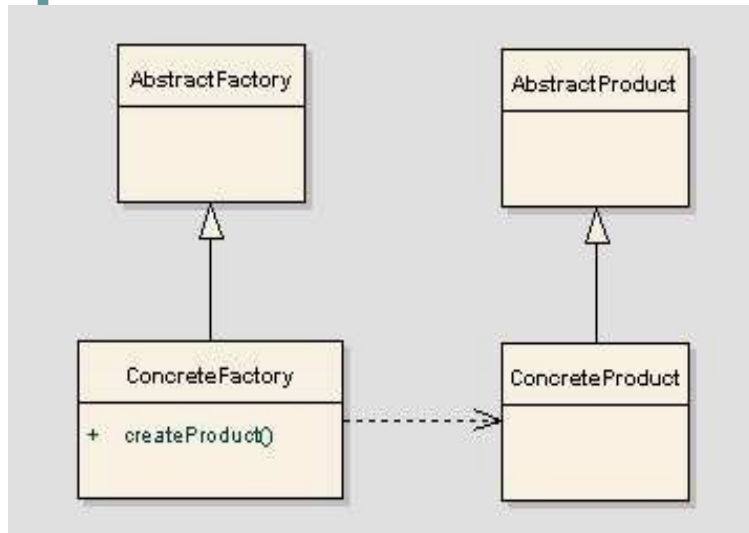
Patrones de Diseño

Facade.



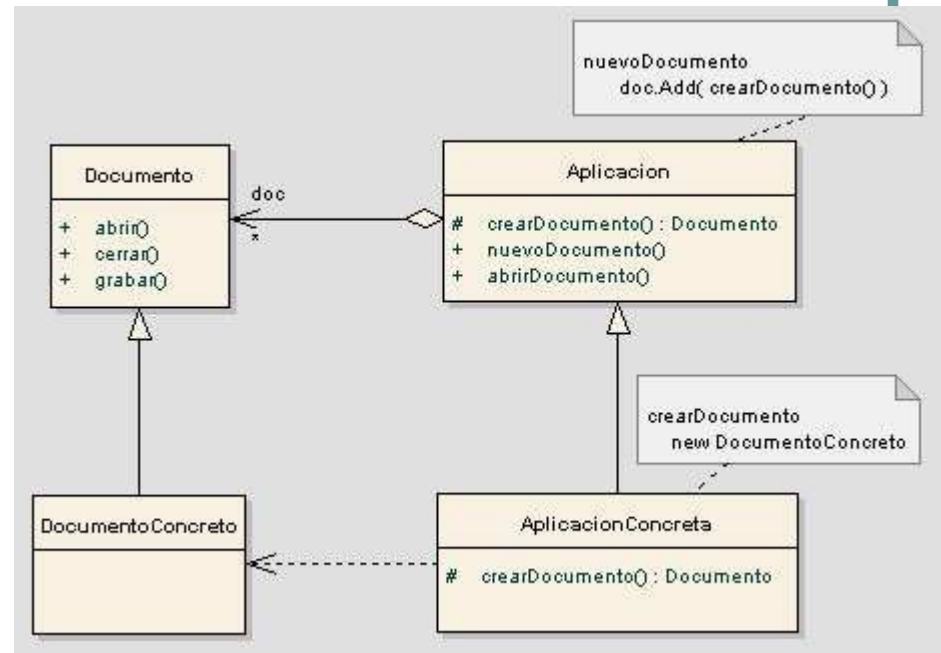
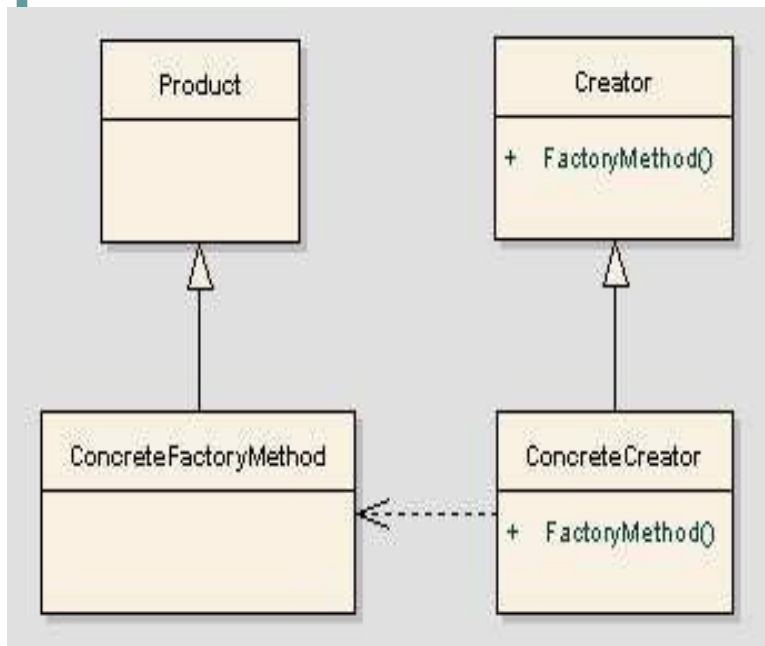
Patrones de Diseño

Abstract Factory



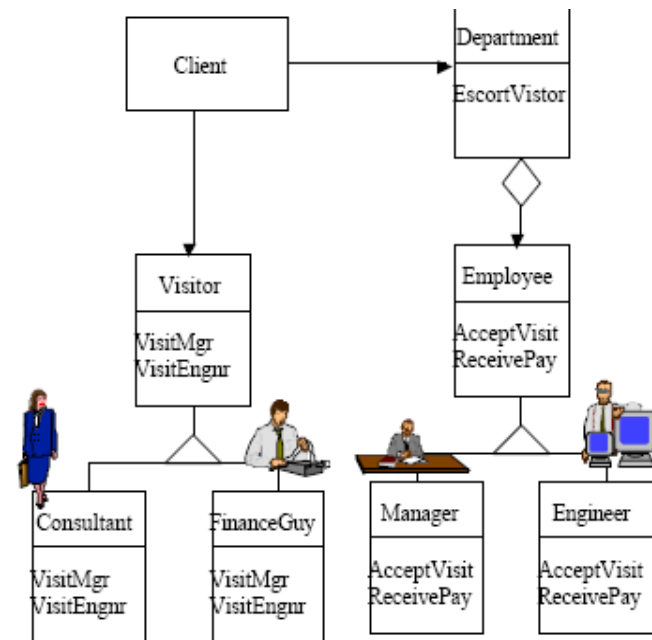
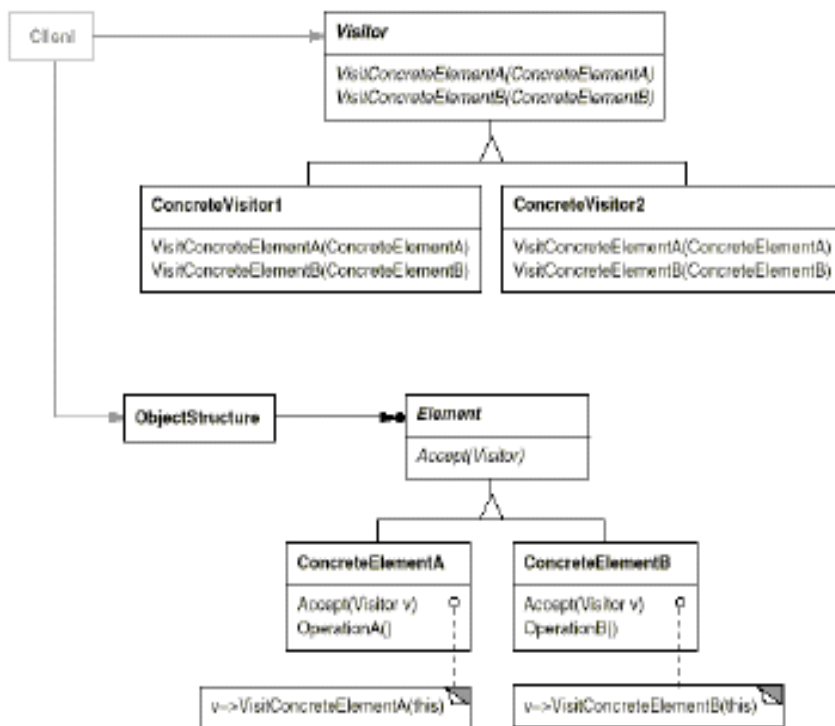
Patrones de Diseño

Factory Method



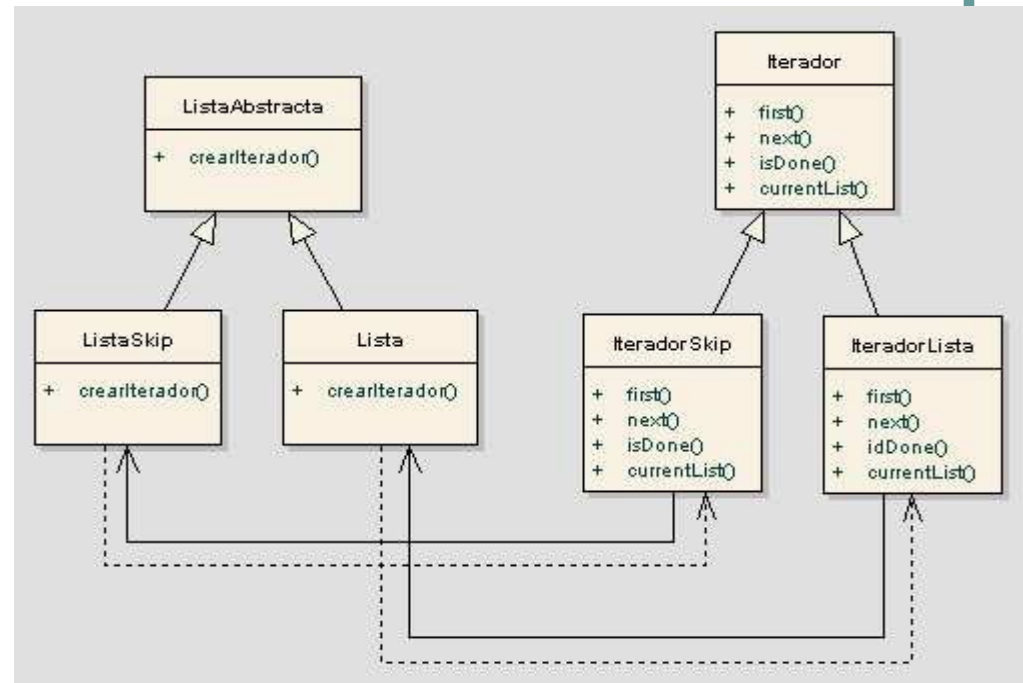
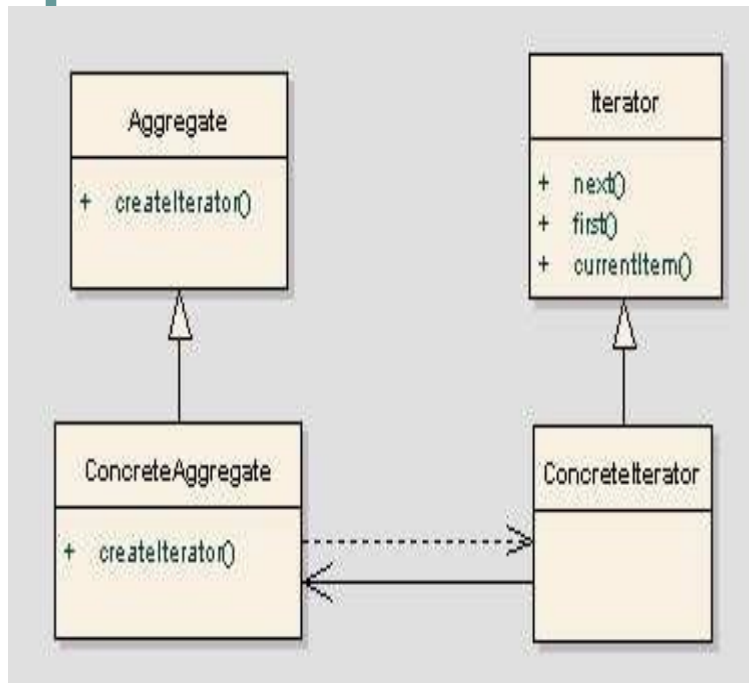
Patrones de Diseño

Visitor



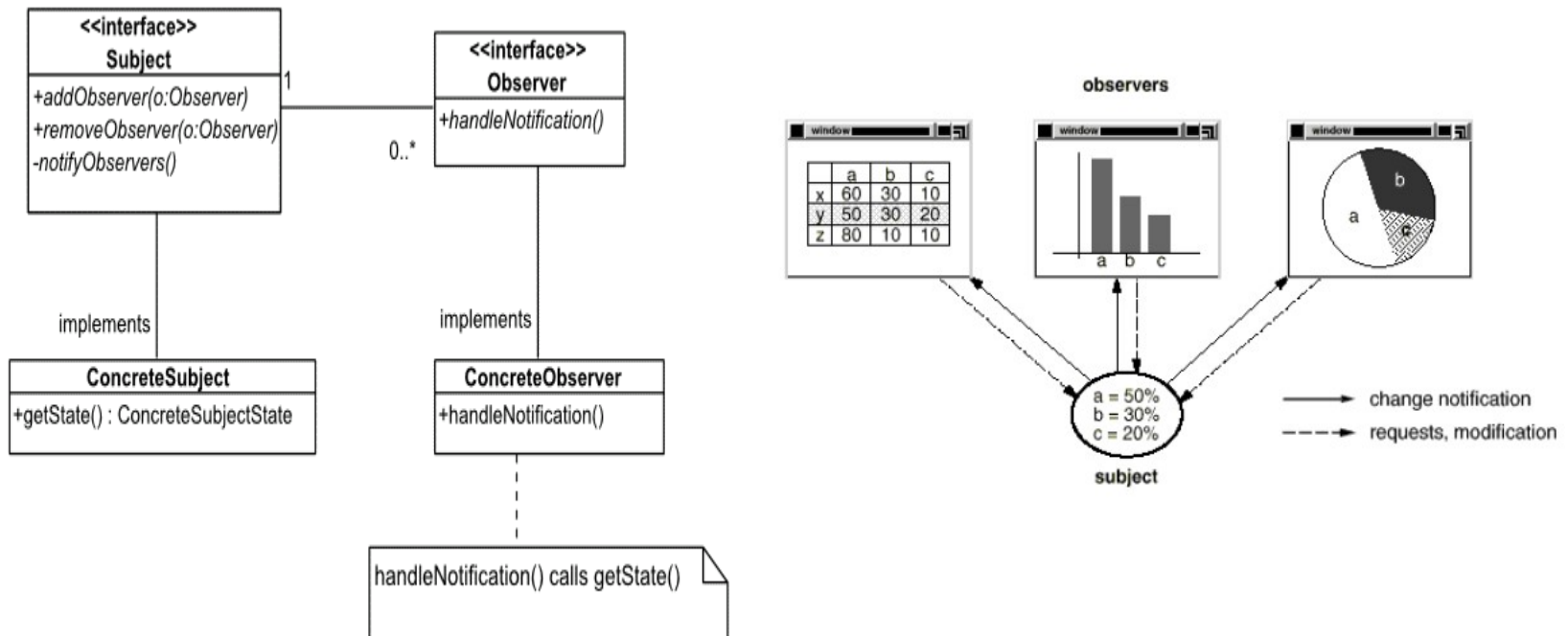
Patrones de Diseño

Iterator



Patrones de Diseño

Observer



Patrones de Análisis

Idiomas: patrones de bajo nivel específico a un lenguaje de programación; tales como Java y C++.

Patrones de Diseño: se enfoca en un pequeño grupo de objetos que colaboran entre sí. Gamma, Helm, Johnson, Vlissides (1995)

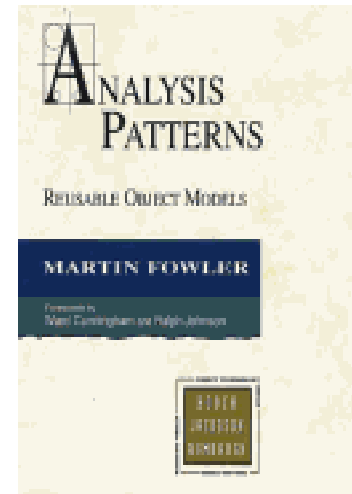
Patrones Arquitectónicos: se enfoca en la estructura de subsistemas de un sistema. Buschmann, Meunier, Rohnert, Sommerlad, Stal (1996)

Patrones de Análisis: se enfoca en patrones encontrados durante el análisis OO. Fowler (2002)

Patrones de Análisis

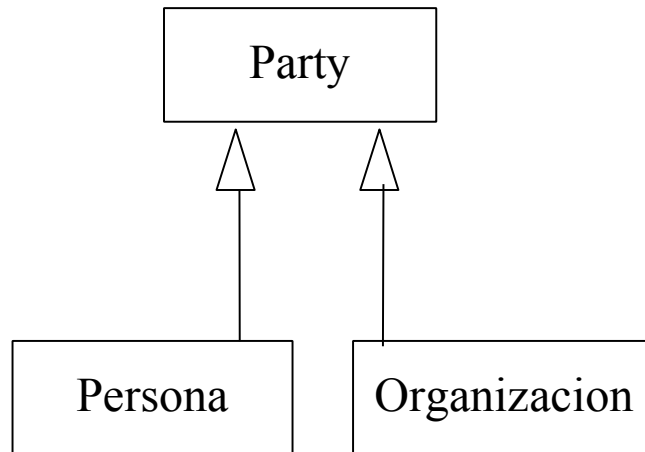
Características

- Durante etapa de análisis surgen abstracciones de análisis particulares.
- Estas abstracciones representan patrones reusables detectados en diferentes dominios
- A diferencia de los patrones de diseño, no profundizan en detalles de implementación



Patrones de Análisis

Ejemplo



Idiomas

Idiomas: patrones de bajo nivel específico a un lenguaje de programación; tales como Java y C++

Patrones de Diseño: se enfoca en un pequeño grupo de de objetos que colaboran entre sí. Gamma, Helm, Johnson, Vlissides (1995)

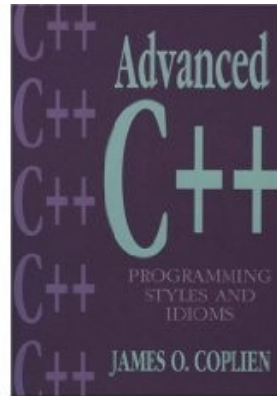
Patrones Arquitectónicos: se enfoca en la estructura de subsistemas de un sistema. Buschmann, Meunier, Rohnert, Sommerlad, Stal (1996)

Patrones de Análisis: se enfoca en patrones encontrados durante el análisis OO. Fowler (2002)

Idiomas

Características

- Bajo nivel de abstracción
- Define una guía implementando los componentes y relaciones del patrón
- Considera al patrón en un lenguaje de programación específico

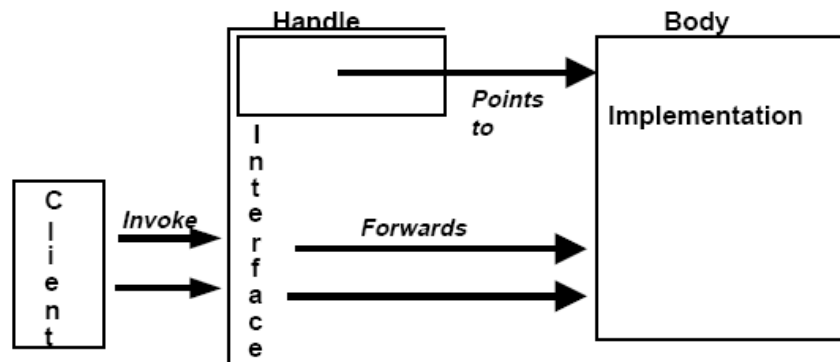


Idiomas

Handle/Body

Problema: Cómo separar interfase de implementación en objetos C++

Solución: Separar una clase diseño en dos clases de implementación. Una toma sobre el rol de un identificador y presenta la interfase de la clase a el usuario. Esta primer clase es el “handle”. La otra clase, llamada “body” encapsula la implementación. El “handle” delega las invocaciones de función miembro a “body”.



Contenido

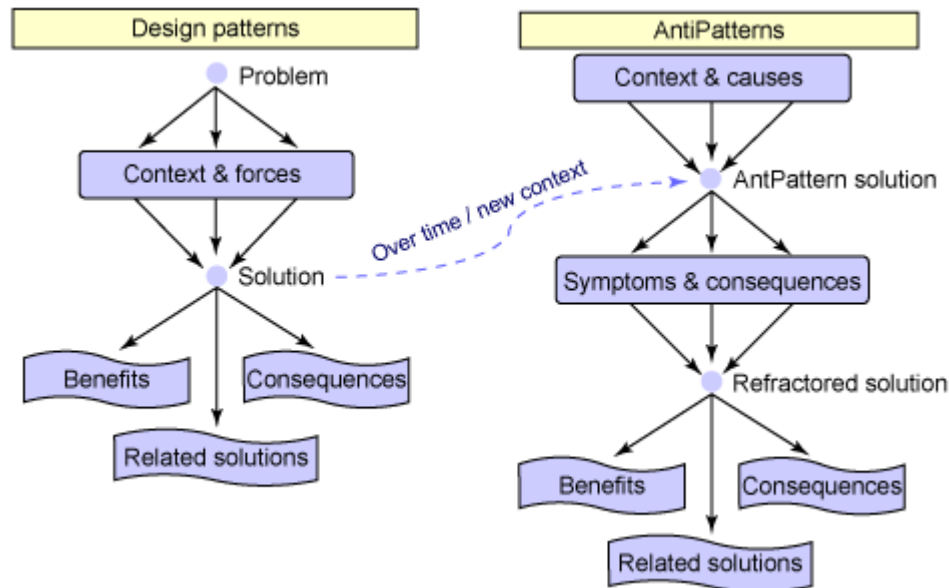
Patrones

- Introducción
- Patrones arquitectónicos
- Patrones de diseño
- Patrones de análisis
- Idiomas
- **Conceptos asociados**
- Patrones en MDA

Conceptos asociados

Antipatrones

Proveen experiencia al reconocer problemas recurrentes y además proveen soluciones para remediarlos.



[Fuente imagen: IBM]

Conceptos asociados

Antipatrones

- Blob (“clases gigantes”)
- Lava Flow (“código muerto”)
- Funcional Decomposition (“Diseño no orientado a objetos”)
- Poltergeists (“No se sabe bien lo que hacen algunas clases”)
- Golden hammer (“Para un martillo todo son clavos”)
- Spaghetti code
- Cut-and-paste programming

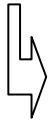
Conceptos asociados

Antipatrones

Blob:

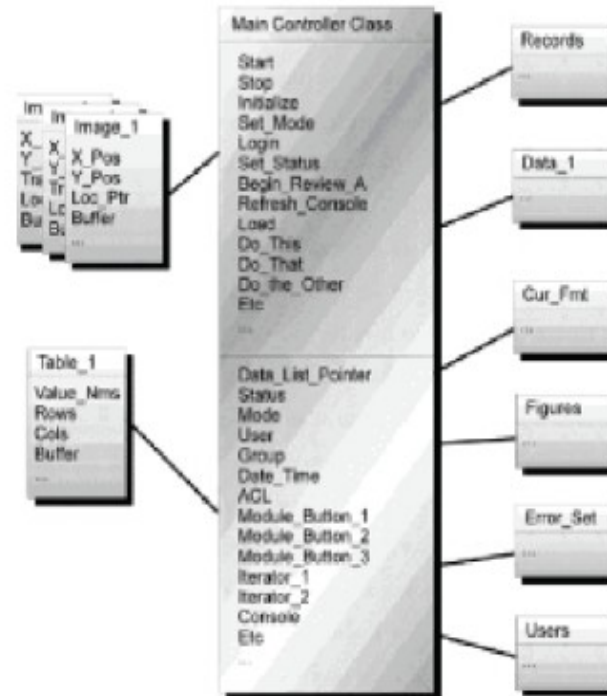
Una clase “Dios” que monopoliza el procesamiento

Muchos métodos sin argumentos



Refactoring

- Separar en clases pequeñas evitando asociaciones transitivas,
- Minimizar la complejidad quitando responsabilidades y repartiéndolas en otras clases



Conceptos asociados

Refactoring

- trata de la estructura interna del software,
- preserva el comportamiento observable,
- mejora una situación dada de acuerdo a un objetivo,
- los pasos de refactoring son pequeños y pueden ser combinados sistemáticamente en secuencias

Aspectos a considerar

- Una forma de analizar la calidad de los modelos es mirar las instancias de patrones predefinidos.
- Es posible detectar patrones deseables o patrones indeseables.
- La reutilización de patrones posibilita obtención de modelos robustos y de alta calidad.
- La detección de patrones indeseables (anti-patrones) en una etapa temprana del diseño previene errores
- El soporte de patrones esta empezando a ser incorporado por herramientas (IBM Rational Software Architect)
- Se plantea la especificación de patrones en numerosos trabajos de investigación

Aspectos a considerar

Curva de aprendizaje

* Un catálogo de patrones no favorece al diseñador experto, sino al diseñador inexperto.

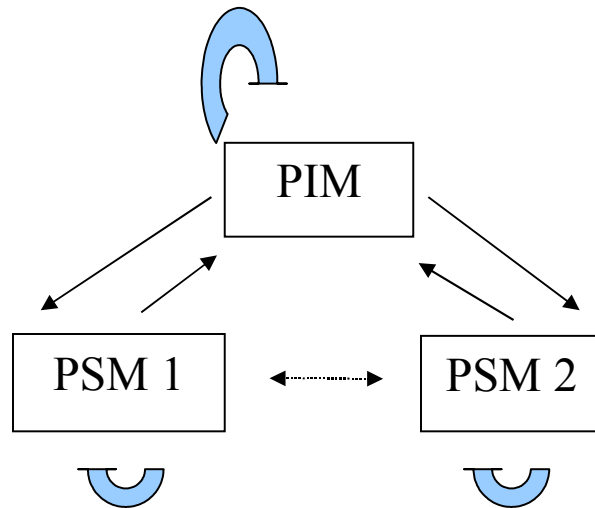
Contenido

Patrones

- Introducción
- Patrones arquitectónicos
- Patrones de diseño
- Patrones de análisis
- Idiomas
- Conceptos asociados
- **Patrones en MDA**

Patrones en MDA

Aplicación de Patrones en Refactoring



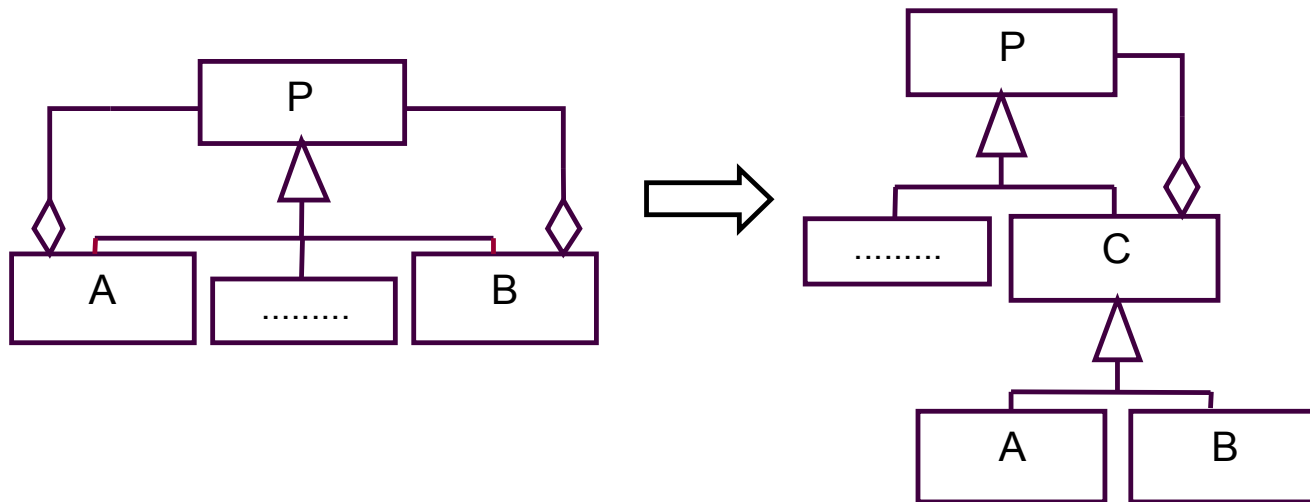
Algunos conceptos

- Refinamiento
- Antirefinamiento
- **Refactoring**
- Puentes

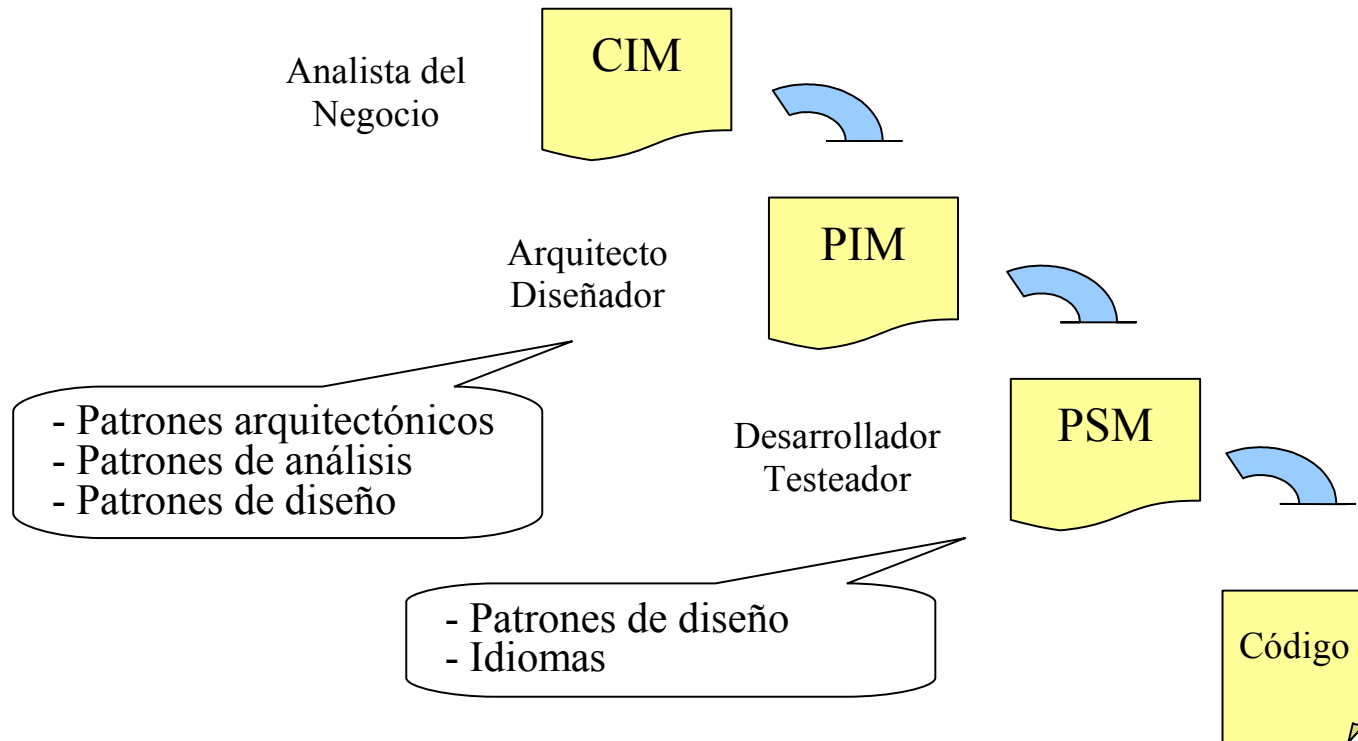
Patrones en MDA

Refactoring en PIM

Extract Composite: extrae una superclase que implementa el *Composite* cuando subclases en una jerarquía implementan el mismo *Composite*.
(Kerievsky)

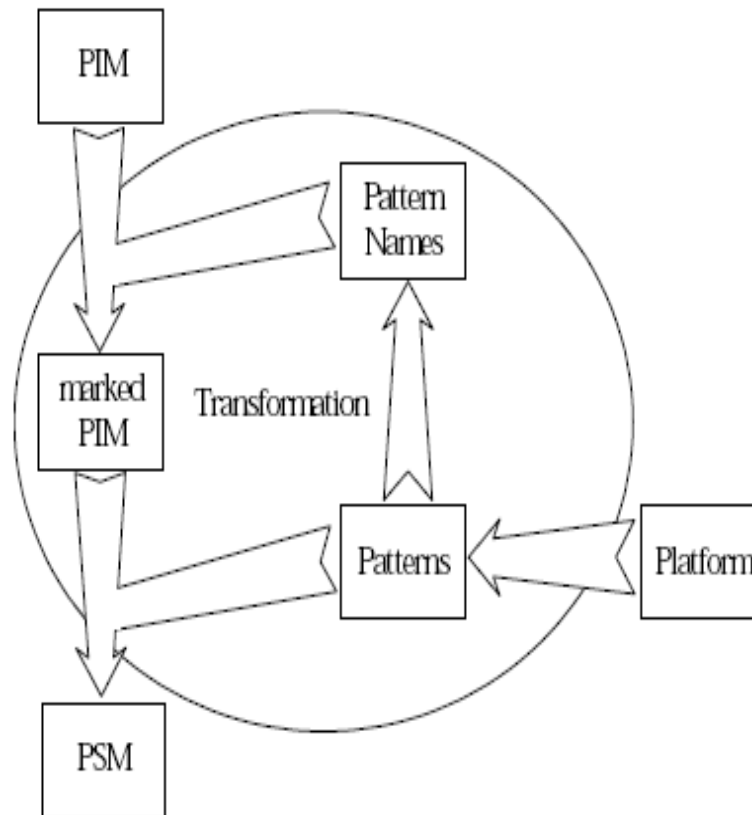


Patrones en MDA



Patrones en MDA

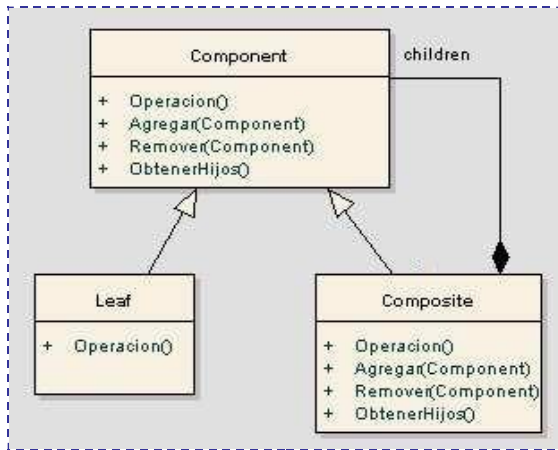
Patrones usados en especificación de transformación



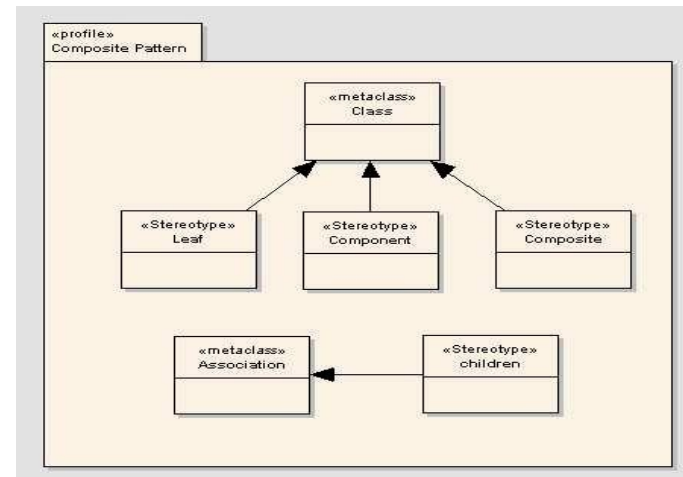
Patrones en MDA

Definición de Patrones como Perfiles UML

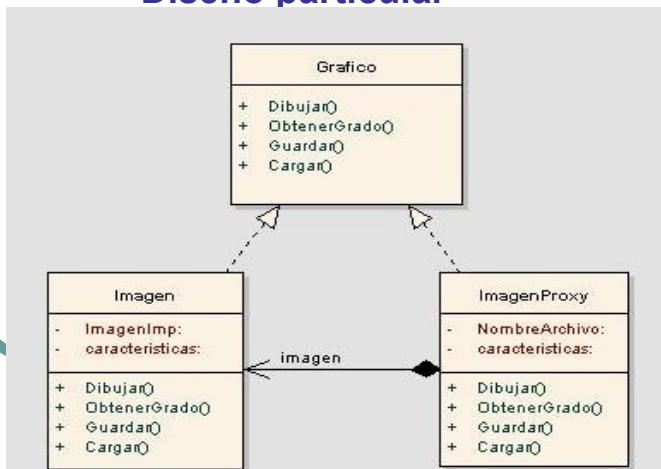
Patrón



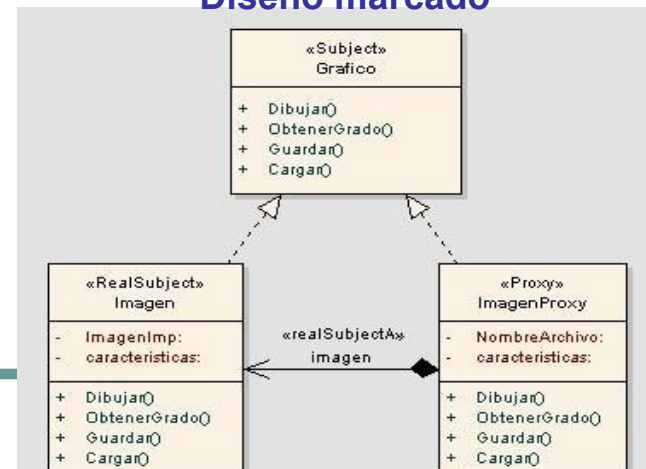
Perfil



Diseño particular



Diseño marcado



Patrones en MDA

Definición de Patrones como metamodelo

