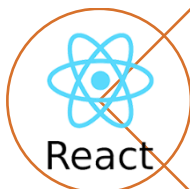


Actividad de Aprendizaje N° 05A

Introducción al Desarrollo con React

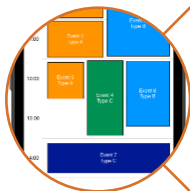
1. Qué es React: Framework Fronted



React es una *librería de código 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.

La **documentación de React JS** por ahora podemos encontrarla en:

Documentación General <https://es.reactjs.org/>

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

2. Historia de React.JS



El ingeniero de software de Facebook, **Jordan Walke**, lo creó.

React.JS se utilizó por primera vez en 2011 para la función Newsfeed de Facebook.

El lanzamiento inicial al público (V0.3.0) fue en julio de 2013.

La versión actual de React.JS es V18.0.0 (abril de 2022).

La versión actual de create-react-app es v5.0.1 (abril de 2022). reate-react-app

Incluye herramientas integradas como webpack, Babel y ESLint.

3. Desarrollo con React con CDN o Node JS



La carpeta **node_modules** es un directorio esencial en proyectos que usan **Node.js** y **npm**. Su propósito principal es **almacenar todas las dependencias y paquetes instalados** mediante **npm install**.

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 versión 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>

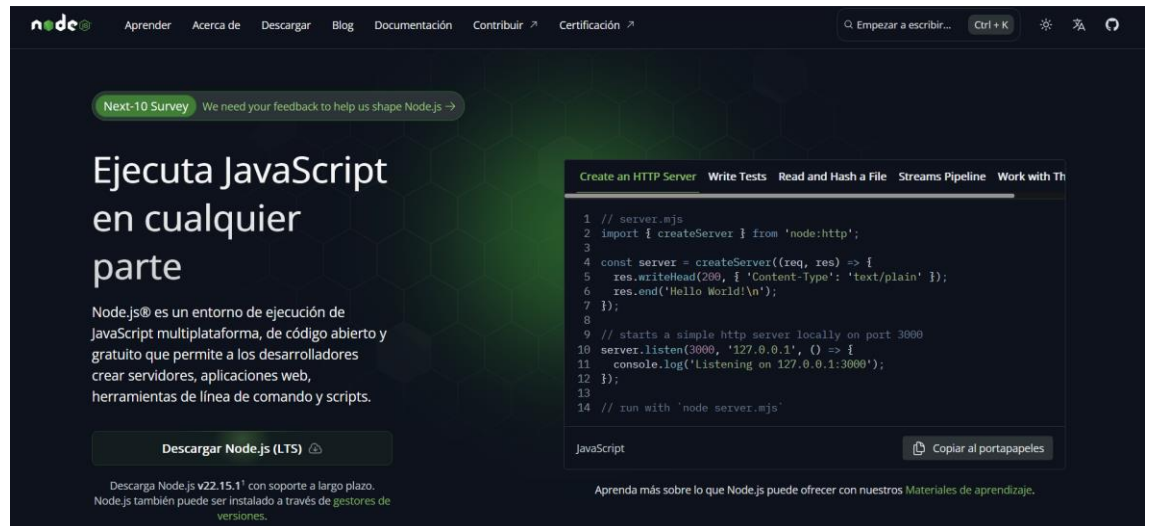
  <script type="text/babel">
    function Hola() {
      let txt = " por Jaime";
      return <h1>Hola Mundo! {txt} </h1>;
    }

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

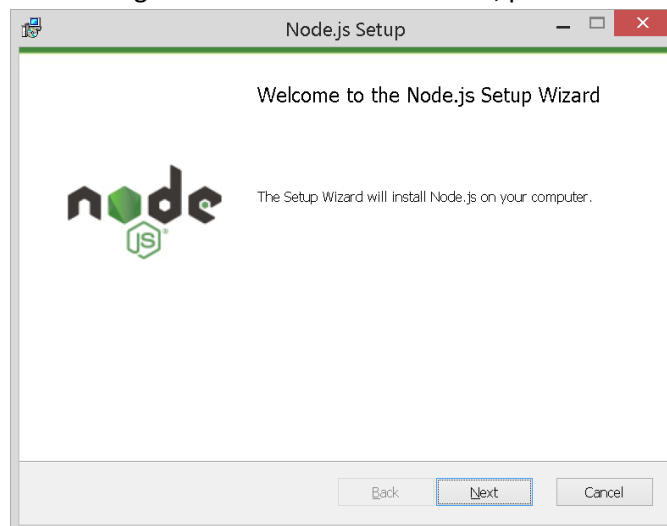
4. Probar la página.

4. Node JS – Instalación

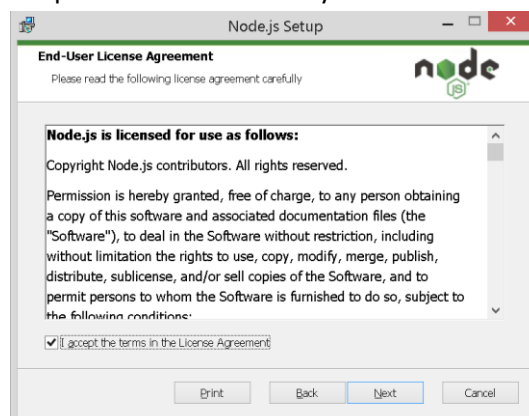
Ingresar a la web: <https://nodejs.org/es/>



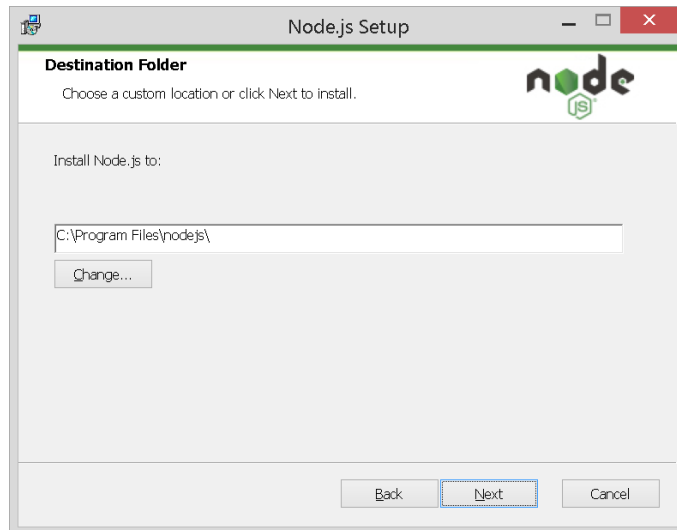
1. Descargar el archivo versión LTS
2. Ejecutar el archivo descargado.
3. Seguir el asistente de instalación, pulse en next



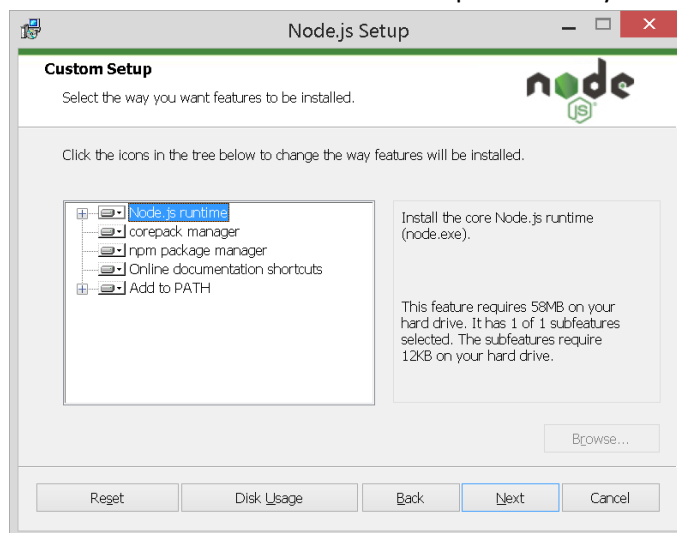
Aceptar la licencia de uso y next



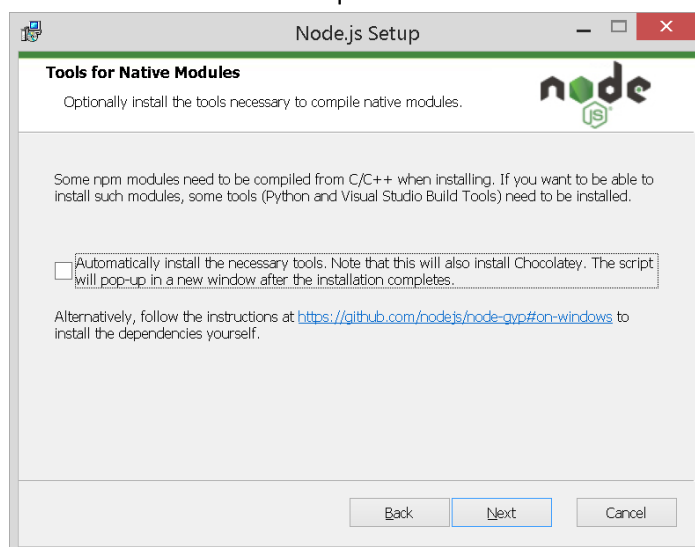
Seleccionar directorio de instalación y next



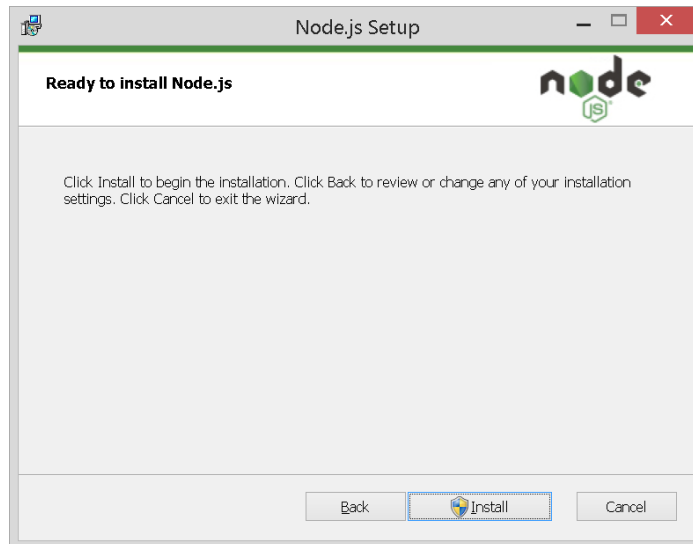
Seleccionar elementos de instalación por defecto y next



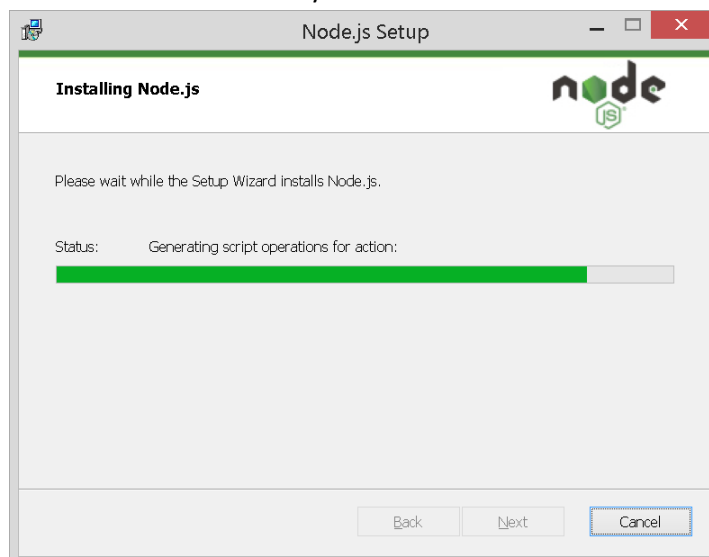
Desactiva el check no se requiere de instalación adicional y next



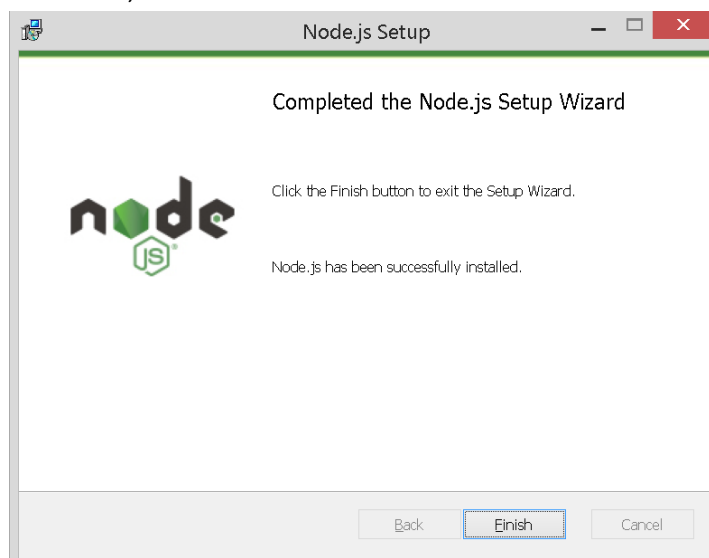
Finalmente click en Instalar



Continuará la instalación y finalizará



Concluido, click en Finish



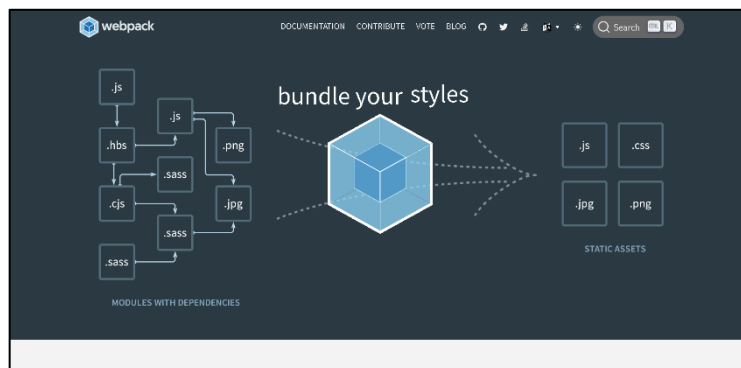
5. Herramientas de React – Entornos de Desarrollo

Para el desarrollo eficiente de proyectos en React.js y otros frameworks de JavaScript, es fundamental contar con un **entorno de desarrollo** bien configurado. Esto implica comprender y utilizar herramientas especializadas para:

- **Inicialización del proyecto:** Configuración de dependencias y estructura mediante **Vite, Create React App**
- **Compilación:** Conversión del código fuente a una versión optimizada y ejecutable, generalmente gestionada por Webpack o Turbopack.
- **Transpilación:** Transformación de código JavaScript moderno (ES6+) y JSX a versiones compatibles con navegadores mediante Babel.
- **Gestión y automatización de tareas:** Análisis de código estático como ESLint, Prettier para formateo, Extensiones y Snippets.
- **Sistema de Control de Versiones** como Git o Husky para automatización de procesos en control de versiones.

Webpack/TurboPack/RollUp

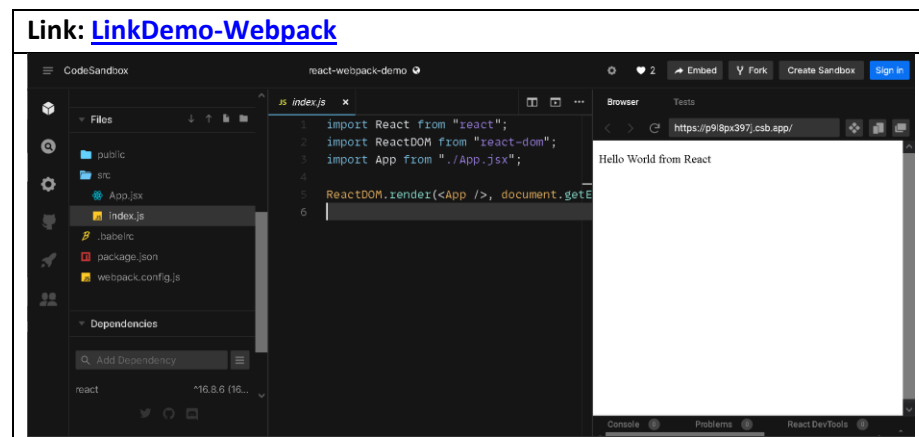
Actualmente el desarrollo web utiliza múltiples dependencias, por lo que Webpack convierte esas dependencias en archivos que pueda entender el navegador de una manera más rápida e interoperable.



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/>



Babel JS/SWC (Speedy Web Compiler)

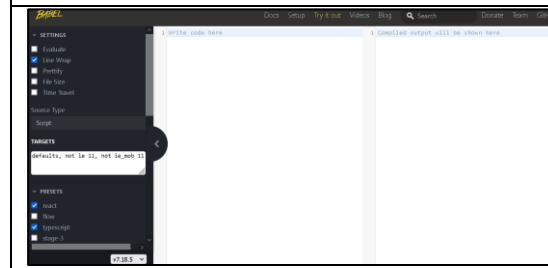


Babel es una herramienta que transforma nuestro código JS de última generación (o con funcionalidades extras) a un código de Javascript que cualquier navegador o versión de Node.js pueda entender.

Transforma *Typescript*, *JSX*, etc

<https://babeljs.io/>

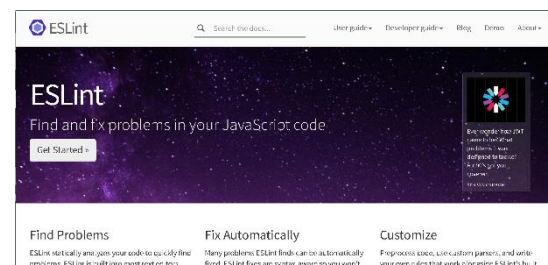
Link Demostración: <https://babeljs.io/repl>



1. Abrir Link
2. Agregar código

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

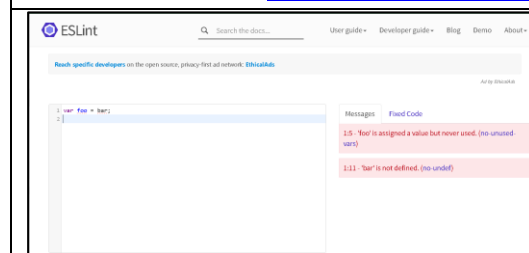
ESLint



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>



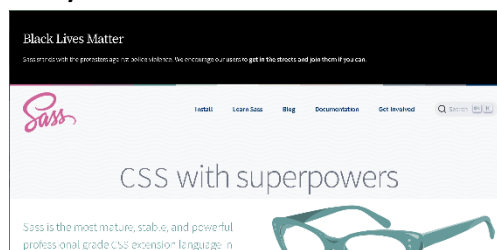
1. Abrir Link
2. Agregar código

```
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**

Otras Herramientas

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

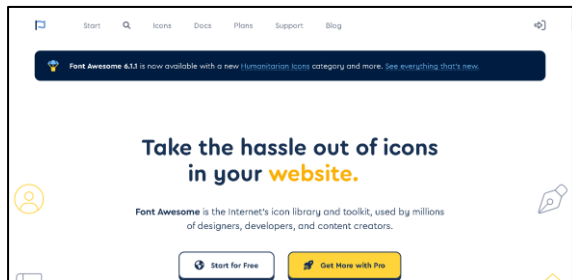
```
$primary-color: #333;  
  
.navbar {  
  background-color: $primary-color;  
}  
  
.footer {  
  border: 1px solid $primary-color;  
}
```

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/>

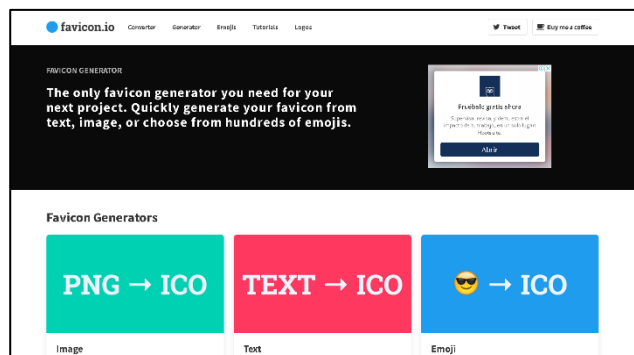


Favicon Icon



Herramienta para crear Favicons. . <https://favicon.io/>

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



Más Herramientas

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

6. Herramientas de Instalación de React

En el desarrollo de aplicaciones de *mediana escala* con React.js, es común la necesidad de *configurar herramientas* como *Babel*, *Webpack*, *ESLint* y *un servidor de renderizado* para manejar la distribución del contenido.

Este proceso puede generar la conocida *Fatiga de JavaScript*, debido a la complejidad y variabilidad en versiones y dependencias.

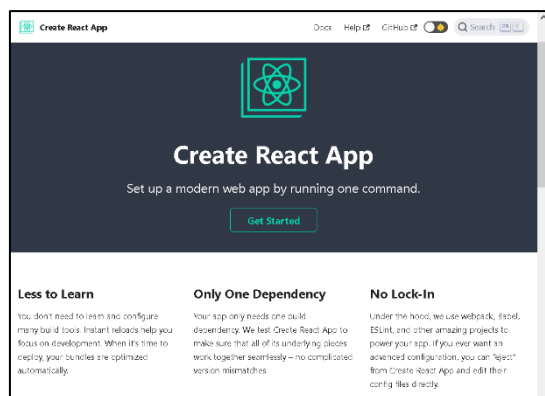
La configuración manual resulta tediosa, ya que requiere validar compatibilidad entre versiones y ajustar archivos de configuración, como *webpack.config.js*, *.babelrc* y *.eslintrc*. Para optimizar, es recomendable utilizar herramientas de instalación automatizada. Se usa herramientas con el objetivo de **NO tener que preocuparte por la configuración**.

Las principales herramientas son:

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

Este documento se revisarán sólo las 3 primeras herramientas.

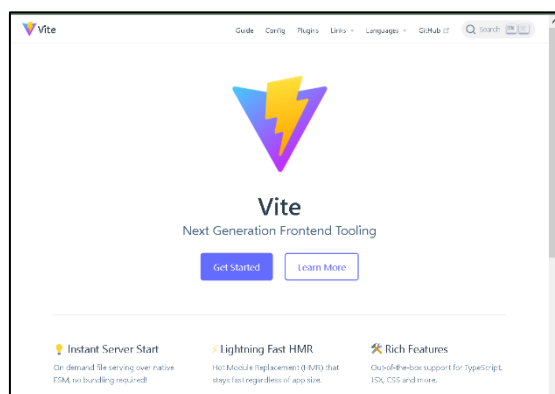
Create React App:



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 .

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

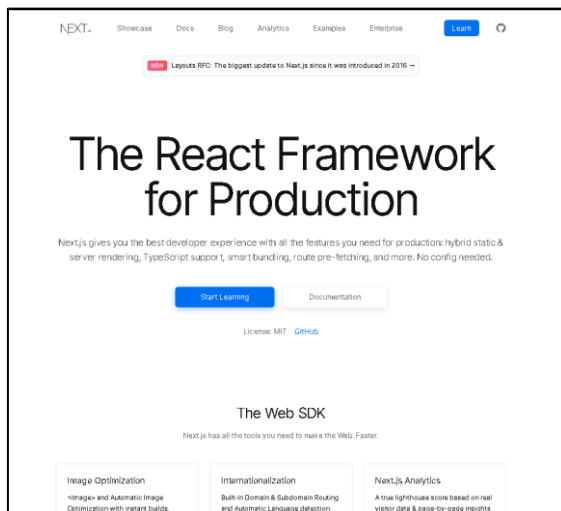
Vite:



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.

<https://vitejs.dev/>

Next.js:



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!

<https://nextjs.org/>

7. 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
```

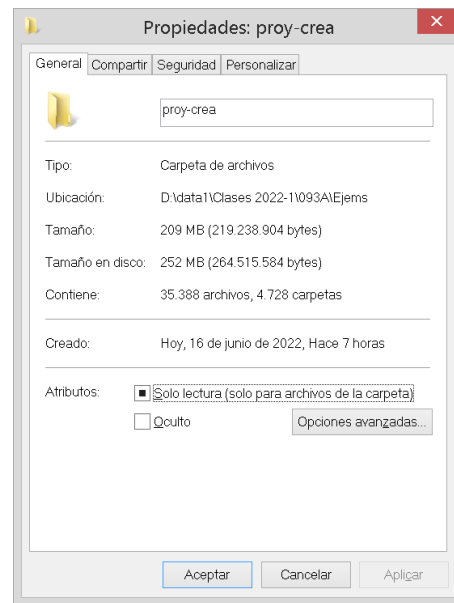
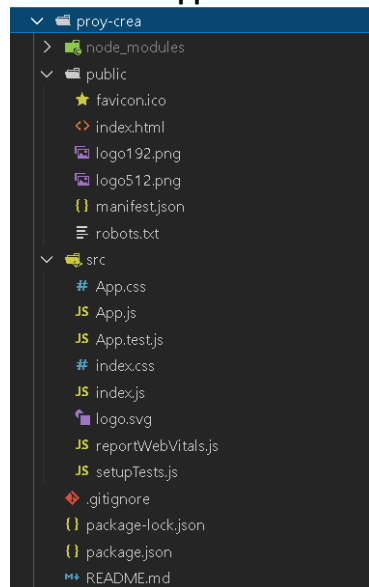
Next.js

```
>npx create-next-app
#Ingresa: nombreProyecto

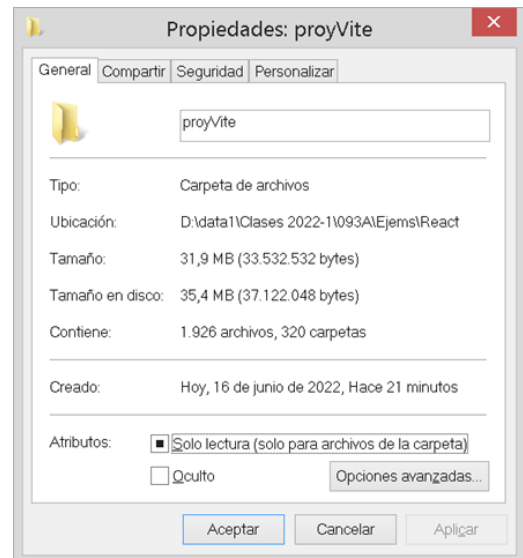
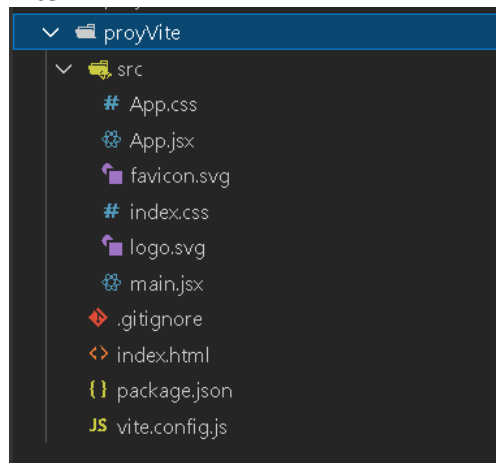
cd nombreProyecto
npm install
npm run dev
```

8. Estructuras de un Proyecto en React

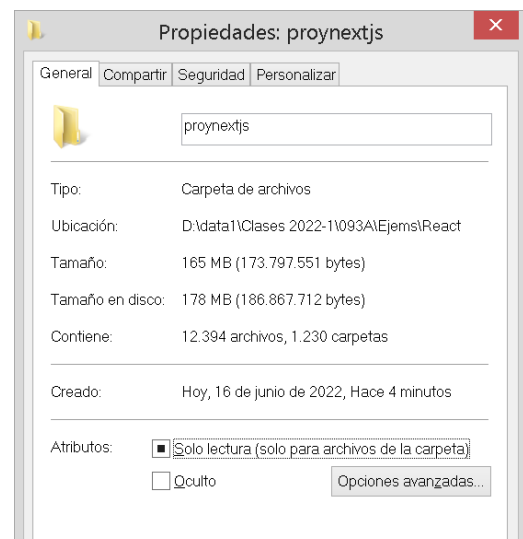
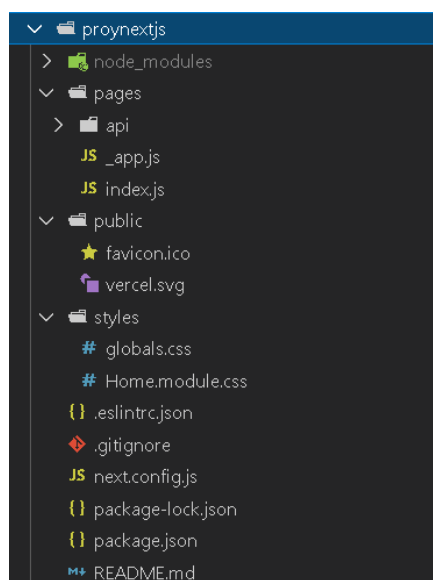
Create-react-app



Vite

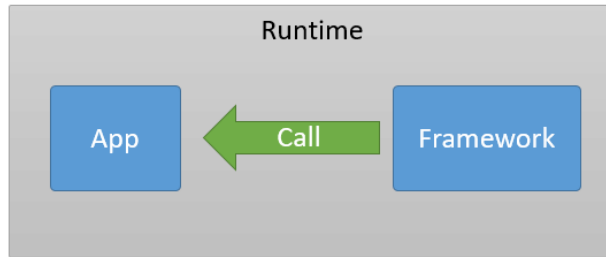


Next.js



9. ¿Cómo Funciona React en su Estructura de Archivos?

Patrón de diseño – Inversión de Control



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?