

Inversión de control

Inversión de control (*Inversion of Control* en inglés, **IoC**) es un principio de diseño de software en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales. En los métodos de programación tradicionales la interacción se expresa de forma imperativa haciendo llamadas a procedimientos o funciones. Tradicionalmente el programador especifica la secuencia de decisiones y procedimientos que pueden darse durante el ciclo de vida de un programa mediante llamadas a funciones. En su lugar, en la inversión de control se especifican respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que algún tipo de entidad o arquitectura externa lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir.

Esta nueva filosofía es muy útil cuando se usan frameworks de desarrollo. Es el framework el que toma el control, el que define el flujo de actuación o el ciclo de vida de una petición. Es decir, es el framework quien ejecuta el código de usuario.¹

La inversión de control es un término genérico que puede implementarse de diferentes maneras. Por ejemplo se puede implementar mediante eventos o mediante Inyección de Dependencias.²

Índice

- [Idea intuitiva](#)
- [Motivación](#)
- [Definición](#)
- [Ejemplos](#)
- [Véase también](#)
- [Referencias](#)
- [Enlaces externos](#)

Idea intuitiva

El principio sobre el que se basa la Inversión de control es el llamado principio de Hollywood llamado así porque es la habitual frase empleada por los directores de casting en Hollywood para evitar estar recibiendo llamadas de aspirantes preguntando si han sido aceptados o no:¹

No nos llames; nosotros te llamaremos

Veamos un ejemplo típico para explicar el concepto.³ Imaginemos que estamos escribiendo un programa que tenemos el siguiente programa en pseudocódigo:

```
escribir "¿Cuál es tu nombre?"
nombre <- Introducir nombre por teclado
procesar_nombre(nombre)
escribir "¿Cuál es tu pregunta?"
pregunta <- Introducir pregunta por teclado
procesa_pregunta(pregunta)
```

En la interacción existente, el código es el que decide el flujo: cuándo preguntar, y cuándo leer y procesar las respuestas.

Supongamos que queremos adaptar nuestro programa a un sistema gráfico de ventanas. Tendríamos el pseudocódigo:

```
importar ventanas, botones, campos_texto
ventana_principal <- crear nueva Ventana.
etiqueta_nombre <- crear nueva etiqueta "¿Cuál es tu nombre?"
añadir a ventana_principal << etiqueta_nombre
nombre <- crear nuevo campo_texto
añadir a ventana_principal << nombre
en_caso_de_evento('dejar foco') sobre nombre, procesar_nombre(nombre)
etiqueta_pregunta <- crear nueva etiqueta "¿Cuál es tu pregunta?"
añadir a ventana_principal << etiqueta_pregunta
pregunta <- crear nuevo campo_texto
añadir a ventana_principal << pregunta
en_caso_de_evento('dejar foco') sobre pregunta, procesar_pregunta(pregunta)
Ventanas.buclePrincipal
```

En este segundo pseudocódigo no se tiene control sobre cuándo se llamarán a las funciones para procesar el nombre y la pregunta, sino que se ha entregado al control al sistema de ventanas. Él decidirá cuándo llamar a estas funciones, basándose en la vinculación que hemos hecho (en_caso_de_evento(...)). El control se ha invertido ya que, en lugar de invocar a una clase, es el *framework* el que invoca lo que le indicamos.

Motivación

En los comienzos de la programación, los programas eran lineales y monolíticos. El flujo de ejecución era simple y predecible, ejecutándose línea tras línea.

Aparecieron dos conceptos que revolucionaron la programación: la **modularidad** y la **reutilización** de los componentes: Se crean bibliotecas de componentes reutilizables. El flujo se complica, saltando de componente a componente, y aparece un nuevo problema: la **dependencia** (acoplamiento) entre nuestros componentes.

El problema se empieza a considerar lo suficientemente importante como para definir nuevos conceptos en el diseño como la Inversión de Control (IoC) implementada a través de Inyección de Dependencias o de eventos

Definición

La inversión de control ocurre cuando trabajamos con una biblioteca y la propia biblioteca es la que invoca el código del usuario. Es típico que la biblioteca implemente las estructuras de alto nivel y es el código del usuario el que implementa las tareas de bajo nivel.

Ejemplos

La Inversión de control es el concepto central del Framework de Spring y usa la implementación por Inyección de dependencias, ya que implementa un "Contenedor" que se encarga de gestionar las instancias (así como sus creaciones y destrucciones) de los objetos del usuario.

La mayoría de las bibliotecas para el desarrollo de aplicaciones con GUI, como GTK o QT usan inversión de control y están dirigidas por eventos.⁴

Véase también

- [Inyección de Dependencias](#)
- [Spring Framework](#)

Referencias

1. Inversión de Control - Principio de Hollywood: “Don’t call us, we’ll call you” (<https://medium.com/all-you-need-is-clean-code/inversi%C3%B3n-de-control-principio-de-hollywood-dont-call-us-we-ll-call-you-179e9c70e3d0>). Miguel Ángel Sánchez. Sep 26, 2017
2. Dependency Injection (DI) vs. Inversion of Control (IOC) (<https://www.codeproject.com/Articles/592372/Dependency-Injection-DI-vs-Inversion-of-Control-IO>). Shivprasad koirala. 31 de diciembre de 2013
3. Inversión de control (<http://www.davidvalverde.com/blog/inversion-de-control/>). David María Valverde Martínez
4. Introducción a la programación dirigida por eventos (<https://www.genbetadev.com/paradigmas-de-programacion/introduccion-a-la-programacion-dirigida-por-eventos>).Oscar Campos. 21 julio 2011

Enlaces externos

- [Inversion Of Control \(http://martinfowler.com/bliki/InversionOfControl.html\)](http://martinfowler.com/bliki/InversionOfControl.html) - Martin Fowler
- [Inversion of Control Containers and the Dependency Injection pattern \(http://martinfowler.com/articles/injection.html\)](http://martinfowler.com/articles/injection.html) - Martin Fowler

Obtenido de «https://es.wikipedia.org/w/index.php?title=Inversión_de_control&oldid=125248615»

Esta página se editó por última vez el 16 abr 2020 a las 23:14.

El texto está disponible bajo la Licencia Creative Commons Atribución Compartir Igual 3.0; pueden aplicarse cláusulas adicionales. Al usar este sitio, usted acepta nuestros términos de uso y nuestra política de privacidad. Wikipedia® es una marca registrada de la Fundación Wikimedia, Inc., una organización sin ánimo de lucro.