

## **Taller 1 React**

**Luisa Fernanda Triviño Gil**

**CC: 1030612775**

**Estudiante Sena ADSO**

### **Ejercicio:**

#### **1. Investiga la historia de React y menciona dos hitos importantes en su desarrollo.**

En 2021 ReactJS superó a jQuery y se convirtió en el framework web más utilizado por los desarrolladores de software en todo el mundo. Facebook y las comunidades de desarrolladores de todo el mundo mantienen ReactJS.

React JS proporciona la facilidad de optimizar todos los componentes reutilizables que permite a los desarrolladores de aplicaciones incluir las nuevas partes en el desarrollo de sus aplicaciones. Un desarrollador de aplicaciones puede ver el progreso oportuno de todo el proyecto cuando se utiliza ReactJS.

#### **2. Escribe una breve explicación sobre por qué Facebook decidió crear React.**

Nace de la necesidad de crear un código que funcione de manera eficiente y productiva, sea fácil de usar y entender. El código usado anteriormente causaba inconvenientes al momento de su actualización.

### **Ejercicio:**

#### **3. Menciona tres ventajas de usar React en el desarrollo de aplicaciones web.**

- ReactJS es un framework front-end muy sencillo y fácil de entender en comparación con otros como Vue o Angular.
- Es utilizado por los desarrolladores para crear software que funciona bien tanto en Android como en iOS. Facilita el uso de los códigos reutilizados ahorrando así el tiempo de escribir el nuevo código.
- ReactJS viene con JSX, una popular extensión sintáctica, el cual es muy utilizado por los desarrolladores para escribir componentes personalizados, crear árboles de elementos React a partir de maquetas HTML y desarrollar aplicaciones de gran volumen

#### **4. Explica cómo el Virtual DOM mejora el rendimiento de una aplicación.**

Es una característica clave de React que mejora el rendimiento y las actualizaciones eficientes de la interfaz de usuario. Con esto, React puede realizar actualizaciones por lotes, minimizar los reflujos y los repintados, y aplicar los cambios de manera eficiente. Esto hace que las actualizaciones de la interfaz de usuario sean rápidas y fluidas, lo que proporciona una mejor experiencia de usuario en el proceso.

#### **Ejercicio:**

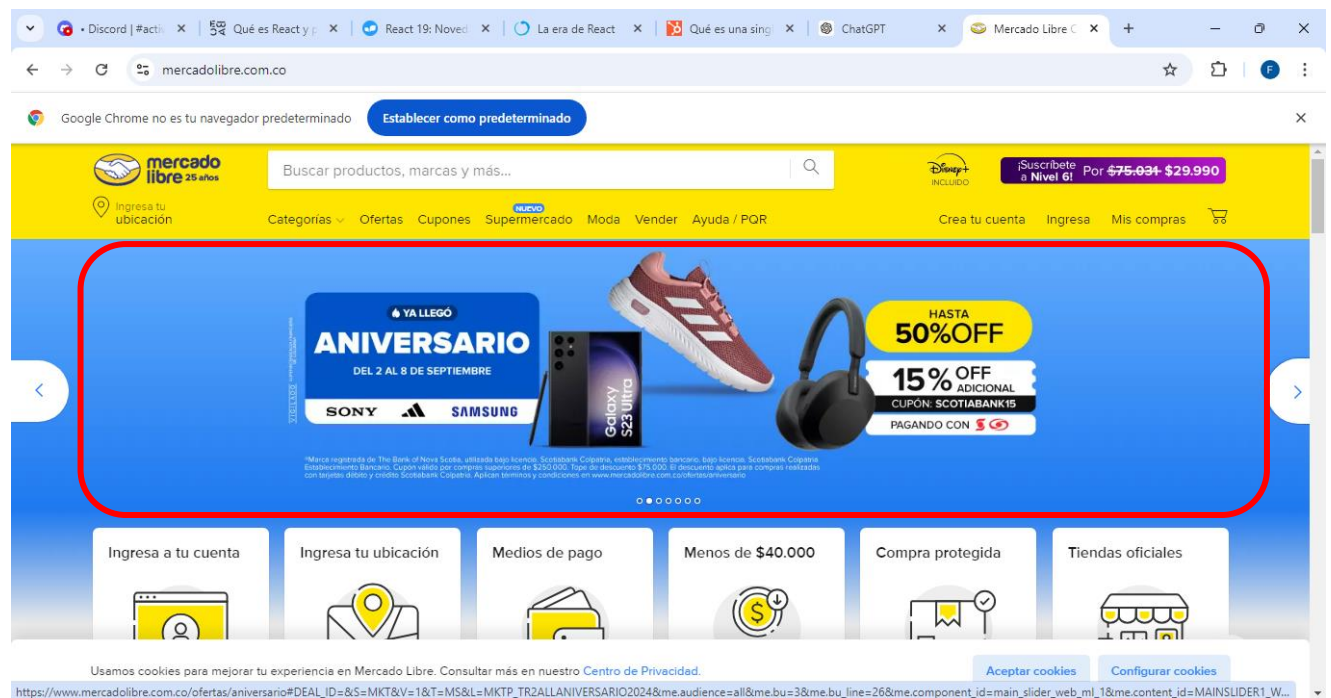
**5. Define qué es una Single Page Application (SPA):** Es un tipo de aplicación web que carga una sola página HTML y actualiza dinámicamente el contenido a medida que el usuario interactúa con la aplicación.

#### **6. Explica cómo React facilita la creación de una SPA.**

#### **7. Proporciona un ejemplo de cómo un componente de React puede actualizar la interfaz sin recargar la página.**

Su característica principal es que proporciona una experiencia de usuario fluida y rápida, similar a una aplicación de escritorio, pues los cambios de contenido ocurren de forma instantánea y sin interrupciones. Esto se logra mediante la manipulación del DOM (Modelo de Objetos del Documento) y el enrutamiento interno en la app.

Ejemplo: Lo señalado en el recuadro es un ejemplo de componente en la página de Inicio de Mercado libre que se actualiza, sin necesidad de actualizar toda la página.

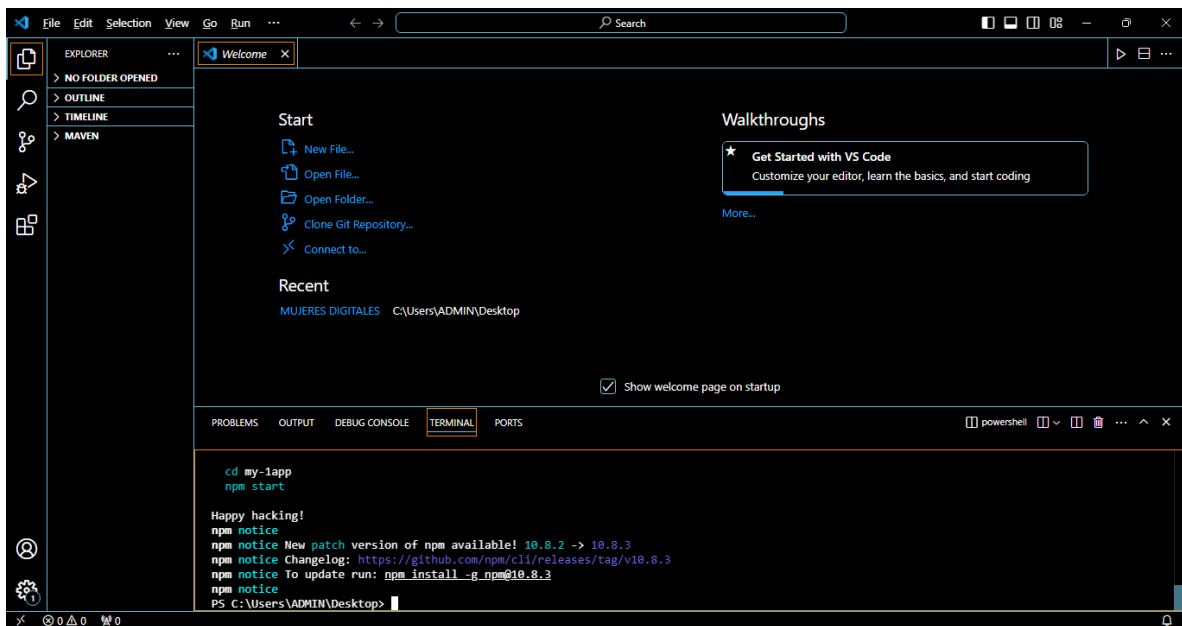
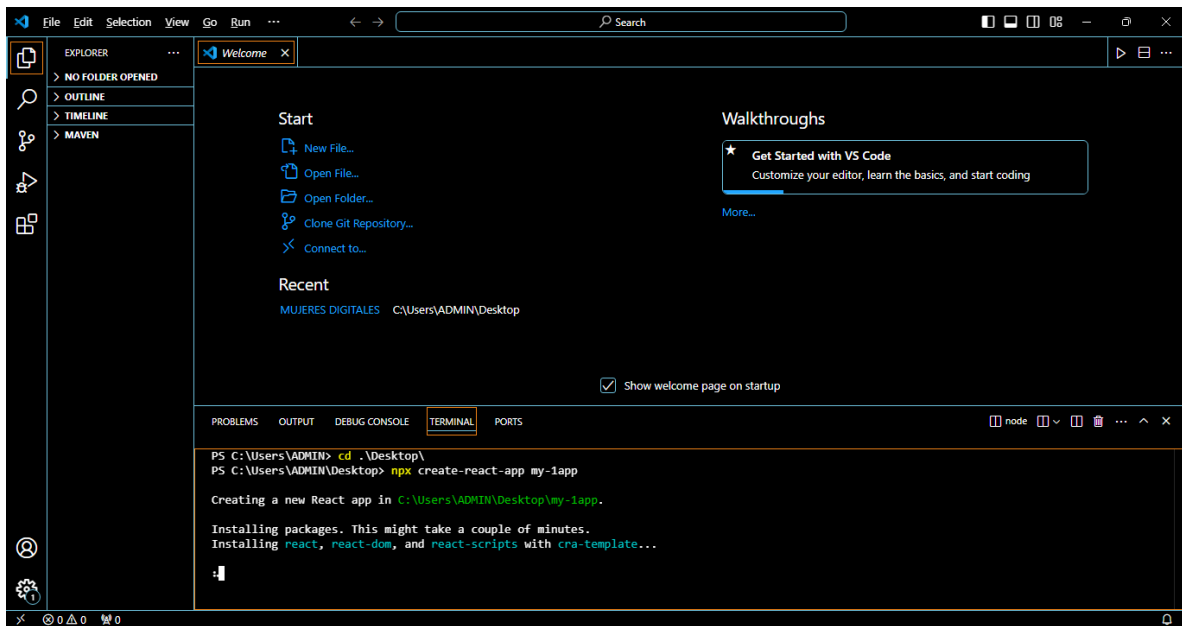


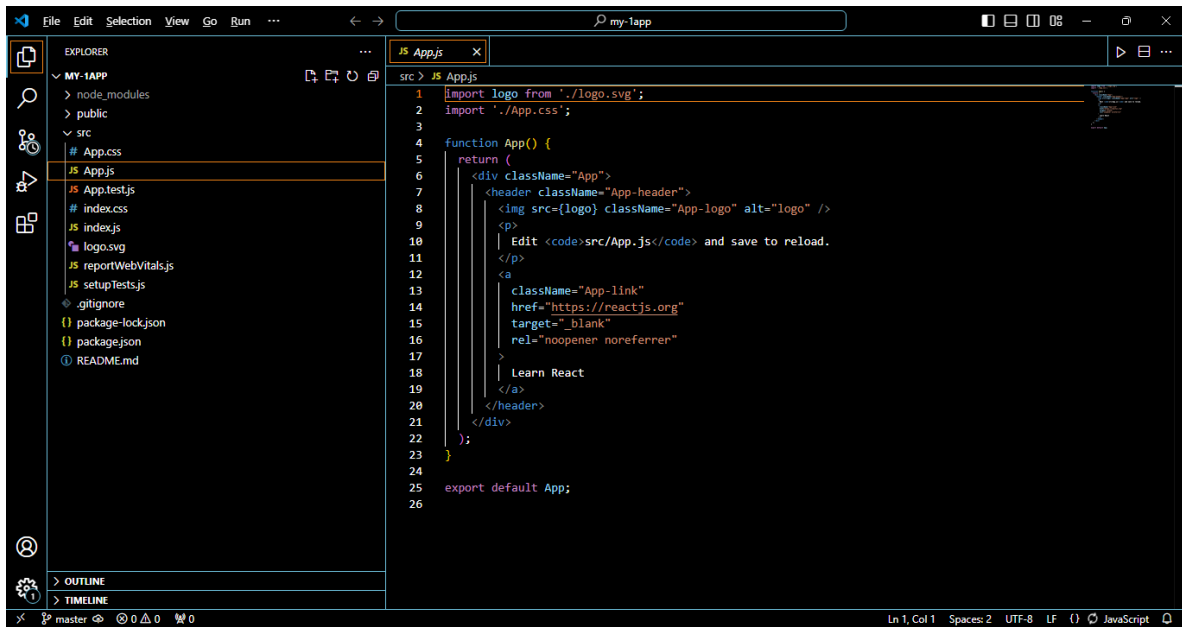
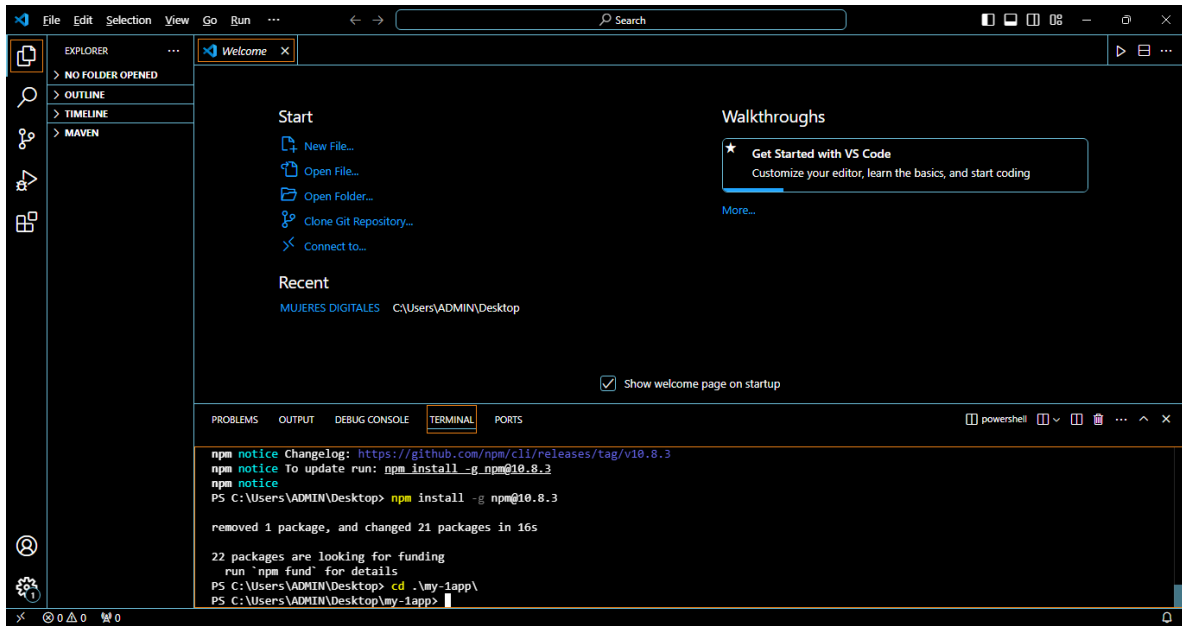
## 8. Crea un nuevo proyecto React utilizando Create React App

Se crea App en el Desktop con el nombre my-1app

## 9. Inicia el servidor de desarrollo y comparte una captura de pantalla de tu proyecto en ejecución

Se relacionan capturas de creación de App en React





The screenshot shows the VS Code editor with the `App.js` file open. The file content is as follows:

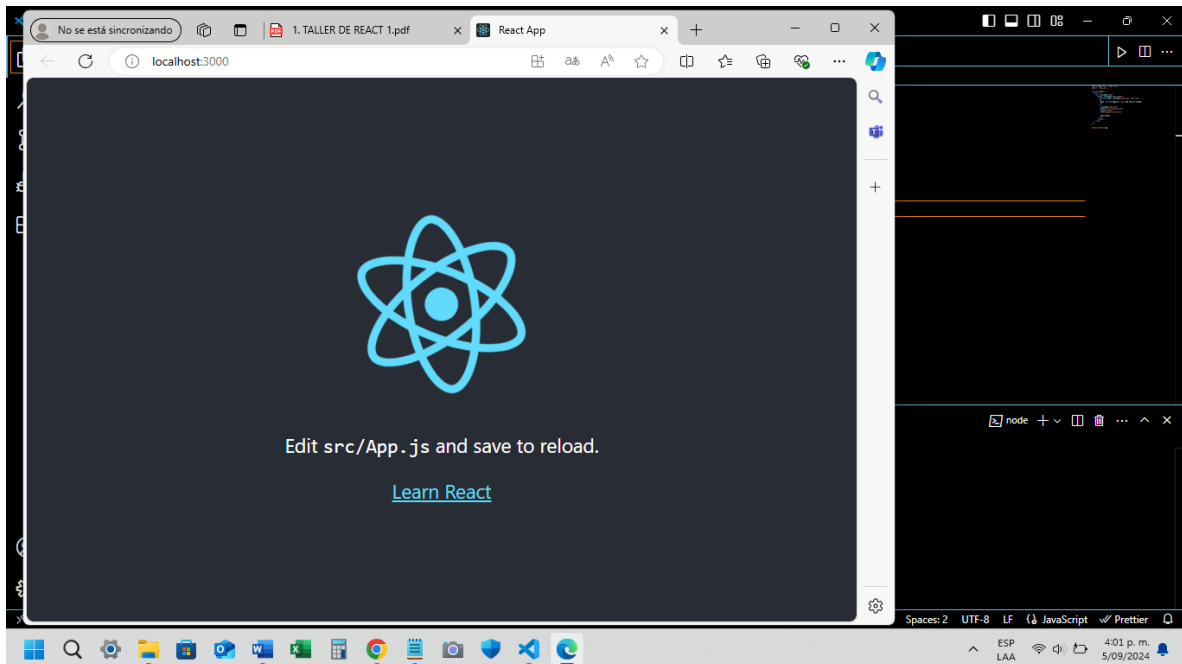
```
1 import logo from './logo.svg';
2 import './App.css';
3
4 function App() {
5   return (
6     <div className="App">
7       <header className="App-header">
8         <img src={logo} className="App-logo" alt="logo" />
9       </p>
10      <p>
11        Edit <code>src/App.js</code> and save to reload.
12      </p>
13      <a
14        className="App-link"
15        href="https://reactjs.org"
16        target="_blank"
17        rel="noopener noreferrer"
18      >
19        Learn React
20      </a>
21    </div>
22  );
23}
```

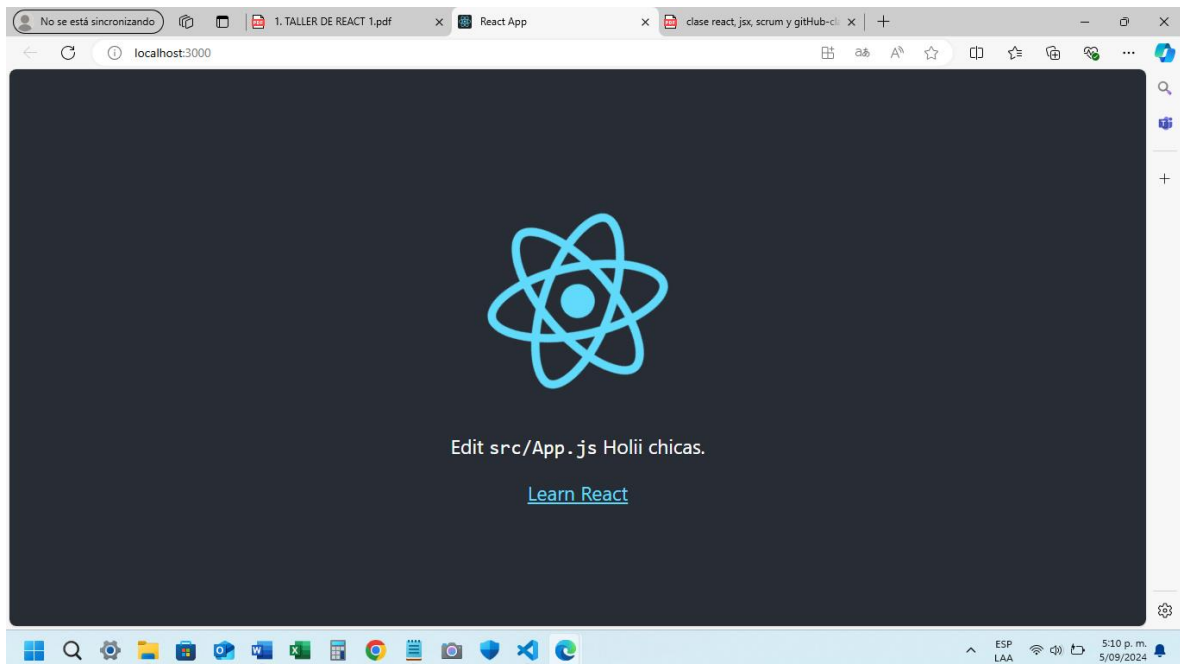
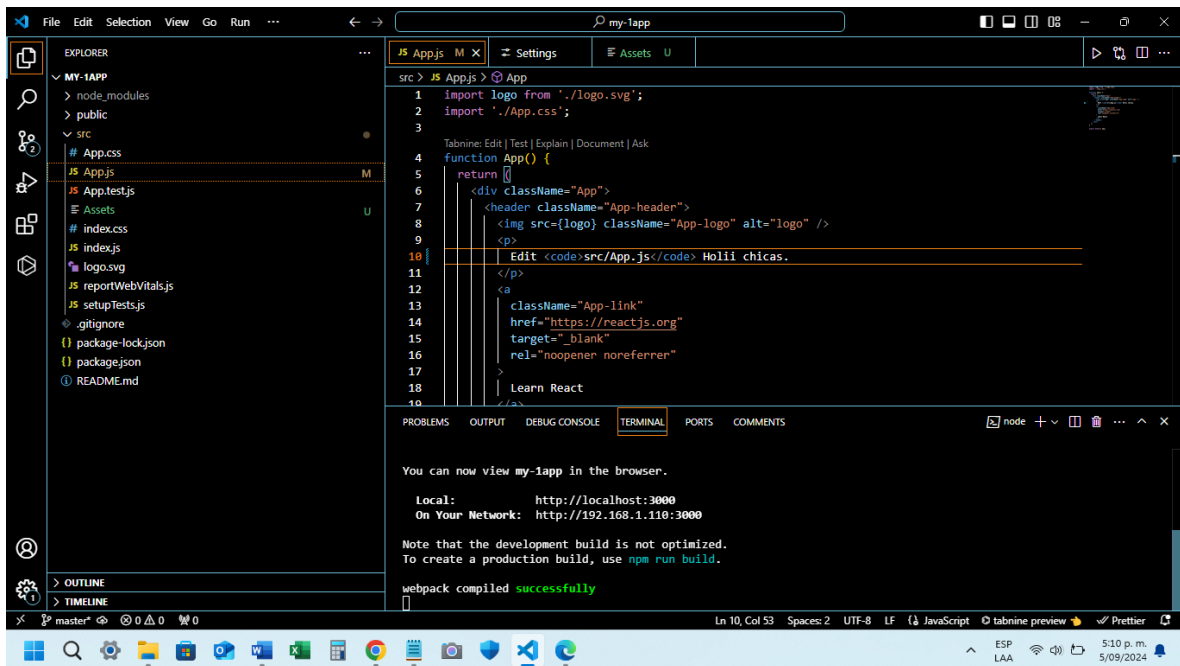
The terminal output at the bottom shows the following messages:

```
You can now view my-lapp in the browser.
Local:      http://localhost:3000
On Your Network: http://192.168.1.110:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```





## 10. Explica brevemente el propósito de las carpetas src y public en un proyecto React.

La carpeta SRC contiene el código fuente de tu aplicación React, se encuentran los archivos JavaScript, los estilos CSS, y otros archivos que forman la lógica y la interfaz de usuario de tu aplicación.

La carpeta Public se puede colocar archivos HTML, imágenes, fuentes, y otros recursos que deben estar disponibles directamente en el navegador. El archivo index.html dentro de public es el punto de entrada principal de la aplicación React.

### **11. Explica cómo JSX se diferencia del HTML tradicional.**

JSX combina la flexibilidad de JavaScript con una sintaxis similar a HTML, permitiendo una integración más estrecha entre la lógica y el marcado, mientras que el HTML tradicional es un lenguaje de marcado estático.

### **12. Define los roles principales en un equipo SCRUM**

- Scrum Master: Facilita el proceso SCRUM, elimina obstáculos y asegura que el equipo siga las prácticas SCRUM.
- Product Owner: Responsable de definir la visión del producto y gestionar el Product Backlog.
- Equipo de Desarrollo: Se encarga de construir y entregar incrementos del producto en cada sprint.

### **13. Explica qué es un sprint y cómo se planifica.**

Un sprint es un ciclo de desarrollo de duración fija entre 2 y 4 semanas, tiempo en el cual el equipo de desarrollo trabaja para completar una cantidad de tareas específicas.

Se planifica realizando reuniones de la siguiente manera, con el fin de brindar un seguimiento oportuno y real del desarrollo del proyecto.

- ✓ Daily Stand-ups: Reuniones diarias breves en las que el cada miembro del equipo comparte las labores realizadas del día anterior y el día actual.
- ✓ Sprint Planning: Reunión al inicio de cada sprint para decidir que tareas se abordan durante el sprint.
- ✓ Sprint Review: Reunión al final de cada sprint para demostrar lo que se ha completado y recibir retroalimentación.
- ✓ Sprint Retrospective: Reunión para reflexionar sobre lo que salió bien y mal y como mejorar en el próximo sprint.