**PATRONES**

Glosario:

* Armazón: framework

Clasificación de patrones:

* **Patrones arquitecturales**: expresan paradigmas fundamentales para estructurar un sistema de software.
* **Patrones de diseño:**
* **Patrones elementales (idioms)**

# Patrones de diseño

Los patrones de diseño están orientados al cambio; esto es algo básico en cualquier diseño de software.

Se clasifican en tres tipos:

* **De creación**
* **Estructurales**
* **De comportamiento**

**De creación:**

*Estos patrones tienen como objetivo el que el comportamiento se defina más por la composición de un conjunto pequeño de comportamientos fundamentales que por la definición mediante herencia de todos los comportamientos posibles.*

*Encapsulan el conocimiento sobre las clases concretas que se van a utilizar.*

*Ocultan la manera de crear objetos de estas clases y cómo se juntan.*

*La visión global de los objetos del sistema son sus interfaces.*

* De clase:

[Factoría abstracta](#_Factoría_abstracta)

Builder

* De objeto

[Método Factoría](#_Método_Factoría)

Prototype

Singleton

Object Pool

**Estructurales:**

* De clase:

Herencia múltiple

Class Adapter

* De objeto:

Object Adapter

Bridge

Composite

Decorator

Facade

Flyweight

Proxy

**Patrones de comportamiento:**

Chain of responsability

Command

Interpreter

Iterator

Mediator

Memento

Observer

State

Strategy

Template method

Visitor

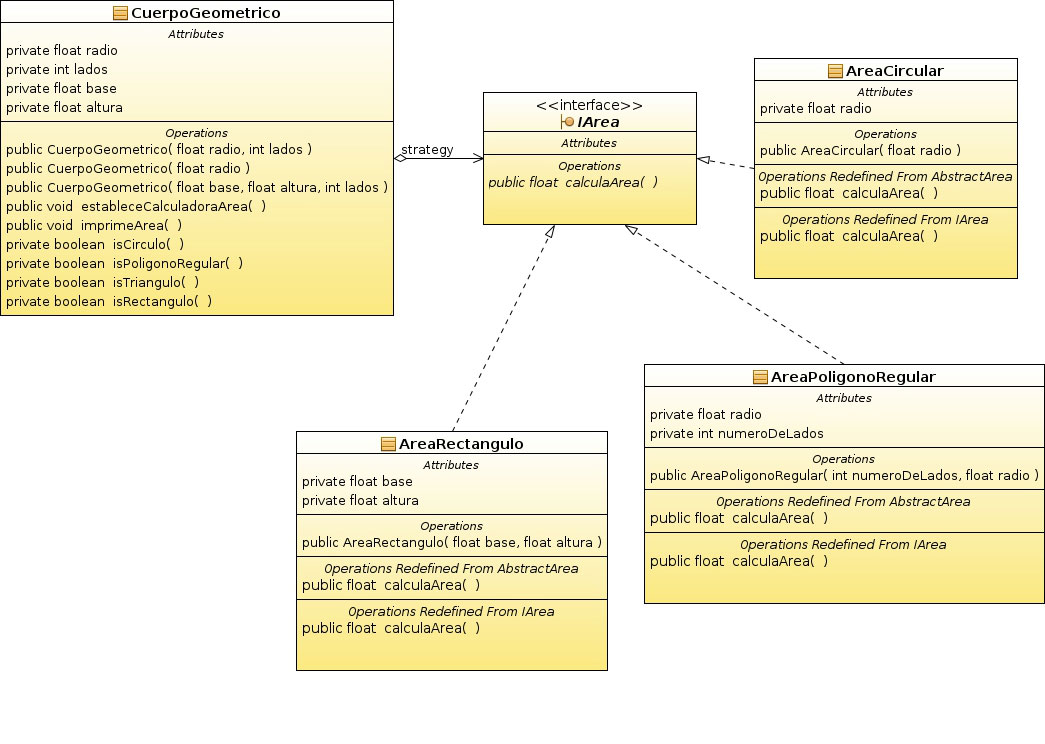
# Factoría abstracta

Factoría para construir familias de objetos.

# Método Factoría

Otras denominaciones: *Factory Method* y *Virtual Constructor*

**Patrón Strategy**



Tenemos la interfaz *IArea*, que será la que defina el método o métodos que conformarán la estrategia, las clases que la implementan (*AreaCircular, AreaRectangulo y AreaPoligonoRegular),* y el contexto (*CuerpoGeometrico)*, que es donde se desarrollará la estrategia.

**Patrón Composite**