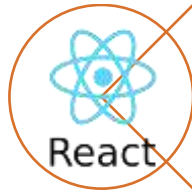


Actividad de Aprendizaje N° 06A

Introducción al Desarrollo con React

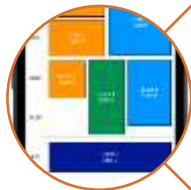
1. Qué es React: Framework Fronted



React es una *biblioteca de JavaScript* para crear *interfaces de usuario*.



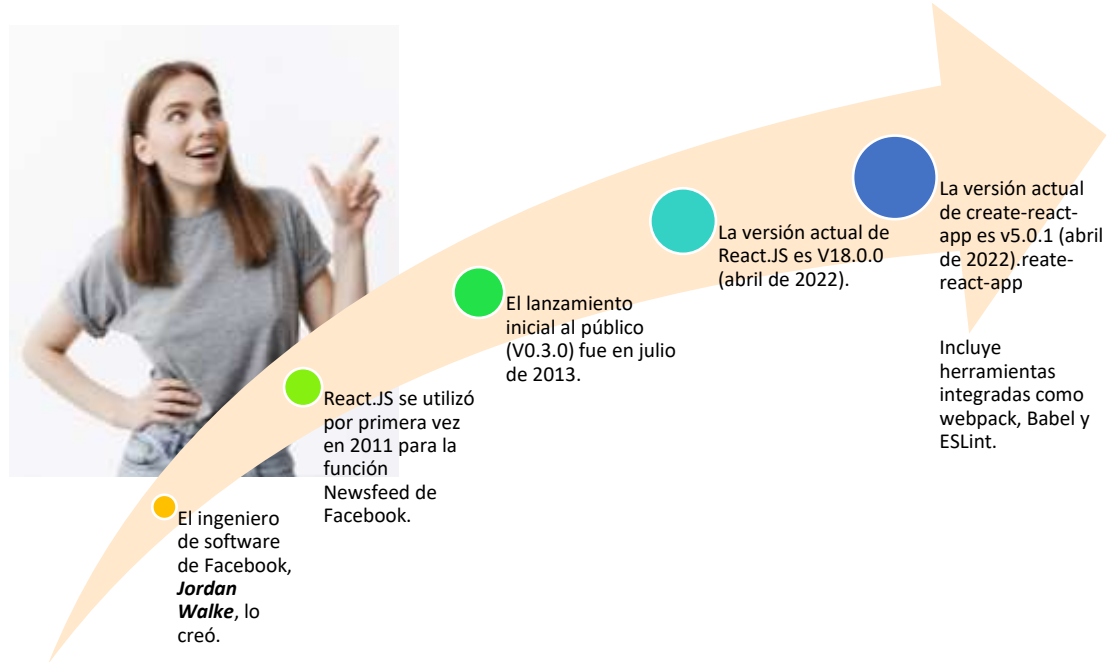
React se utiliza para crear *SPA* aplicaciones de una sola página.



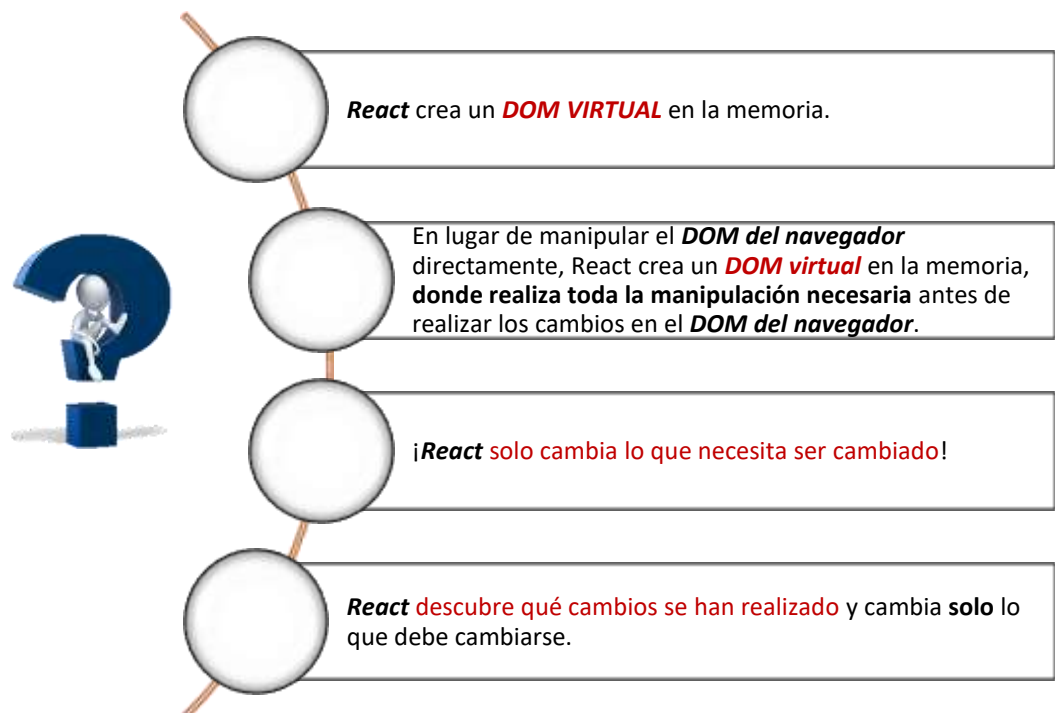
React nos permite crear *componentes* de interfaz de usuario reutilizables.



2. Historia de React.JS



3. ¿Cómo funciona React?



4. Documentación de React

La documentación de React JS por ahora podemos encontrarla en: <https://es.reactjs.org/>



Link: <https://es.reactjs.org/docs/getting-started.html>



5. Herramientas de React

Para trabajar con React JS y cualquier framework es conveniente conocer las herramientas que se utiliza para creación de un proyecto, para compilación, transpilación, y diferentes tareas necesarias para gestionar de forma adecuada la aplicación.

Ecosistema de Desarrollo con React - Dependencias

Incluso podríamos crear nuestros propios ambientes de desarrollo, pero no es conveniente porque la diversidad de configuraciones genera mayor complejidad. Algunas herramientas que se utilizan son:

Babel, Parcel, Webpack

Babel JS

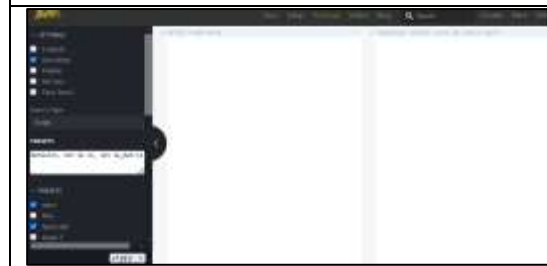


Babel es una herramienta que transforma nuestro código JS de última generación (o con funcionalidades extras) a un código de Javascript que cualquier navegador o versión de Node.js pueda entender.

Tranforma *Typescrip*, *JSX*, etc

<https://babeljs.io/>

Link Demostración: <https://babeljs.io/repl>



1. Abrir Link
2. Agregar código

```
let nombre: string = `Jaime Suasnabar`;  
let estatura: number = 1.72;  
let sentence: string = `Hola, mi nombre  
es ${ name }. Mi estatura es ${ age + 1  
} metros.`;
```

3. Explicación

ESLint



Es una herramienta de software que revisa y "observa" tu código en busca de errores que puedan afectar tu código. Algunos "linteres" incluso pueden darte sugerencias de como arreglar el error o incluso arreglarlo ellos mismos..

<https://eslint.org/>

Link Demostración: <https://eslint.org/demo>



1. Abrir Link
2. Agregar código

```
var v = confirm("¿Quieres enviarlo?");  
if ( v ) {  
    document.write("Ok, gracias.");  
} else {  
    document.write("Estaremos aquí para  
cuando te decidas ;)");  
}
```

3. }Explicación

VSCode tiene una extensión **ESLint**

Sass / Scss



SASS *Syntactically Awesome Stylesheets* (definido como CSS con superpoderes) es un metalenguaje de hojas de estilo. Éste es un lenguaje de script que es traducido a CSS.

SASS es el pre-procesador, mientras que SCSS es la sintaxis de CSS pero con todas las ventajas de SASS. <https://sass-lang.com/>

Link: <https://beautifytools.com/scss-compiler.php>



1. Abrir Link

2. Agregar código

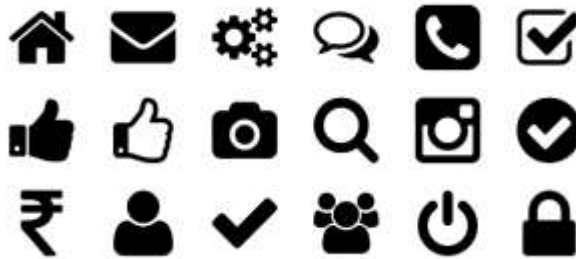
```
$primary-color: #333;

.navbar {
  background-color: $primary-color;
}

.footer {
  border: 1px solid $primary-color;
}
```

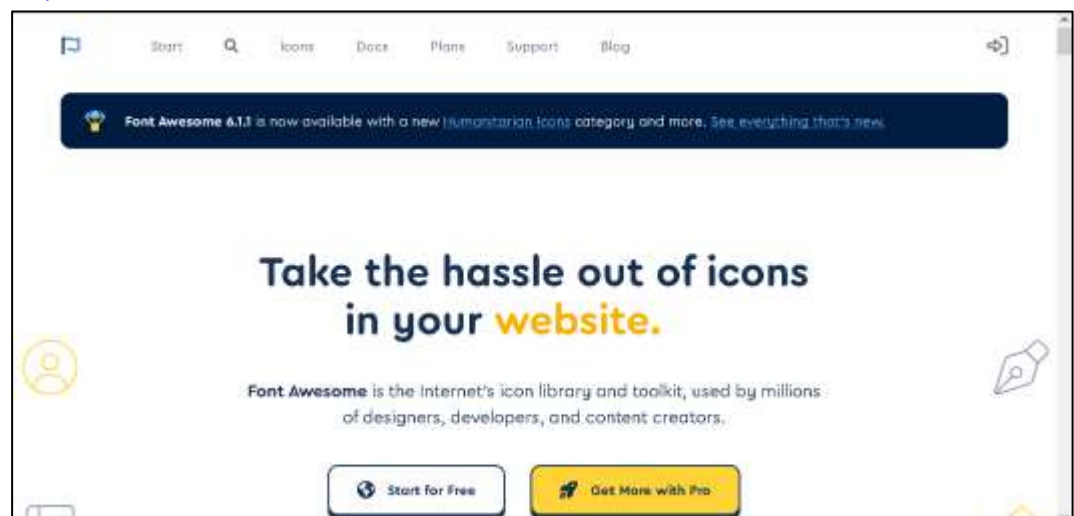
3. Explicación

Font Awesome



Font Awesome se utiliza para sustituir **imágenes de iconos** comunes por *gráficos vectoriales* convertidos en *fuentes*. No es necesario el uso de Javascript para su funcionamiento y dispone de una versión gratuita y otra de pago a CSS.

<https://fontawesome.com/>



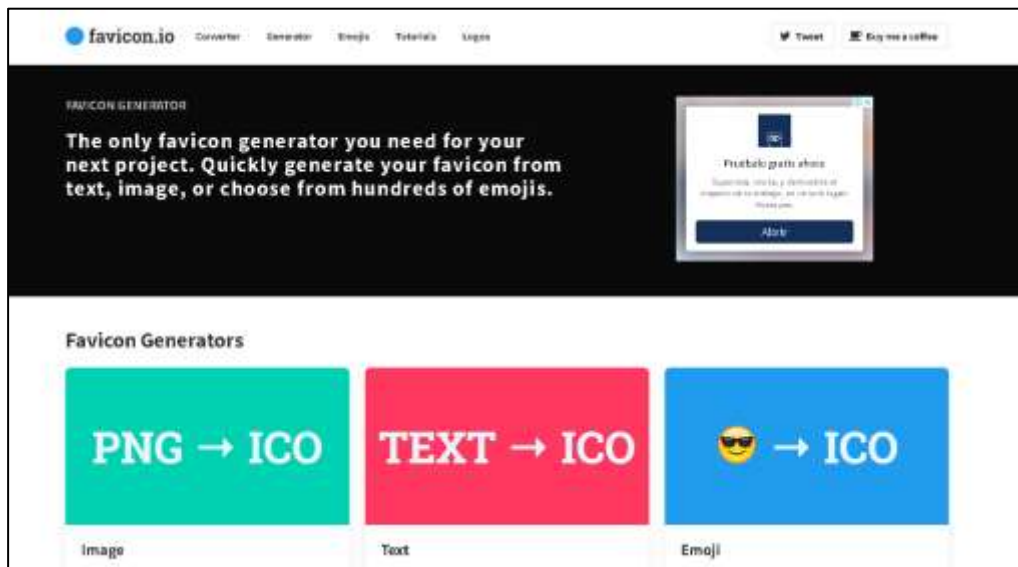
Otras

- MaterialDesign Icons
- FlatIcon

Favicon Icon

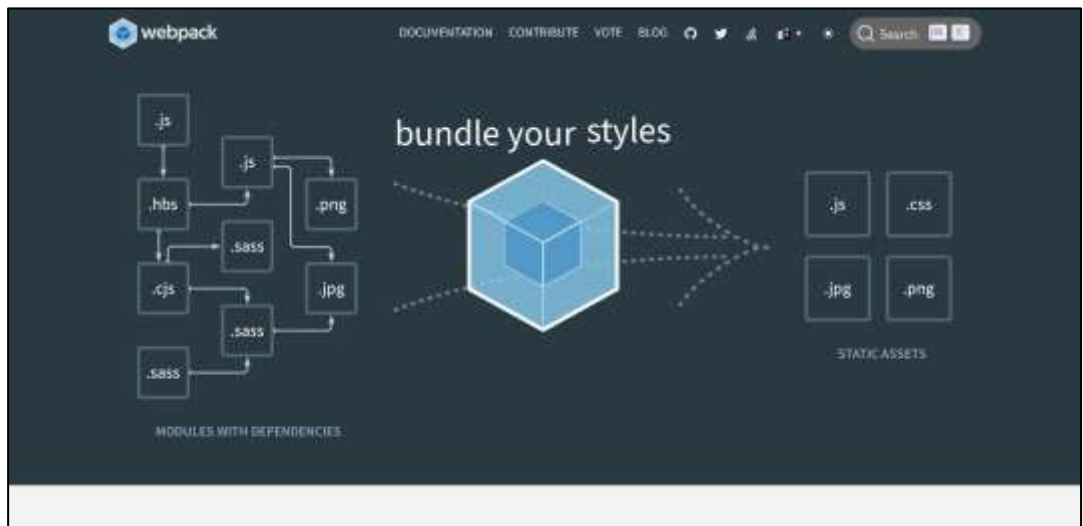


Un favicon es un **pequeño icono** (normalmente de 16×16 píxeles o de 32×32 px) que se utiliza para *identificar un sitio web* en las **pestañas de un navegador**, en un listado de favoritos, o en cualquier otra parte que requiera una identificación pequeña. <https://favicon.io/>



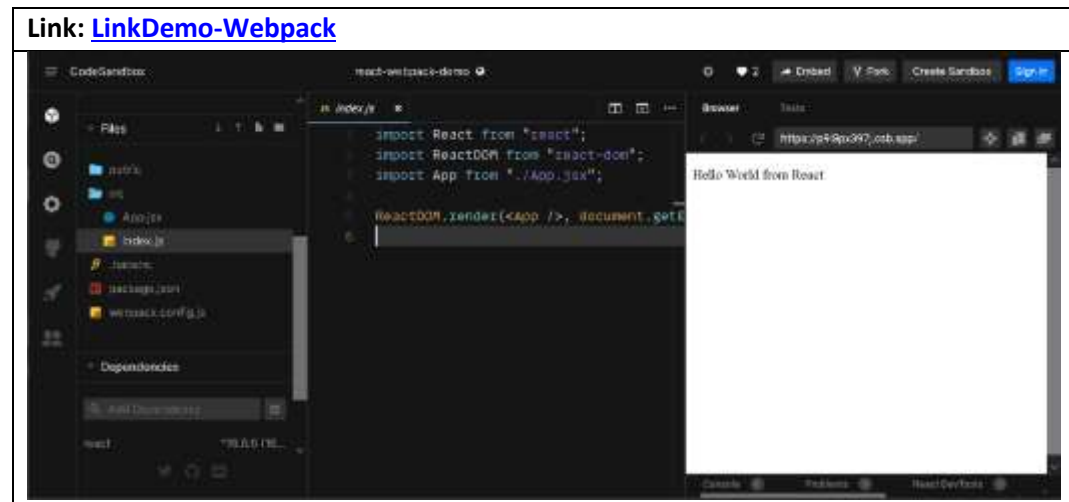
Webpack

Actualmente el desarrollo web utiliza múltiples dependencias, por lo que Webpack convierte esas dependencias en archivos que pueda entender el navegador de una manera más rápida e interoperable.



Para el desarrollador es mucho más cómodo trabajar con archivos separados y con un diseño arquitectónico modular. Es común que una aplicación web contenga archivos css en plano, archivos de preprocesadores como SaaS o Less, archivos en TypeScript, entre otros. Webpack nos ayudará a transpilar todos estos archivos y entregarlos al navegador para que este los entienda.

<https://webpack.js.org/>



Otras Dependencias

- Google Maps
- Google Fonts
- Google Website Optimizer
- Google Analytics
- Google Trends

6. Herramientas de Instalación de React

Si hiciste una aplicación medianamente grande con React.js seguramente tuviste que configurar Babel, Webpack, un linter (ESLint?), un servidor de render, etc, etc., y de seguro sufriste la conocida Fatiga de JavaScript.

Manualmente es muy tedioso y requiere revisar información de compatibilidad de las diferentes versiones, por lo que se hace necesario el uso de herramientas de instalación.

Se usa herramientas con el objetivo de ***NO tener que preocuparte por la configuración***

Las principales herramientas son:

- Create-react-app
- Vite
- Remix Run
- NextJS
- GatsbyJS
- BlitzJS
- Hydrogen

Este este documento se revisarán sólo las 3 primeras herramientas.

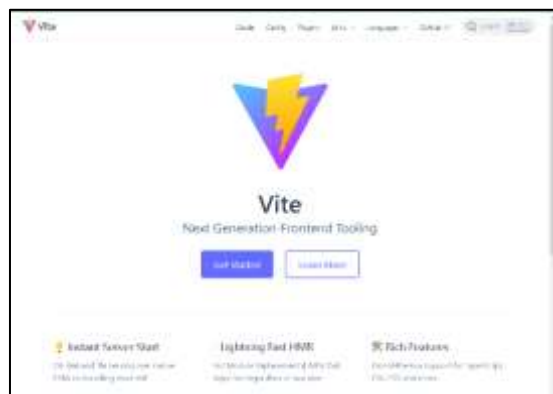
Create React App:



<https://create-react-app.dev/>

Configura tu ambiente de desarrollo de forma que puedas usar las últimas características de Javascript, para una buena experiencia de desarrollo, y optimizando tu aplicación para producción. Requiere Node $\geq 14.0.0$ y npm ≥ 5.6 .

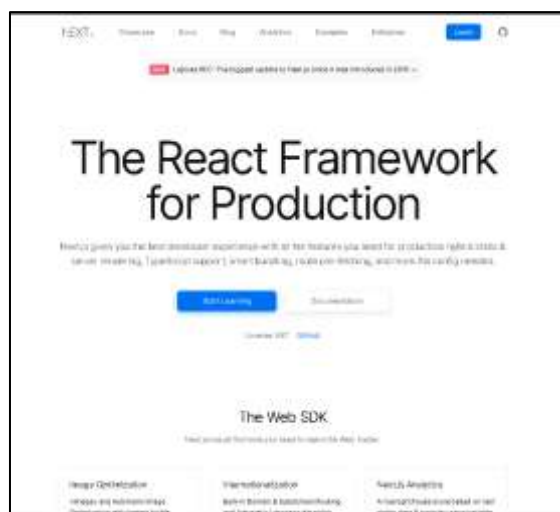
Vite:



<https://vitejs.dev/>

Vite se define como una herramienta de *frontend* que te ayudará a *crear tus proyectos* de forma *agnóstica* (sin atarte a ningún framework concreto) y que su desarrollo y construcción final sea lo más sencilla y cómoda posible. Está desarrollada por Evan You, el creador de Vue.

Next.js:



<https://nextjs.org/>

Next.js es un pequeño framework construido sobre React.js que viene a ayudar a reducir esta fatiga. ¿Cómo? Next nos permite, instalando una sola dependencia, tener configurado todo lo que necesitamos para crear una aplicación de React usando Babel, Webpack, server render y muchas otras técnicas como HMR o separación de código y... ¡hasta hace más fácil hacer deploy de nuestras aplicaciones!

Primera Unidad: Desarrollo Web Fronted
GUÍA DE LABORATORIO N° 05:
INSTALACIÓN DE REACT CON DIFERENTES HERRAMIENTAS

Docente: Jaime Suasnabar Terrel

Fecha:

Duración: 90 minutos

Instrucciones: Instalar NodeJS y React de forma eficiente mediante las herramientas Vite, Remix, NextJS y Create React App.

OBJETIVOS

- Conocer el funcionamiento de React
- Instalar NodeJS
- Instalar React de mediante las herramientas Vite, Remix, NextJS y Create React App.
- Comparar las estructuras de directorio

PROCEDIMIENTO

1. Primeros Pasos en React



Para obtener una *descripción general* de lo que es **React**, puede **escribir código React directamente en HTML**.

Nota: Pero para usar React en producción, necesita npm y Node.js instalados.

Enlaces CDN

Para usar React necesita dos librerías: React y ReactDOM están disponibles a través de un CDN en su versión 17 de desarrollo.

```
<script src="https://unpkg.com/react@17/umd/react.development.js"
crossorigin></script>
<script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"
crossorigin></script>
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
```

El CDN versión 18 de React están disponibles en:

```
<script crossorigin
src="https://unpkg.com/react@18/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-
dom.development.js"></script>
```

React directamente en HTML

1. Comience por incluir tres scripts de la version 17, los dos primeros nos permiten escribir código React en nuestros JavaScripts, y el tercero, Babel, nos permite escribir sintaxis JSX y ES6 en navegadores más antiguos.
2. Agregar la etiqueta `<div id="mydiv"></div>`
3. Agregar la etiqueta `<script type="text/babel">...</script>`

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Page 01</title>
  <script src="https://unpkg.com/react@17/umd/react.development.js" crossorigin></script>
  <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js" crossorigin></script>
  <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
</head>
<body>
  <div id="mydiv"></div>

  <script type="text/babel">
    function Hola() {
      let txt = " por Jaime";
      return <h1>Hola Mundo! {txt} </h1>;
    }

    ReactDOM.render(<Hola />, document.getElementById('mydiv'))
  </script>
</body>
</html>
```

4. Probar la pagina.

2. Node JS – Instalación

Ingresar a la web: <https://nodejs.org/es/>



Descargar el archivo versión LTS

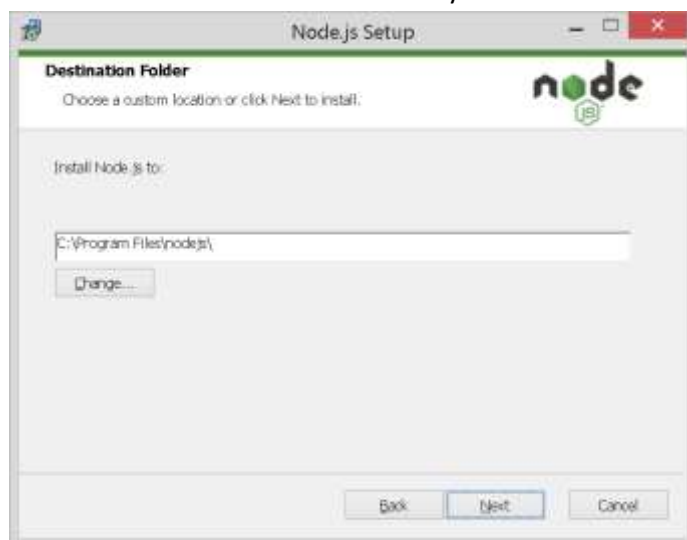
Seguir el asistente de instalación, pulse en next



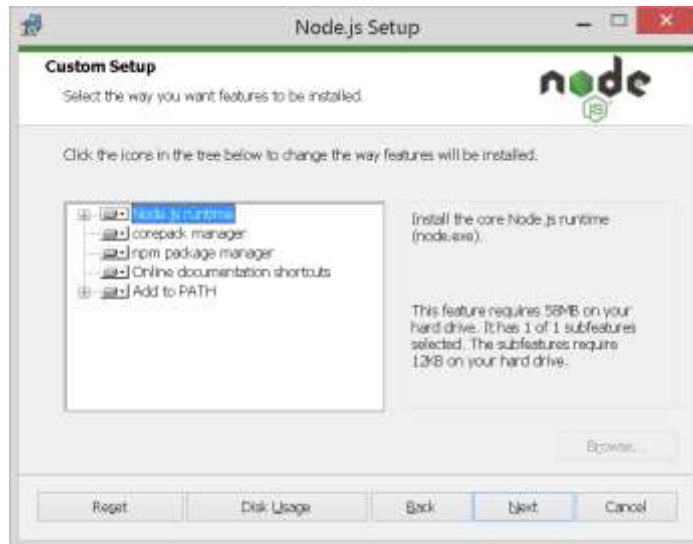
Aceptar la licencia de uso y next



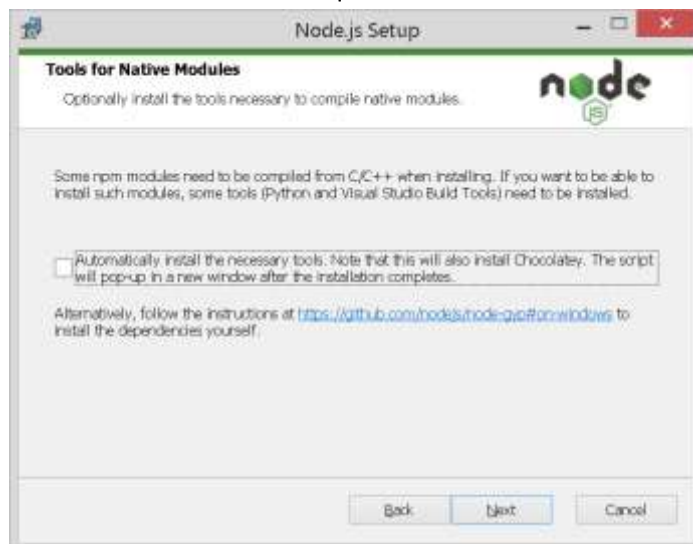
Seleccionar directorio de instalación y next



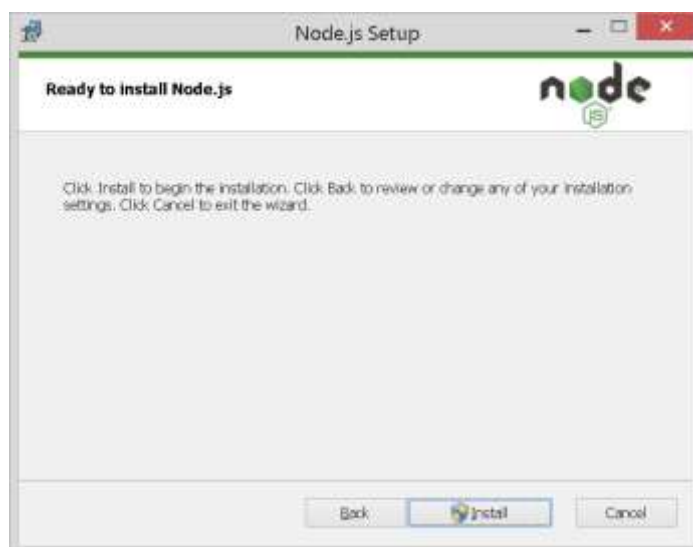
Seleccionar elementos de instalación por defecto y next



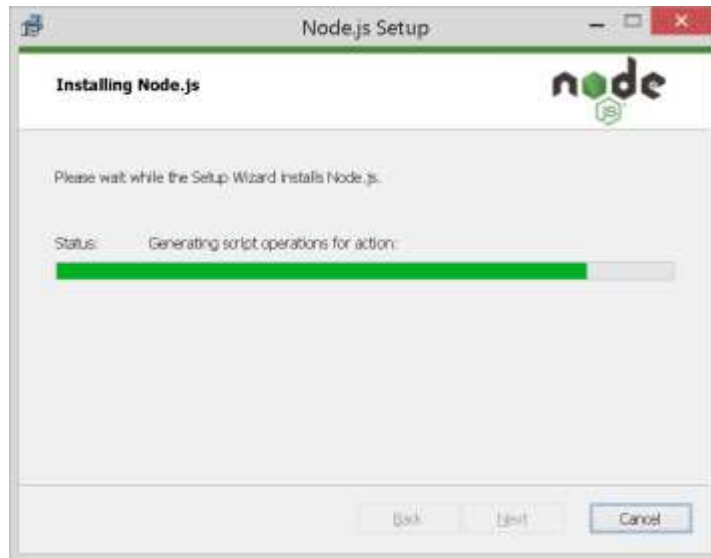
Desactiva el check no se requiere de instalación adicional y next



Finalmente click en Instalar



Continuará la instalación y finalizará



Concluido, click en Finish



3. Creación de un Proyecto en React

Veremos 3 forma de crear proyectos con React

Create React App

```
>npx create-react-app nombreProyecto  
  
cd nombreProyecto  
npm start
```

Vite

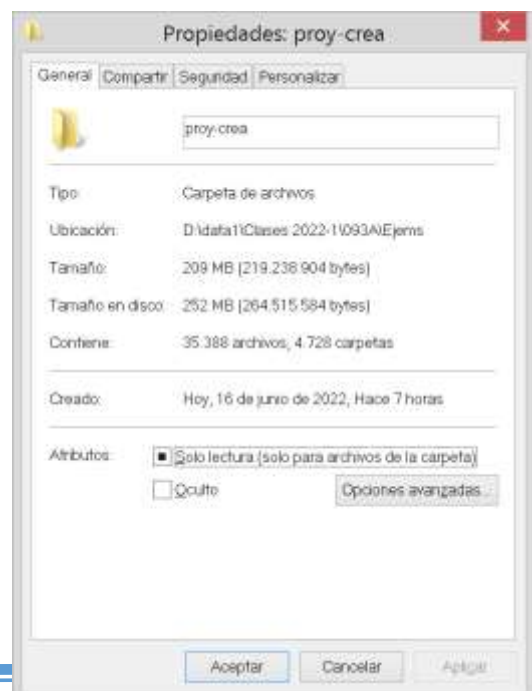
```
npm init vite@latest  
#Ingresar: nombreProyecto  
  
0  
  
>npm create vite nombreProyecto  
#Ya tiene nombreProyecto, <Enter>  
  
cd nombreProyecto  
npm install  
npm run dev
```

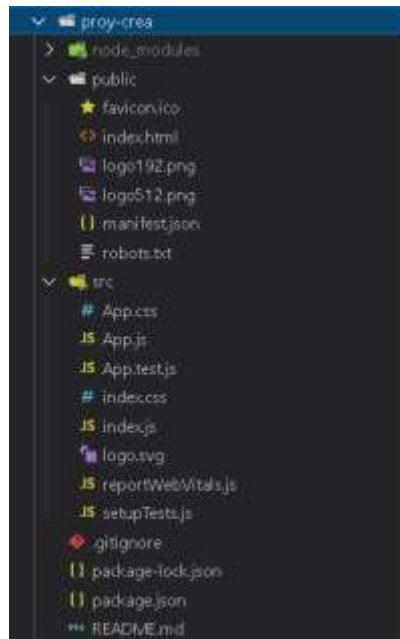
Next.js

```
>npx create-next-app  
#Ingresa: nombreProyecto  
  
cd nombreProyecto  
npm install  
npm run dev
```

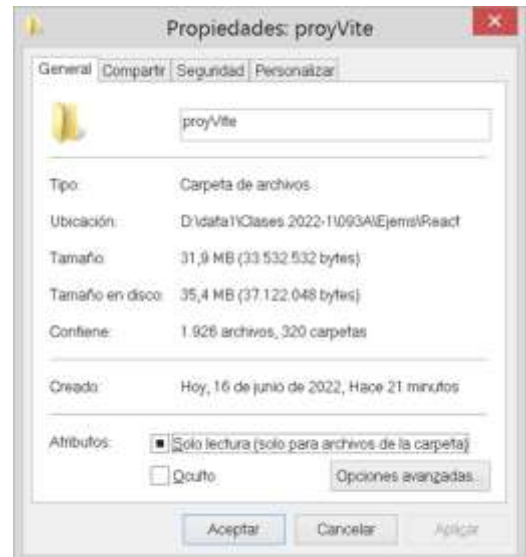
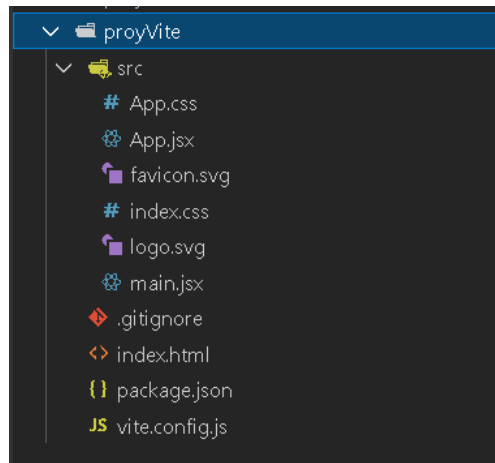
4. Estructuras de un Proyecto en React

Create-react-app

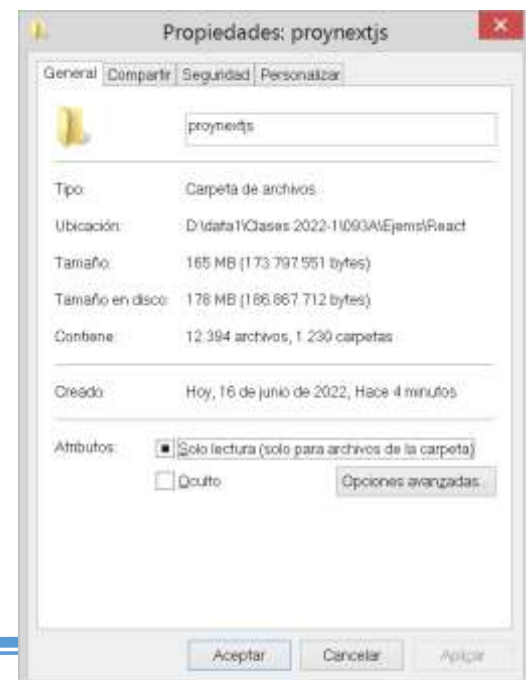


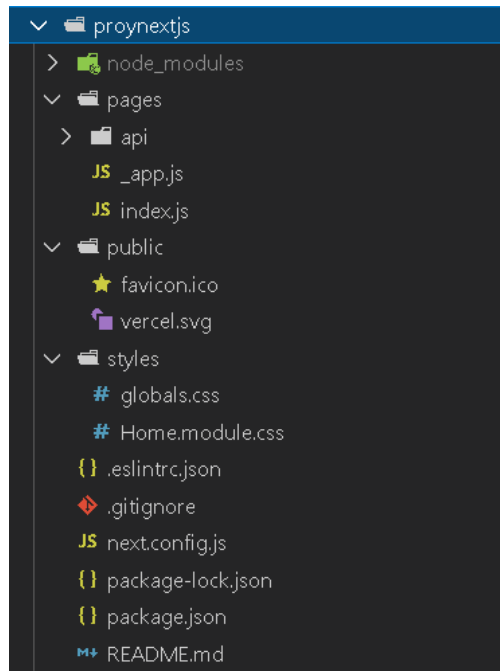


Vite



Next.js





5. Extensiones

Extensiones de Visual Studio Code

- Activitus Bar
- Bracket Pair Colorizer 2 Colores en las llaves
 - Setting: Open Setting(JSON)
 - Copiar colores personalizados
 -
- Tokyo Night Tema de Visual Studio
- Material Icon Theme paquete de iconos
- NextJS Snippets tsx and jsx
- ES7 + React/Redux/React-Native snippets
- Simple React Snippets
- **Auto Close Tag**
- Paste JSON as Code
- Typescript importer
- Css modules
- DotENV

Extensiones de Google Chrome

- React Developer Tools (Desde Chrome web store)
- JSON Viewer Pro

Alternativos

- Auto Rename Tag
- Code snippets – React/React Native/React Hooks/Styled Components
- CSS Peek
- Tailwind CSS intellisense

Evaluación de competencia

1. ¿Qué es Vite? ¿Cuál es la diferencia con create-react-app?
2. ¿Qué es el package.json? ¿Qué contiene? ¿Para qué sirve?
3. ¿Qué es el directorio node_modules? ¿Qué contiene? ¿Para qué sirve?
4. Cuando crea un proyecto en Vite, se crea varios archivos, ¿Cuál es el archivo de inicio?
5. ¿Para que sirve el archivo main.jsx?
6. ¿Para que sirve el archivo app.jsx?
7. Investigue donde se encuentra almacenado la librería react, explique que contiene sus diferentes archivos.
8. Investigue la clase React, que propiedades y métodos contiene.
9. Investigue la clase ReactDOM, que propiedades y métodos contiene.
10. Investigue acerca del servidor web interno que tiene el framework react, ¿cuales son sus limitaciones?