

MVC vs SPA

2025

Tomás Cabello Fernández

Desarrollo Web en Entorno Servidor

Desarrollo de Aplicaciones Web

MVC vs SPA

MVC (Modelo-Vista-Controlador)

El patrón **MVC** es una forma de organizar el código de una aplicación para separar las responsabilidades. Su principal objetivo es dividir la aplicación en tres componentes interconectados, haciendo el código más manejable y fácil de mantener.

- **Modelo (Model):** Se encarga de la lógica de negocio y de los datos. Es la "memoria" de la aplicación. Interactúa con la base de datos para obtener, guardar o modificar información.
- **Vista (View):** Es la interfaz de usuario. Se encarga de mostrar los datos del Modelo. En una aplicación web, la Vista es el código HTML y CSS que el usuario ve en su navegador.
- **Controlador (Controller):** Actúa como intermediario. Recibe las peticiones del usuario (por ejemplo, un clic en un botón), las procesa, interactúa con el Modelo para obtener o actualizar datos, y luego le indica a la Vista qué información debe mostrar.

EJEMPLO DE PETICIONES Y RESPUESTAS EN MVC

En el modelo MVC tradicional, cada acción del usuario generalmente provoca una nueva carga de página.

Petición (Cliente -> Servidor):

1. El usuario hace clic en un enlace como <ejemplo.com/productos/123>.
2. El navegador envía una petición GET al servidor.

MVC vs SPA

Respuesta (Servidor -> Cliente):

1. El **Controlador** de productos del servidor recibe la petición.
2. El **Controlador** solicita al **Modelo** que busque el producto con el ID 123 en la base de datos.
3. El **Modelo** devuelve los datos del producto al **Controlador**.
4. El **Controlador** usa esos datos para generar una **Vista** completa (un archivo HTML con el título, la descripción, etc. del producto).
5. El servidor envía el archivo HTML completo de la nueva página al navegador. El navegador lo renderiza, reemplazando el contenido de la página actual.

SPA (Single-Page Application)

Una **SPA** es un modelo de aplicación web que carga una única página HTML y luego actualiza dinámicamente el contenido a medida que el usuario interactúa, sin necesidad de recargar la página completa. La lógica de la aplicación se ejecuta en el navegador (en el lado del cliente).

- **HTML Inicial:** El navegador solo carga una página HTML básica una vez.
- **Gestión del Contenido:** A partir de ese momento, JavaScript es el motor que maneja la navegación y la interacción. Cuando el usuario hace clic en un enlace, JavaScript intercepta la acción, hace una petición al servidor para obtener solo los datos necesarios (por ejemplo, en formato JSON) y luego actualiza el HTML de la página sin recargarla por completo.

MVC vs SPA

EJEMPLO DE PETICIONES Y RESPUESTAS EN SPA

En una SPA, las peticiones y respuestas se centran en el intercambio de datos, no en páginas completas.

Petición (Cliente -> Servidor):

1. El usuario hace clic en un enlace de producto en la aplicación.
2. El código JavaScript en el navegador intercepta el evento y utiliza una llamada **asíncrona** (como una petición fetch o XMLHttpRequest) para solicitar los datos del producto a la API del servidor, por ejemplo a `ejemplo.com/api/productos/123`.

Respuesta (Servidor -> Cliente):

1. El servidor (que puede seguir una estructura MVC o similar en el backend) recibe la petición.
2. El servidor procesa la solicitud, busca el producto en la base de datos y, en lugar de generar una página HTML completa, solo devuelve los datos del producto en formato **JSON**.
3. El navegador recibe la respuesta JSON.
4. El código JavaScript toma los datos JSON y los inserta en la plantilla HTML existente, actualizando solo la parte de la página que necesita cambiar. La URL del navegador puede cambiar para reflejar el estado actual, pero la página no se recarga.

TOMÁS CABELLO FERNÁNDEZ