

# Patrones de diseño

*18 de Enero de 2018 a las 13:40*

En este artículo trataré un tema que todo desarrollador debe conocer porque le puede resultar muy útil en cualquier proyecto que se pueda encontrar que los **patrones de diseño**. Mediante una sencilla y pequeña explicación sobre este concepto de programación. La forma de como explicar los patrones, será mediante una serie de preguntas y respuestas.

## ¿Qué son los patrones de diseño?

Un patrón es una solución de diseño para una serie de problemas, cada patrón está enfocado a resolver un determinado problema, pretende facilitar el trabajo de un desarrollador proporcionando una solución de un problema que se encuentra de forma habitual en el desarrollo de aplicaciones. No tiene mucho sentido crear un patrón para resolver un problema particular, porque solo sería efectivo para ese problema.

Los patrones deben tener un serie de características:

- **Funcionales:** Son soluciones probadas y documentadas, esto es una garantía para demostrar su efectividad para resolver problemas.
- **Fáciles de implementar:** Para que sea una buena solución no deben tener mucha dificultad en su implementación. Una dificultad excesiva puede significar que la solución no esta bien planteada
- **Flexibles:** Un patron de diseño puede sufrir variaciones y seguir siendo solución al mismo problema. Por ejemplo, tenemos el patrón de diseño MVC(Modelo-Vista-Controlador) y una variación es el patrón MVT(Modelo- Vista-Plantilla) y los dos sirven para desarrollo web.
- **Reutilizable:** La solución aportada en el patrón puede utilizarse en problema similares y en diferentes escenarios.

Un patrón de diseño solo proporciona una solución de diseño pero no su implementación en el lenguaje de programación utilizado por el desarrollado, solo describe como solucionar un determinado problema. La descripción de esta solución se detalla en un documento con una estructura estandarizada, se utiliza una plantillas para este cometido. Existe diversas propuesta de plantillas con ligeras variaciones, donde muestran la información del patron de una forma organizada.

La plantilla mas utilizada contiene los siguientes apartados.

- **Nombre del patrón.**
- **Clasificación del patrón.**

Blog de J.A. Jimenez Toro, [rooteando.com](http://rooteando.com)

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

- **Intención.**
- **También conocido como.**
- **Motivación.**
- **Aplicabilidad.**
- **Estructura.**
- **Participantes.**
- **Colaboraciones.**
- **Consecuencias.**
- **Implementación.**
- **Código de ejemplo.**
- **Usos conocidos.**
- **Patrones relacionados.**

## ¿Para que sirven?

Sirve para facilitar el desarrollo de aplicaciones, proporcionando al desarrollador una solución de diseño que puede implementar, pudiendo reutilizar la misma solución en muchos problemas y no tener que crear una solución para cada problema, ahorrando tiempo en la búsqueda de una solución y en el desarrollo.

Su uso te permite asegurar que tu código es óptimo, porque los patrones de diseño proporciona una solución probada en múltiples desarrollo y obtienes una garantía sobre el código.

Existen muchos patrones de diseño, aunque es un grupo reducido los más utilizados y su conocimiento, en concreto de los más utilizados, aporta al desarrollador una solución probada y efectiva a un determinado problema. Esto no significa que si tenemos un problema que puede aplicarse un patrón de diseño, debemos implementarlo de forma obligatoria, porque puede que no sea la solución más óptima y existe una solución alternativa más óptima. Eso sí, debes tener en cuenta que cuando no aplicas un patrón para solucionar un problema, es por razones muy poderosas, recuerda que los patrones son soluciones ampliamente probadas en su efectividad.

Un patrón de diseño es recomendable conocerlo y utilizarlo, pero no es obligatorio su uso.

El uso de un patrón aporta una serie de ventajas en el desarrollo de aplicaciones, como pueden ser:

Blog de J.A. Jimenez Toro, [rooteando.com](http://rooteando.com)

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

- Todos los patrones son soluciones de diseño que integran una serie de principios de diseño que son recomendables que tu desarrollo los incluya .
- Mucha información disponible, al ser soluciones ampliamente usadas en múltiples desarrollos, cualquier problema que tengas en su uso o implementación será fácil encontrar información al respecto.
- Facilitan su entendimiento, el código que genere será fácilmente entendible por otro desarrollador que conozca el patrón aplicado.
- No requiere probar la solución, son soluciones ampliamente probadas, otros desarrolladores ya han realizado las pruebas pertinentes para demostrar su eficacia.

Aunque los patrones son útiles no conviene abusar de ellos, porque en ocasiones el uso de patrones genera malos hábitos, como modificar los requisitos de un problema para poder aplicar un patrón. También su aplicación , y abuso, puede perjudicar el desarrollo , porque los patrones son soluciones estandarizadas que añaden una capa de abstracción, estamos añadiendo la solución de otros, y mas complejidad al desarrollo.

## ¿Cómo se implementan?

Anteriormente se ha comentado que los patrones solo proporcionan una solución de diseño y no la implementación para un lenguaje de programación, el desarrollador deberá implementarla en su desarrollo. Puede realizar este proceso de dos formas:

- Desarrollo propio, no es lo mas habitual, porque hay disponibles implementaciones en múltiples lenguaje de programación practicamente para todos los patrones. No es recomendable salvo que estes muy seguro de lo que estas haciendo.
- Uso de librerías o framework, es la opción mas recomendable, son herramientas que ha sido probadas, asegurandose que su implementación es correcta.

El desarrollador debe identificar correctamente el problema y patrón que puede aplicar, este proceso se debe realizar correctamente, porque aplicar un patrón a un problema que no le corresponde será una pesadilla para nuestro desarrollo. Aquí el sentido común y la experiencia adquirida sera nuestro gran aliado.

## ¿Qué tipos hay?

Los patrones de diseño pueden ser organizados en diversas categorías, por múltiples factores, la mas habitual es en función del proposito de la solución que aportan. Son divididos en tres categorías:

- Creacionales: Solución al problema mediante la creación objetos.
- Estructurales: La solución que aporta es mediante la composición(relación) de

Blog de J.A. Jimenez Toro, [rooteando.com](http://rooteando.com)

Contenido esta bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

- Comportamiento: Especifican soluciones que definen la interacción (comportamiento) entre los objetos.

## Antipatrones

Los patrones de diseño son soluciones probadas y documentadas que podemos implementar para un determinado problema, es recomendable usarlas porque casi siempre serán una buena solución.

Pero existe el concepto contrario, los **antipatrones**, que son el concepto contrario. Son patrones que proporcionan malas soluciones para determinados problemas, todo desarrollador debe evitar implementar estas soluciones.

Igual que los patrones de diseño, los antipatrones están documentados y probados que son malas soluciones. De esta forma si un desarrollador conoce los antipatrones, puede evitar implementar soluciones que son malas para sus desarrollos.

## Conclusión

El conocimiento tanto de los patrones como de los antipatrones proporciona una buena práctica de programación, permitiendo desarrollar mejores aplicaciones.

El desarrollador debe identificar correctamente el problema para saber si puede aplicar y no abusar de los patrones en el desarrollo de la aplicación. También debe tener en cuenta que puede existir soluciones alternativas a un problema.

Respecto a la implementación es muy aconsejable que utilice herramientas para esa finalidad, el desarrollo será más fácil y seguro.

## Enlaces

Mi primera recomendación para conocer los patrones de diseño, es la *Wikipedia*, donde explican de una manera correcta y resumida, una buena primera toma de contacto.

[https://es.wikipedia.org/wiki/Patr%C3%B3n\\_de\\_dise%C3%B1o](https://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o)

El libro donde empecé los patrones de diseño en programación *Design Pattern* escrito por el grupo de autores conocidos como *Gang of Four*.

<http://wiki.c2.com/?DesignPatternsBook>

Blog de J.A. Jimenez Toro, [rooteando.com](http://rooteando.com)

Contenido está bajo licencia [Creative Commons Reconocimiento No Comercial Sin Obra Derivada 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

---

Otro libro interesante sobre patrones de diseño es *Head First Design Patterns*.

<http://shop.oreilly.com/product/9780596007126.do>