
MICROKERNEL

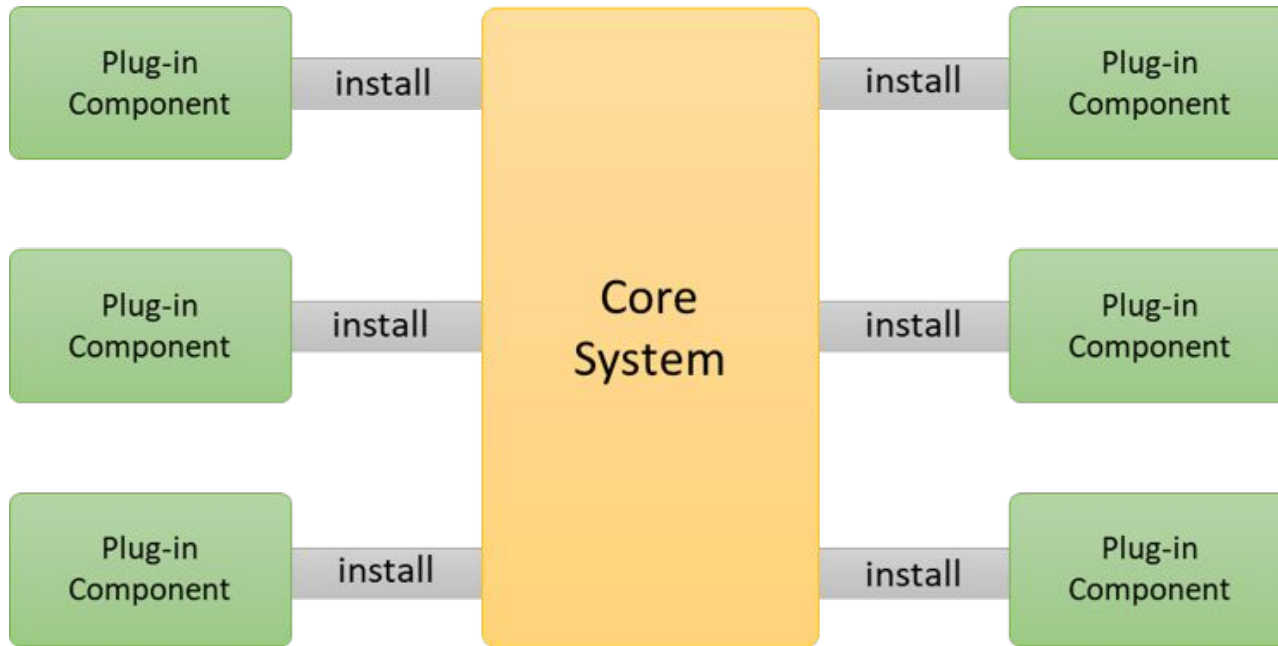
Karina Reyna Anchondo
David Pérez García

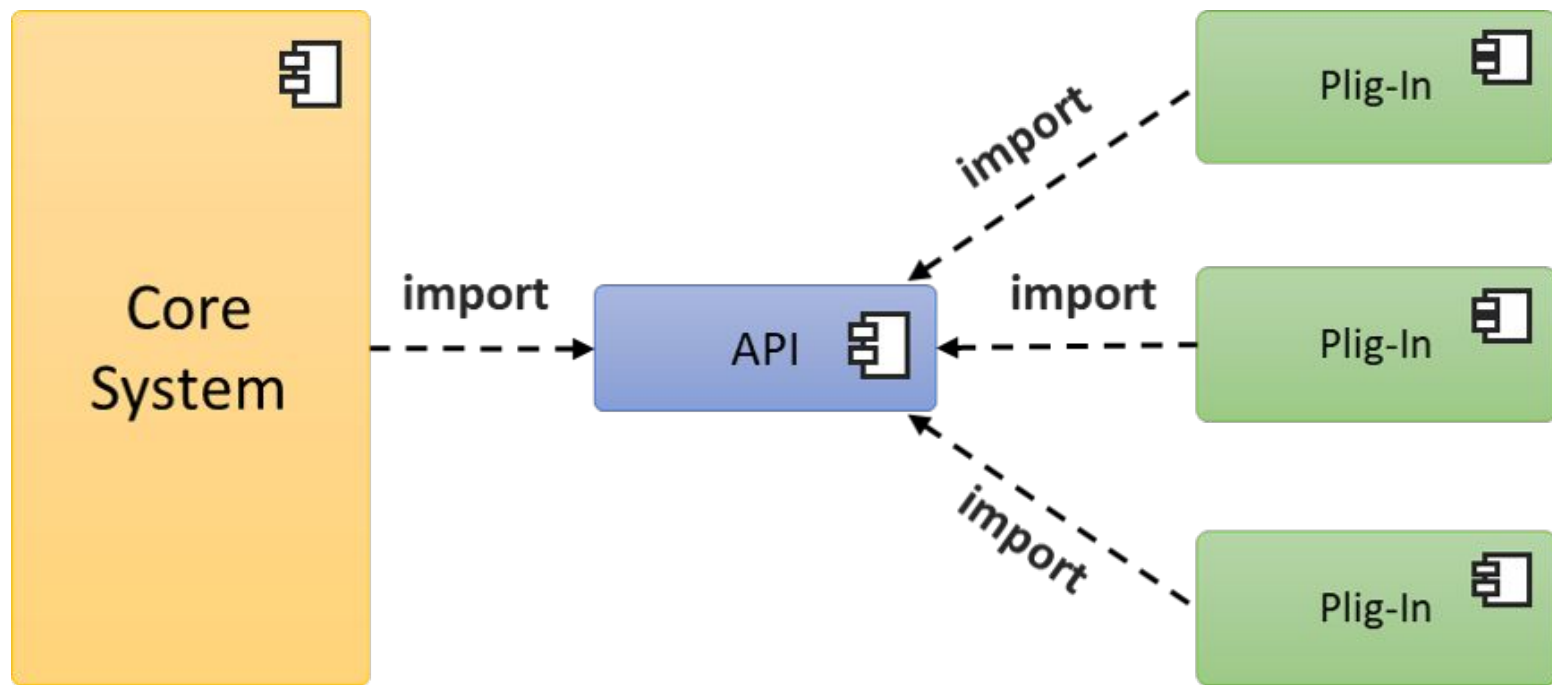
Que es?

conocido como arquitectura de Plug-in, permite crear aplicaciones extensibles, mediante la cual es posible agregar nuevas funcionalidades mediante la adición de pequeños plugins que extienden la funcionalidad inicial del sistema.



1 Componente Core Muchos Plugins o módulos





Por qué es importante?

El patrón de arquitectura Microkernel se aplica a sistemas de software que deben estar habilitados para adaptarse a requerimientos cambiantes del sistema



Ventajas

- Testabilidad: Debido a que los Plugins y el sistema Core son desarrollados de forma por separado, es posible probarlos de forma aislada.
 - Performance: En cierta forma, muchas de las aplicaciones basadas en Microkernel trabajan de forma Monolítica una vez que el Plug-in es instalado, lo que hace que todo el procesamiento se haga en una sola unidad de software.
 - Despliegue: Debido a la naturaleza de Plugins es posible instalar fácilmente todas las características adicionales que sea necesarias, incluso, pueden ser agregar en tiempo de ejecución, lo que en muchos casos ni siquiera requiere de un reinicio del sistema.
 - Reutilización: Debido a que los Plugins puede ser instalados en cualquier instancia del sistema Core, es posible reutilizar los módulos en varias instancias, incluso, es posible comercializarlas de forma independiente. Solo como ejemplo, existen empresas que se dedican exclusivamente a desarrollar Plugins para venderlos, como es el caso de los Plugins de Wordpress.
-



Desventajas

Los sistemas que utilizan una arquitectura de Microkernel no son fáciles de desarrollar, pues necesitamos crear aplicaciones que son capaces de agrandar dinámicamente su funcionalidad a medida que nuevos plugins son instalados, al mismo tiempo que debemos de tener mucho cuidado de que los plugins no modifiquen o alteren la esencia de la aplicación.

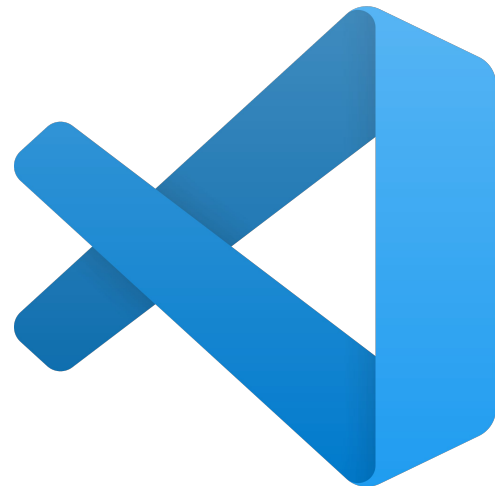
el sistema Core y los plugins son desarrollados por lo general por equipos separados, por lo que el sistema Core debe de dejar muy en claro como los plugins deben se desarrollados y deben de tener un archivo descriptor que le diga al sistema Core como debe de instalarse o debería de mostrar el plugin ante el usuario.



Ejemplo utilizado en

- Eclipse, Netbeans
- Visual Studio Code
- sistemas de ofimática como:

Word, Powerpoint, Excel



GRACIAS POR SU ATENCION
