



Clase 10

Frontend

Agenda

Frontend

ReactJS

Svelte

Tailwind

The background of the slide is a dark, textured collage of financial data. It includes several line graphs with fluctuating lines, bar charts, and tables of numbers. Some numbers like '2.28', '345771', and '1000' are visible in a light blue font. The overall color palette is dark with hints of blue and orange.

Frontend

¿Qué es?, Funcionalidades, Tecnologías principales

¿Qué es?

El frontend es la parte visual e interactiva de una aplicación o sitio web, es decir, todo lo que el usuario final ve y utiliza directamente. Se encarga de proporcionar una experiencia de usuario (UX) agradable, fluida y funcional mediante la combinación de tecnologías como HTML, CSS y JavaScript.

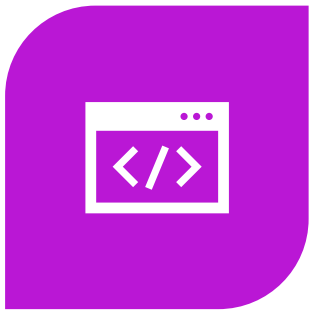


Funcionalidades

- Interfaz de usuario (UI)
- Interacción con el usuario
- Comunicación con el backend
- Optimización y accesibilidad
- Responsivo



Tecnologías principales



HTML (HYPERTEXT MARKUP
LANGUAGE)



CSS (CASCAIDING STYLE SHEET)



JAVASCRIPT



FRAMEWORKS

HTML



HTML

HTML, acrónimo en inglés de HyperText Markup Language, hace referencia al lenguaje de marcado utilizado en la creación de páginas web. Este estándar que sirve de referencia del software que interactúa con la elaboración de páginas web en sus diferentes versiones

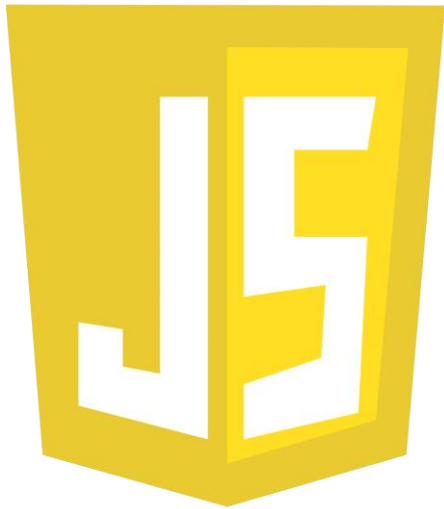
CSS

CSS, en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

CSS



JavaScript



JavaScript

Es un lenguaje de programación dinámico y versátil utilizado principalmente para agregar interactividad a sitios web. Es ampliamente compatible con navegadores y permite crear aplicaciones web modernas, dinámicas y basadas en eventos.

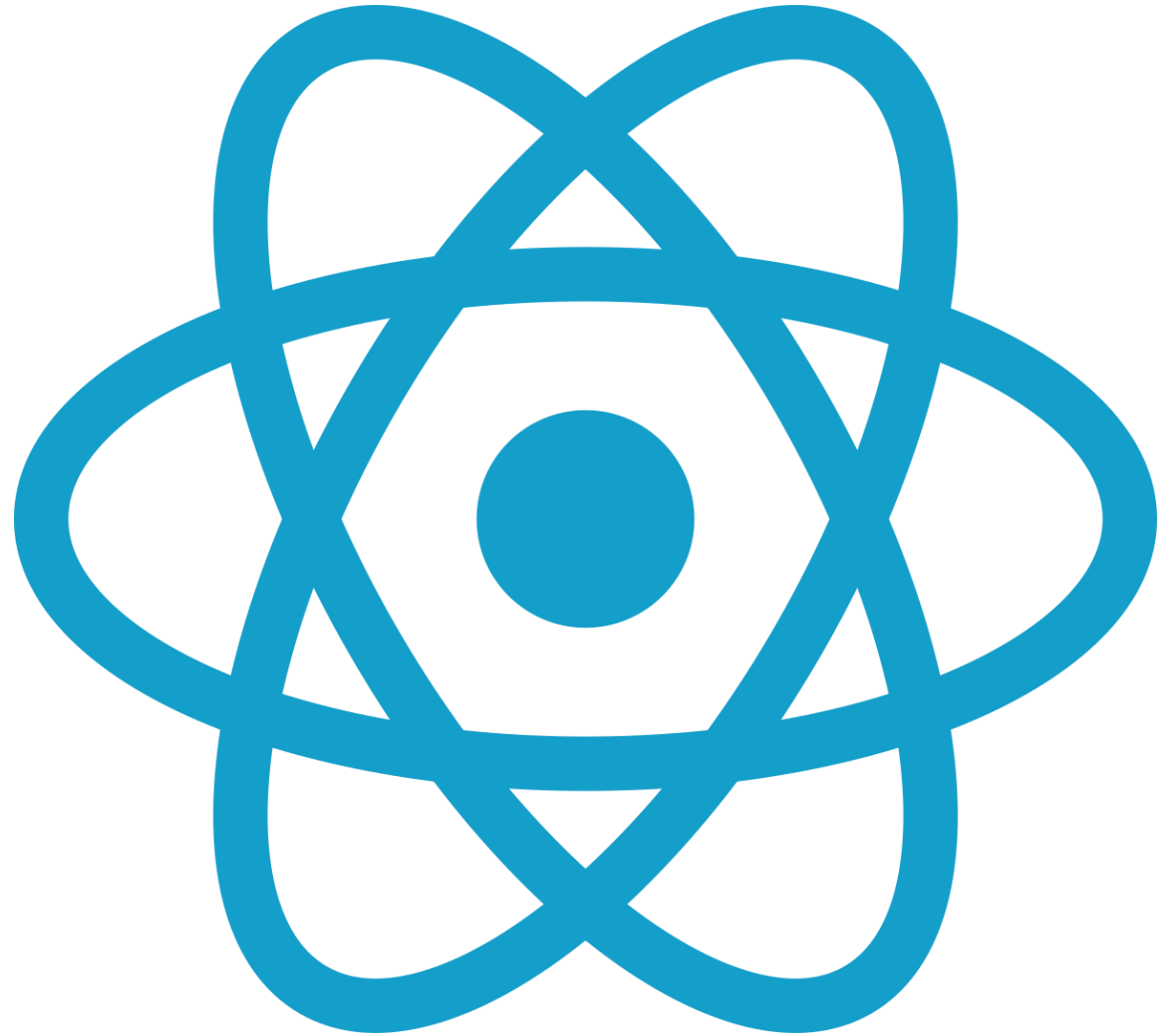


TypeScript

Es un superconjunto de JavaScript que agrega tipado estático y otras características avanzadas para mejorar la robustez y mantenibilidad del código. Es ideal para proyectos grandes y se compila en JavaScript para su ejecución.

React

Es una librería de JavaScript desarrollada por Facebook para construir interfaces de usuario dinámicas y basadas en componentes. Se centra en la gestión eficiente del estado y el DOM virtual, lo que permite actualizar solo las partes de la interfaz que cambian. Ideal para construir aplicaciones web modernas y escalables.



Vite

Es un build tool rápido y ligero para proyectos frontend. Proporciona un entorno de desarrollo con recarga en caliente (HMR) y compilación optimizada para producción. Compatible con frameworks como React, Vue y Svelte, su velocidad proviene de usar ES Modules durante el desarrollo y herramientas como esbuild para la precompilación.



Svelte

Es un framework de JavaScript que compila componentes en código JavaScript puro, eliminando la necesidad de un DOM virtual. Esto permite aplicaciones más rápidas y ligeras con menos recursos. Es conocido por su simplicidad y sintaxis intuitiva, ideal para proyectos con rendimiento crítico.



Vue

Es un framework progresivo de JavaScript para construir interfaces de usuario. Combina características de React y Angular, como componentes y directivas, con una curva de aprendizaje más suave. Su enfoque modular lo hace adecuado para proyectos pequeños y grandes por igual.



Angular

Es un framework completo desarrollado por Google para crear aplicaciones web robustas y escalables. Usa TypeScript de manera nativa, implementa un enfoque basado en componentes y cuenta con herramientas como inyección de dependencias y un robusto sistema de formularios. Es ideal para aplicaciones empresariales y de gran escala.






Creación de proyectos

Tener instalado previamente NodeJS y npm, en caso de usar angular ng (npm install -g @angular/cli)

- React: `npm create react app <<nombre_proyecto>>`
- Vite: `npm create vite@latest <<nombre_proyecto>>`
- Svelte: `npm create vite@latest <<nombre_proyecto>> -- --template svelte`
- Vue: `npm create vue@latest <<nombre_proyecto>>`
- Angular: `ng new <<nombre_proyecto>>`



Hooks

De Estado, Contexto, Referencia, Efectos, Performance, Recursos,
Customizables



Hooks de Estado

Un hook de estado permite que un componente "recuerde" información como la entrada del usuario. Por ejemplo, un componente de formulario puede usar el estado para almacenar el valor de entrada, mientras que un componente de galería de imágenes puede usar el estado para almacenar el índice de la imagen seleccionada.

- React: `useState`, `useReducer`
- Vite (React): `useState`, `useReducer`
- Svelte: Svelte tiene un sistema de estado reactivo integrado. Puedes declarar variables locales o stores y Svelte se encargará de actualizar automáticamente la interfaz.
- Vue: Vue utiliza un sistema de datos reactivo con variables reactivas y el Composition API.
- Angular: Angular maneja el estado a través de Services y observables de RxJS. También permite manejar estados locales con variables dentro de los componentes.



Hooks de Contexto

Un hook de contexto permite que un componente reciba información de padres lejanos sin pasarla como accesorios. Por ejemplo, el componente de nivel superior de su aplicación puede pasar el tema de la interfaz de usuario actual a todos los componentes siguientes, sin importar cuán profundos sean.



Hooks de Referencia

Los hooks de referencias permiten que un componente contenga información que no se utiliza para la renderización, como un nodo DOM o un ID de tiempo de espera. A diferencia del estado, actualizar una referencia no vuelve a representar su componente. Las referencias son una «escotilla de escape» del paradigma React. Son útiles cuando necesita trabajar con sistemas que no son React, como las API del navegador integradas.



Hooks de Efectos

Los hooks de efectos permiten que un componente se conecte y sincronice con sistemas externos. Esto incluye lidiar con la red, el DOM del navegador, animaciones, widgets escritos usando una biblioteca de UI diferente y otro código que no sea de React.



Hooks de Performance

Los hooks de performance es una forma común de optimizar el rendimiento de la renderización es omitir el trabajo innecesario. Por ejemplo, puede decirle a React que reutilice un cálculo almacenado en caché o que omita una nueva renderización si los datos no han cambiado desde la renderización anterior.



Hooks de Recursos

Un componente puede acceder a los recursos sin tenerlos como parte de su estado. Por ejemplo, un componente puede leer un mensaje de una Promesa o leer información de estilo de un contexto.



Hooks Customizables

Construir tus propios Hooks te permite extraer la lógica del componente en funciones reutilizables.

Un Hook personalizado es una función de JavaScript cuyo nombre comienza con "use" y que puede llamar a otros Hooks.

Tailwind CSS

Tailwind CSS es un framework de CSS de código abierto para el diseño de páginas web. La principal característica de esta biblioteca es que, a diferencia de otras como Bootstrap, no genera una serie de clases predefinidas para elementos como botones o tablas.





Dudas o Comentarios
