

**DESARROLLO DE SOFTWARE FRONTEND III**

**MARTINEZ COSIO JOSE ALFREDO**

**Aarón Hernández García**

**Renderización Angular (SSR y CSR)**

**programación reactiva RxJS**

**05**

**/0**

**8**

**/202**

**5**

**Inicio programación reactiva RxJS**

**Conceptos clave utilizados**

**1. interval()**

La función interval() de RxJS crea un observable que emite valores secuenciales (0, 1, 2, …) con un intervalo de tiempo determinado. En este caso, se usará con 2000 milisegundos (2 segundos).

**2. pipe()**

pipe() permite encadenar operadores para transformar o filtrar los valores que emite un observable. Es la forma en que se “procesan” los datos antes de usarlos.

**3. map()**

El operador map() sirve para transformar los valores emitidos por un observable. En este ejercicio, transformará los índices emitidos por interval() en frases.

**4. take()**

El operador take(n) se utiliza para limitar la cantidad de emisiones que hace un observable. Es importante aquí para detener automáticamente la emisión de frases después de mostrar la última.

Proyecto Angular Propuesto: **“Mi Sitio Simple”**

**1) SSR (Server-Side Rendering)**

**Server-Side Rendering** significa que el contenido de una página web es **generado por el servidor** y enviado al navegador ya procesado en formato HTML. Esto permite que el usuario vea el contenido casi de inmediato, ya que no necesita esperar a que el navegador cargue y ejecute JavaScript para mostrar la información.

**Ventajas:**

* Carga inicial más rápida (útil para SEO y usuarios con internet lento).
* Mejor indexación por parte de motores de búsqueda.
* Experiencia más fluida en dispositivos antiguos o con poca capacidad.

**Ejemplo:** Frameworks como **Next.js** en React o **Angular Universal** usan SSR.

**2) CSR (Client-Side Rendering)**

**Client-Side Rendering** significa que el navegador del usuario es el responsable de **construir y renderizar la página web** usando JavaScript. Al principio, el servidor solo entrega una estructura básica (HTML vacío), y luego el navegador descarga y ejecuta el código necesario para mostrar el contenido.

**Ventajas:**

* Mayor interactividad una vez cargada la aplicación.
* Transiciones rápidas entre páginas sin recargar completamente el sitio.
* Menor carga en el servidor.

**Ejemplo:** Aplicaciones creadas con **Angular**, **React**, o **Vue** que no usan SSR funcionan mediante CSR por defecto.