# ¿Qué es un marco de software?

Llego muy tarde para responderlo. Pero me gustaría compartir un ejemplo, en el que solo pensaba hoy. Si te dijera que cortaras un pedazo de papel con dimensiones de 5m por 5m, entonces seguramente lo harías. Pero supongamos que le pido que corte 1000 trozos de papel de las mismas dimensiones. En este caso, no hará la medición 1000 veces; obviamente, haría un marco de 5m por 5m, y luego, con la ayuda de este, podría cortar 1000 trozos de papel en menos tiempo. Entonces, lo que hizo fue crear un marco que haría un tipo específico de tarea. En lugar de realizar el mismo tipo de tarea una y otra vez para el mismo tipo de aplicaciones, crea un marco que tiene todas esas instalaciones juntas en un paquete agradable, por lo tanto, proporciona la abstracción para su aplicación y, lo que es más importante, para muchas aplicaciones.

# Respuesta con 421 votos.

Técnicamente, no necesitas un marco. Si está creando un sitio realmente simple (piense en la web en 1992), puede hacerlo todo con HTML codificado y algo de CSS.

Y si desea hacer una aplicación web moderna, tampoco necesita usar un marco para eso.

En su lugar, puede elegir escribir toda la lógica que necesita, siempre. Puede escribir su propia capa de almacenamiento / persistencia de datos o, si está demasiado ocupado, simplemente escriba SQL personalizado para cada acceso a la base de datos. Puede escribir sus propias capas de autenticación y manejo de sesión. Y su propia lógica de rending de plantillas. Y su propia lógica de manejo de excepciones. Y sus propias funciones de seguridad. Y su propio marco de prueba de unidad para asegurarse de que todo funcione bien. Y el tuyo ... [continúa durante bastante tiempo]

**Por otra parte, si se hace uso de un marco**, podrás beneficiarse de la buena, por lo general revisada por expertos y el trabajo de decenas si no cientos de otros desarrolladores, que bien puede ser mejor de lo que muy bien probado. Podrá construir lo que desea rápidamente, sin tener que perder tiempo construyendo o sin preocuparse demasiado por los elementos de infraestructura enumerados anteriormente.

Puede hacer más en menos tiempo y saber que es muy probable que el código marco que está utilizando o extendiendo se haga mejor que usted mismo.

¿Y el costo de esto? Invertir algo de tiempo aprendiendo el marco. Pero, como lo atestiguarán prácticamente todos los desarrolladores web, definitivamente vale la pena el tiempo dedicado a aprender a obtener beneficios masivos (realmente, masivos) al usar el marco que elija.

**Respuesta con 91 votos**

Un marco de software, en programación de computadoras, es una abstracción en la que el código de usuario que proporciona funcionalidad genérica puede ser anulado o especializado de forma selectiva mediante código de usuario que proporciona funcionalidad específica. Los marcos son un caso especial de las bibliotecas de software, ya que son abstracciones reutilizables de código envuelto en una interfaz de programación de aplicaciones (API) bien definida, pero contienen algunas características distintivas clave que las separan de las bibliotecas normales.

Los marcos de software tienen estas características distintivas que los separan de las bibliotecas o aplicaciones de usuario normales:

1. **inversión de control** : en un marco, a diferencia de las bibliotecas o las aplicaciones de usuario normales, el flujo general de control del programa no lo dicta la persona que llama, sino el marco. [1]
2. **comportamiento predeterminado** : un marco tiene un comportamiento predeterminado. Este comportamiento predeterminado debe ser un comportamiento útil y no una serie de no-ops.
3. **Extensibilidad** : el usuario puede ampliar un marco de trabajo generalmente mediante una anulación selectiva o especializado por código de usuario que proporciona una funcionalidad específica.
4. **código marco no modificable: el código** marco, en general, no se permite modificar. Los usuarios pueden extender el marco, pero no modificar su código.

Es posible que lo "necesite" porque puede proporcionarle un excelente acceso directo al desarrollar aplicaciones, ya que contiene muchas funciones ya escritas y probadas. La razón es bastante similar a la razón por la que usamos bibliotecas de software.

**Respuesta con 53 votos.**