

Práctica 01

Frameworks



1- Busca información de diferentes frameworks y expón sus principal características. (similitudes y diferencias)

Voy a realizar una comparativa entre tres de los principales frameworks más utilizados. Angular, ReactJS y Vue.js.

- **Angular:** Framework de JavaScript desarrollado en TypeScript (JavaScript tipado), de código abierto y mantenido por Google, se utiliza para crear aplicaciones de una sola página (spa), progressive web apps (PWA) y además se puede utilizar junto con Ionic para crear aplicaciones “nativas” híbridas para dispositivos móviles.
 - **Ventajas:**
 - Uso del patron MVVM (Model View Model) que permite a varios desarrolladores trabajar en la misma sección de forma simultánea.
 - Compilación rápida.
 - Amplia comunidad.
 - Buena documentación.
 - **Inconvenientes:**
 - Curva de aprendizaje media-alta.
 - Problemas de migración al cambiar de versión
- **ReactJS:** Es una biblioteca informática de JavaScript de código abierto y desarrollado por Facebook, Instagram y un conjunto de desarrolladores individuales y empresas. Se utiliza normalmente para desarrollar aplicaciones web de grandes dimensiones y en desarrollo híbrido en dispositivos móviles, su principal característica es su rapidez, simplicidad y escalabilidad.
 - **Ventajas:**
 - Curva de aprendizaje suave.
 - Es flexible, se puede adaptar a todo tipo de proyectos.
 - Amplia comunidad.
 - Migración de versiones sencillas.
 - **Inconvenientes:**
 - Demasiada documentación diseminada sin un discurso unificado.
- **Vue.js:** Framework progresivo open source de JavaScript creado por Evan You ex-trabajador de Google, dividido en diferentes librerías bien acotadas que tienen una responsabilidad específica, permite construir interfaces de usuario de una forma sencilla.
 - **Ventajas:**
 - Fácil aprendizaje.
 - Documentación adaptada.
 - Pesa poco.
 - Velocidad y flexibilidad.

- **Inconvenientes:**

- Escasa documentación que explique como incluirlo en nuestros proyectos.
- Comunidad más reducida que los anteriores (pero sigue subiendo).

2- Hay frameworks orientados a componentes, busca información y expón con tus propias palabras en que consiste.

Este tipo de frameworks, utilizan programación orientada a componentes consiguiendo que el código se desacople muchísimo y por tanto, facilita las labores de codificación y mantenimiento.

Un componente Web es una fracción de una página independiente totalmente del resto, que posee su propio html, css y JavaScript.

Por ejemplo un botón que queremos que al pulsarlo navegue hacia una Url en concreto, si este botón tiene que estar en diferentes páginas de mi aplicación y no utilizara componentes, tendría que escribir su código html, css y JavaScript en cada una de ellas y si en un futuro quisiera modificar la Url a la que apunta, tendría que cambiarlo en todas las páginas.

Si creara ese botón como un componente, podría incluirlo en todas las páginas escribiendo solo una línea de código y si en un futuro el botón tuviera que apuntar a otra Url, simplemente lo cambiaría en el componente y no tendría que modificar nada en las páginas.

Los componentes no solo se limitan a un elemento en particular como un botón, sino que puede ser uno conjunto de ellos. Por ejemplo un formulario que ya haga validaciones en sus campos.

En conclusión creo que trabajar en base a componentes es un gran avance que facilita la construcción y mantenimiento de aplicaciones Web.

3- React-JS es un framework para programación reactiva, busca información al respecto.

La programación reactiva es un paradigma de programación basado en datos que fluyen continuamente, los sistemas reactivos reaccionan a los datos ejecutando una serie de eventos.

Sigue el patrón de diseño Observer, cuando hay un cambio de estado en un objeto, los otros objetos son notificados y actualizados acorde. Por lo tanto, en lugar de sondear eventos para los cambios, los eventos se realizan de forma asíncrona para que los observadores puedan procesarlos.

El principal beneficio de utilizar la programación reactiva es que obtenemos una implementación débilmente acoplada, escalable y que tiende a aislar los fallos.

Gracias a su gran eficiencia, deriva en gastar menos dinero en servidores y centros de datos.

En conclusión la programación reactiva se puede considerar como el futuro próximo del desarrollo Web, cambiando las actuales arquitecturas de microservicios hacia arquitecturas completamente reactivas.