

Frameworks, librerías y patrones

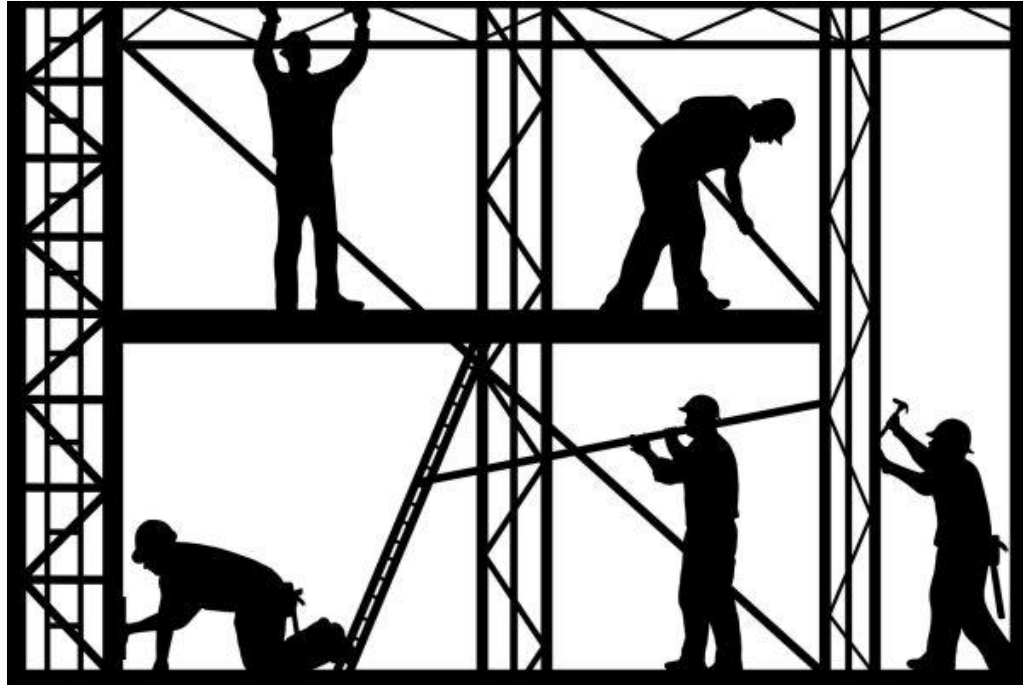
Héctor Quiroga

Vicente Besa

Cristóbal Martínez

Frameworks

- “A *basic structure underlying a system, concept, or text.*”. Oxford University Dictionary
- Es un esquema o esqueleto que facilita el desarrollo e implementación de una aplicación



¿Qué incluye?

- Librerías
- Compilador
- API's
- SDK
- IDE's

¿Para qué?

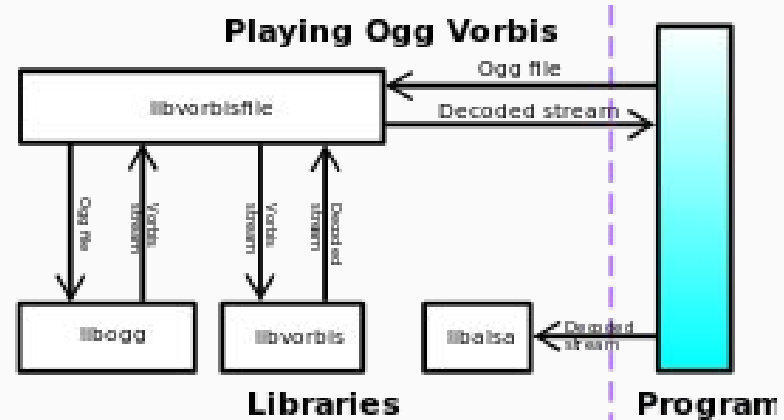
- Simplifica tareas
- Reutilización de Código
- Estandarización del código
- Enfoque en tareas específicas

Ejemplos



Librerías

- Es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.



Propiedades

- *Entrega funcionalidades útiles para el programador*
- *Son propias del lenguaje de programación*
- *Su comportamiento no se espera que se autonomo*
- *Pueden utilizar otras librerías*



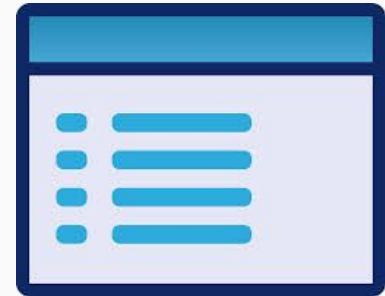
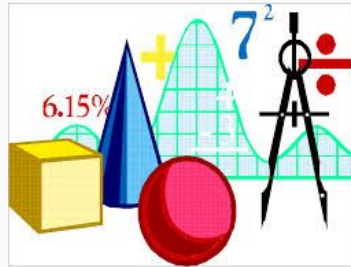
Tipos

Existen 2 tipos de librerías en computación:

- *Librerías estáticas*
- *Librerías dinámicas*

Algunos ejemplos globales

- *Librería de interfaz gráfica*
- *Librerías de manejo de operaciones matemáticas*
- *Librerías para manejar problemas de optimización*
- *Librerías de estructuras de datos especiales*

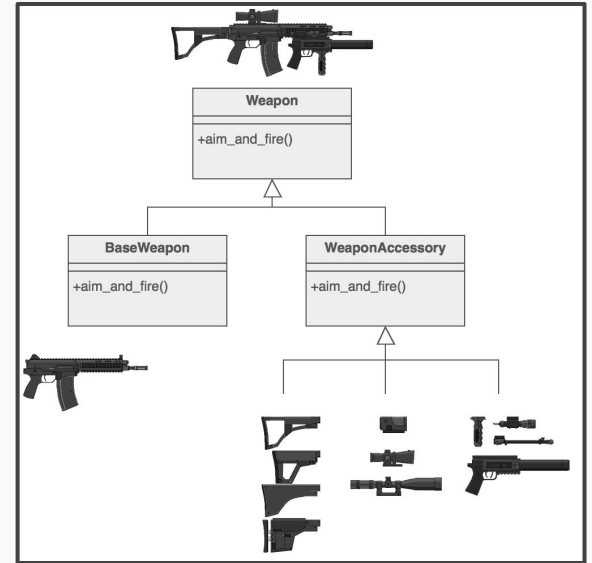


Patrones

- Base de solución a problemas recurrentes.
- Permiten reutilizar diseños y arquitecturas.
- Se caracterizan por tener un:
 - Contexto
 - Problema
 - Solución

Ejemplos de Patrones

- Patrones OOP
 - GOF.
- Patrones no funcionales.



Diferenciación Patrón de Framework/Librería

Un *framework* es un esquema que facilita el desarrollo de una aplicación, y permite desarrollar sobre ésta.

Una librería es un conjunto de implementaciones funcionales que resuelven una funcionalidad definida.

En cambio, un patrón es una **solución abstracta** que resuelve problemas recurrentes, y por lo tanto, un *framework* y una librería puede incluir **varios patrones** dentro de su implementación.