

Clases Frameworks

El uso de librerías / frameworks proporciona una serie de ventajas y desventajas.

Ventajas de usar librerías / frameworks:

- Menor tiempo de desarrollo
- Menor incertidumbre
- Mayor confiabilidad, ya que funciona para muchos casos anteriores.
- Promueve trabajo en equipo.

Desventajas de usar librerías / frameworks:

- Requiere mantener configuraciones de software (versiones de dependencias)
- Riesgo de dependencias no deseadas.
- Código externo
- Desarrollo de código reusable

El reuso es inesperado



Reuso en librería y en framework

Librería:

- Se reusa **código**.
- Librería => **Conjunto de clases**

- Funcionalidad
 - Instancias
 - Clases estaticas/subclases,interfaces
- El sistema/mdulo
 - Crea instancias
 - Invoca funcionalidad

Freamework:

- Se resusa
 - Una manera de ejecutar
 - Una manera de instanciar
 - Una manera de extender
- Es incompleto. No hace nada concreto
- Cookbook: Reglas de uso
 - Instanciacion: Implementar aplicacion
 - Extension: Agregar opciones

Definiciones:

Libreria:

- Conjunto de clases, metodos o funciones ue pueden utilizarse para realizar tareas especificas.
- Nosotros controlamos el flujo del programa y decidimos cuando y como usar la libreria

Por ejemplo podriamos usar una libreria para crear una instancia de alguna clase (En java podemos importar la libreria LocalDate y trabajar con fechas.)

Framework:

Es una aplicacion "semi-completa" , "reusable" que puede ser especializada para producir aplicaciones a medida...

- Estructura predefinida para desarrollar aplicaciones.
- El flujo del programa lo controla el framework y llama a nuestro codigo en algunos momentos.
- Manera de ejecutar las aplicaciones.

Analogia entre Libreria y framework que dijo el profe en clase 👍😊:

Una libreria es como una caja de herramientas: Elegimos que herramienta usar y cuando.
Un framework es como un molde para hornear: Nosotros elegimos el molde, pero el molde define la forma final.

Definicion De framework:

Un framework es una estructura predefinida para desarrollar aplicaciones, ademas proporcionan un loop de control que gestiona el flujo de ejecucion de la aplicacion. Proveen una solucion reusable para una familia de aplicaciones.

Definicion de Hooks, HotSpot, FrozenSpot:

- **Hooks:**

Son puntos definidos en un framework donde se puede inyectar codigo para ser ejecutado en momentos especificos del ciclo de vida del framework.

- No se definen clases ni metodos personalizados, solo engancha puntos definidos por el framework. En java, los Hooks suelen estar relacionados a anotaciones, por ejemplo `@PostConstruct` y `@PreDestroy` son considerados Hooks.
- **HotSpot:** Es una parte del framework abierta a la personalizacion, aca es donde podemos inyectar nuestro propio comportamiento o logica. Permite al programador extender la funcionalidad sin romper la estructura base del framework.
 - Generalmente podemos agregar logica o comportamiento a traves de:
Subclases, implementaciones de interfaces, metodos personalizados, sobreescritura de metodos.
- **FrozenSpot:** Es una parte del framework que no puede modificarse ni personalizarse. Mantiene la estructura y el flujo principal del framework intacto.

Framework de caja blanca vs Framework de caja negra:

- **Framework de caja blanca:**

Es un framework donde el programador puede ver, modificar e interactuar directamente con el codigo fuente del framework.

Una caracteristica importantisima es que los framework de caja blanca, al permitir modificar clases, permite modificar el loop de control, es decir se puede modificar el flujo de ejecucion del framework.

- **Uso:** Se espera que el programador entienda como funciona internamente el framework y pueda extender o personalizar su comportamiento.
- Permite extender clases, sobreescribir metodos.
Ventajas y Desventajas:
- **Ventajas:** Gran flexibilidad y control.

- Desventajas: Requiere un conocimiento profundo del framework y su arquitectura, por lo que aumenta la curva de aprendizaje.
 - **Framework de caja negra:**
Es un framework donde el programador no tiene acceso al código fuente y no se espera que se modifique. Se espera que el programador solo interactúe a través de interfaces o APIs.
 - Uso: Se espera que el programador utilice el framework mediante métodos, funciones o servicios predefinidos, sin preocuparse de cómo están definidos internamente.
- Ventajas y Desventajas:
- Ventajas: Más fácil de usar y mantener, ya que el framework maneja los detalles internos.
 - Desventajas: Menos control y menos capacidad de personalización.

Guía de lectura sobre frameworks

Un framework se puede ver como resultado de refactorizar una aplicación, no solo para mejorar su calidad interna, sino también para abstraer lo que es común entre una familia de aplicaciones similares.

Para separar la implementación de la parte constante (lo que conocemos como **frozenspot**) y de las partes que varían (las que conocemos como **hotspots**), se introducen en el framework **puntos de extensión**.

Estos puntos de extensión tienen forma de plantillas que ofrecen ganchos para que mediante herencia o composición, los programadores puedan definir comportamiento variable. Algunos patrones de diseño (**particularmente el template method y el strategy**) son frecuentemente utilizados para introducir estos puntos de extensión.

La construcción y uso de frameworks es una estrategia para modularizar y reutilizar, que mejora la calidad del software y aumenta la productividad de quienes lo hacen.

En pocas palabras, un framework encapsula el diseño y la implementación de la parte más compleja de dichas aplicaciones y permite su reutilización.

Un framework es en sí mismo, un módulo de software reutilizable. El framework es independiente de las aplicaciones, sin embargo, las aplicaciones dependerán de que las interfaces (conjunto de puntos de extensión) y comportamiento del framework se mantenga estable.



¿Qué es un hook?

Un **hook** es un método:

- Que se **define en la superclase** (con una implementación por defecto)

- Y que **puede ser sobrescrito** en las subclases para **personalizar el comportamiento**

Se suele declarar `protected` para que las subclases lo puedan redefinir, pero no desde fuera del paquete si no es necesario.

NO es lo mismo que un hotspot.

Diferencias:

Término	¿Qué es?
Hook	Método que puede redefinirse en subclases, pero no es obligatorio.
Hotspot	Punto en el código donde el comportamiento puede variar (más general, puede ser un hook, una interfaz, etc.)
Template Method	Método que llama a varios pasos, algunos fijos y otros personalizables (hooks)