

# Documentación

*Sistemas y Tecnologías Web*

*María Marta Ramirez Gil - 21342*  
*Guatemala*

*Universidad del Valle de*

A continuación, se proporciona un manual de las tecnologías vistas en clase que se han utilizado en este proyecto:

## 1. React

React es una biblioteca de JavaScript para construir interfaces de usuario. Permite crear componentes reutilizables y gestionar el estado de la aplicación de manera eficiente. A continuación, se presentan los conceptos básicos de React:

- **Componentes:** Los componentes en React son bloques de construcción reutilizables que encapsulan una parte de la interfaz de usuario y su lógica. Pueden ser componentes funcionales o componentes de clase.
- **Estado (state):** El estado en React representa la información que puede cambiar durante la vida útil de un componente. Se utiliza para almacenar y gestionar datos dinámicos en la aplicación.
- **Props:** Los props (propiedades) son la forma en que los componentes en React reciben datos de sus componentes padres. Se utilizan para pasar datos de un componente a otro y permitir la comunicación entre ellos.
- **Renderizado:** React utiliza el concepto de Virtual DOM (DOM virtual) para realizar renderizado eficiente de los componentes. El Virtual DOM permite actualizar solo los elementos que han cambiado, lo que mejora el rendimiento de la aplicación.

## 2. Tailwind CSS

Tailwind CSS es un framework de CSS utilitario que permite crear estilos personalizados de manera eficiente. A continuación, se presentan los conceptos básicos de Tailwind CSS:

- **Clases utilitarias:** Tailwind CSS se basa en clases utilitarias predefinidas que se aplican directamente en el HTML. Estas clases permiten aplicar estilos y configuraciones específicas de manera rápida y sencilla.
- **Configuración personalizada:** Tailwind CSS se puede personalizar mediante la configuración de los estilos, colores, fuentes y más. Esto permite adaptar el framework a las necesidades del proyecto y mantener una coherencia visual.
- **Responsividad:** Tailwind CSS incluye clases utilitarias para crear diseños responsivos que se adapten a diferentes tamaños de pantalla. Esto facilita la creación de interfaces que se vean bien en dispositivos móviles y de escritorio.

### 3. WebPack

Webpack es un empaquetador de módulos utilizado para construir y empaquetar aplicaciones web. A continuación, se presentan los conceptos básicos de Webpack:

- **Configuración:** Webpack se configura mediante un archivo de configuración (webpack.config.js) en el que se definen las entradas, salidas, reglas y plugins del proyecto.
- **Módulos:** Webpack permite gestionar los módulos y dependencias del proyecto. Puede procesar diferentes tipos de archivos, como JavaScript, CSS, imágenes, entre otros, y generar un paquete optimizado.
- **División del código:** Webpack permite dividir el código en diferentes paquetes (chunks) para optimizar la carga de la aplicación. Esto mejora el tiempo de carga inicial y la eficiencia en la entrega de recursos al navegador.

### 4. JavaScript

JavaScript es un lenguaje de programación ampliamente utilizado en el desarrollo web. A continuación, se presentan los conceptos básicos de JavaScript:

- **Variables:** En JavaScript, se pueden declarar variables para almacenar y manipular datos. Las variables pueden contener diferentes tipos de datos, como números, cadenas, objetos, entre otros.

- **Funciones:** Las funciones en JavaScript permiten agrupar un conjunto de instrucciones y reutilizarlas en diferentes partes del código. Se pueden pasar argumentos a las funciones y devolver valores.
- **Eventos:** JavaScript permite controlar eventos, como clics de botón o envío de formularios, para realizar acciones específicas en respuesta a las interacciones del usuario.
- **DOM (Document Object Model):** El DOM es una representación en memoria de la estructura HTML de una página. JavaScript se utiliza para manipular el DOM y realizar cambios dinámicos en la página.