

Programación con JavaScript II

Sesión sincrónica 1

# Manejadores de paquetes y empaquetadores de módulos



# Bienvenida y actividad de bienestar

**Duración:** 10 minutos.

**Nombre de la práctica:** Encuentra tu pasión con 3 preguntas

**Descripción de la práctica:** Hoy en día se nos pide seguir nuestra pasión, pero ¿y si no sabemos cuál es? Con las siguientes preguntas podrás identificarlas.

**Palabras clave:** Preguntas, pasión, felicidad

**Instrucciones para el participante:**

## Ejercicio 1

Responde en una hoja estas preguntas:

### 1. ¿Cuál es tu pelota de tenis?

Esta pregunta la hizo Drew Houston, fundador de Dropbox en un discurso en MIT. Explicó que las personas más exitosas están obsesionadas con resolver un problema importante, algo que les es significativo, es como un perro persiguiendo una pelota de tenis, entonces si quieres incrementar tu felicidad y éxito debes encontrar tu pelota de tenis, aquello que te jala.

### 2. ¿Qué hago cuándo me siento más hermoso o hermosa?

Esta pregunta te ayuda a identificar no solo lo que te atrae sino lo que te hace brillar, lo que te hace sentir vivo o viva, o mejor, de acuerdo con lo que nos dice Jacqueline Novogratz fundadora de The Acumen Fund.

### 3. ¿Qué creencia tienes que casi nadie comparte contigo?

Esta pregunta de Peter Thiel, co-fundador de PayPal y jefe del Thiel Foundation, está diseñada para dos cosas, para ayudarte a identificar qué es importante para ti y para determinar si vale la pena perseguirlo por su originalidad.

Si encuentras un problema o reto que nadie más está abordando puedes hacer tu propio nicho y crear valor. Aunque nos enseñan a hacer lo que otros están haciendo y luego competir para ganarles, esto equivale a golpear tu cabeza contra la pared, en lugar de atravesar la puerta abierta en la que nadie se fija.

**Fuente:** <https://www.fastcompany.com/3028946/find-your-passion-with-these-8-thought-provoking-questions>

# Actividad guiada

## Parte 1

### Duración: 75 minutos.

En este tema pondrás en práctica algunas de las nuevas características de JavaScript con los siguientes ejercicios. Una vez que hayas instalado el NPM conforme a las instrucciones del *prework* y esté corriendo el sistema operativo Ubuntu sobre Windows, estás listo para comenzar con el proyecto.

### Ejercicio 1

1. Crea tu espacio de trabajo, te sugiero que utilices una ruta como **/home/user/develop/npm** en macOS y **C:\Users \Documents\npm** en Windows.
2. Dentro de la ruta sugerida, crea la carpeta de trabajo, podrías nombrarla **npm-init**.
3. Inicializa el NPM con la instrucción **npm init** para inicializar el proyecto.
4. Te solicitará un nombre de proyecto. Por lo general, toma el nombre de la carpeta, si quisieras utilizar uno diferente, ingrésalo; en caso contrario, solo presiona *Enter*.
5. Deja los valores por *default* de versión, descripción, *entry point*, *test command* y *git repository*.
6. En *keywords*, podrías poner JavaScript, Node.
7. En *autor*, coloca tu nombre y, si gustas, un correo electrónico entre estos signos <>.
8. En *license*, coloca MIT, que es la licencia que permite distribuir el código abiertamente.
9. A continuación, va a mostrar una previa de cómo se va a visualizar el archivo y pide la confirmación para la generación del archivo **package.json**.
10. Instala el paquete *moment* con NPM. Este tipo de instalaciones no tiene restricciones y puedes instalarlo con npm **install moment**.
11. ¿Puedes ver cuántos paquetes fueron instalados en tu proyecto?
12. Ahora instala un paquete que no se puede instalar en ambientes productivos, por lo que se le debe indicar que solo podrá estar en los entornos de desarrollo. Además, indica que se desea instalar la versión 2: **npm install parcel@2 -D**.
13. ¿Cuántos paquetes fueron agregados a tu proyecto?
14. Instala **react** y garantiza que esta dependencia va a ser guardada como tal, ya sea en desarrollo o en producción. Utiliza el comando npm i **react -S o npm install react -save**.
15. Observa cómo se ha modificado el archivo **package.json**. Además, observa que se ha agregado a la carpeta **node\_modules**.
16. Existe otro tipo de paquetes que son paquetes globales, que no están ligados a un proyecto, sino al entorno de trabajo.



## Actividad guiada

### Parte 1

17. Instala con NPM el paquete **cowsay** como paquete global con la instrucción `npm install -g cowsay`.
18. Observa ahora las opciones con las que puedes ejecutar cualquiera de los paquetes instalados, por ejemplo, **cowsay**.
19. Utiliza un parámetro en la llamada al paquete **cowsay**, por ejemplo, **cowsay javascript** y observa la salida.
20. Enlista los paquetes que están instalados con **npm list**.
21. Enlista los paquetes instalados de manera global.
22. Instala el paquete **eslint** con dependencias opcionales, utiliza el parámetro `-o` para hacerlo.
23. Instala el paquete **react-dom** simulando una instalación, pero sin modificar los paquetes de tu instalación para evitar conflictos. Utiliza el parámetro `--dry-run`.
24. Instala la versión 0.15.0 de **json-server**.
25. Actualiza a la última versión de **json-server** con **@latest**.
26. Si se quisieran instalar los paquetes de un proyecto clonado, ejecuta el comando **npm install**.
27. Crea un *script* de NPM, realizando lo siguiente:
  - a. Dentro de la carpeta del proyecto, crea una carpeta **src**.
  - b. Dentro de la carpeta, crea un archivo **index.js**.
    - i. Imprime en la consola un Hola Mundo.
  - c. En el archivo **package.json**, realiza las siguientes modificaciones:
    - i. Elimina el comando **test**.
    - ii. Crea el comando **start** y asígnale el valor `"node src/index.js"`.
  - d. Para ejecutar el comando, abre la terminal de Visual Studio Code y escribe **npm run start** (asegúrate de estar posicionado en el directorio raíz de tu proyecto). La salida debería ser algo parecido a esto:
  - e. Ahora ejecuta el comando `npm start` y observa el resultado.

# Actividad guiada

## Parte 1

```
root@cdctecmi/#npm run start
```

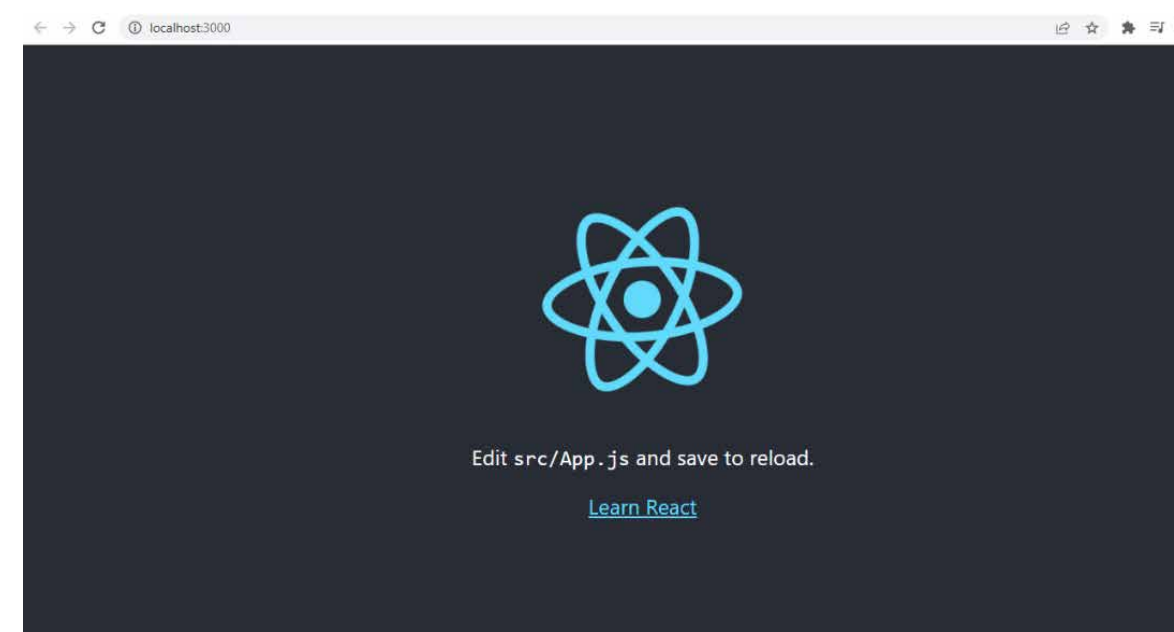
```
> npm-init@1.0.0 start
```

```
> node src/index.js
```

```
Hola Mundo
```

```
root@cdctecmi/#
```

28. Para utilizar la herramienta de ejecución de paquetes, se va a utilizar el paquete **react** de la siguiente manera:
- Desde la terminal de Visual Studio Code (VSC), sal de la carpeta actual para que quedes en la carpeta NPM.
  - En la terminal, escribe **npx create-react-app my-app**.
  - Al ejecutar este comando se instalará un paquete que tardará algunos minutos en terminar.
  - Una vez que haya terminado, ingresa a la carpeta que te indica e ingresa el comando **npm start**.
  - Observa en el navegador que el resultado de esta instalación te arroje una pantalla similar a la siguiente:



# Actividad guiada

## Parte 1

### Consideraciones:

**Aprendedor:** Posiblemente no estés familiarizado con este tipo de nomenclatura y ejecución de comandos desde la terminal, por lo tanto, se sugiere que investigues algunos paquetes y los instales, actualices y desinstales para que tomes práctica.

**Instructor:** Detecta el nivel de ejecución de comandos de terminal y la navegación en los archivos del sistema operativo desde la terminal. Si notas algunas áreas de oportunidad, plantea algunos ejercicios extra para que los aprendedores se vayan familiarizando con la forma de trabajo.

# Receso

Duración: 10 minutos.

Programación con JavaScript II



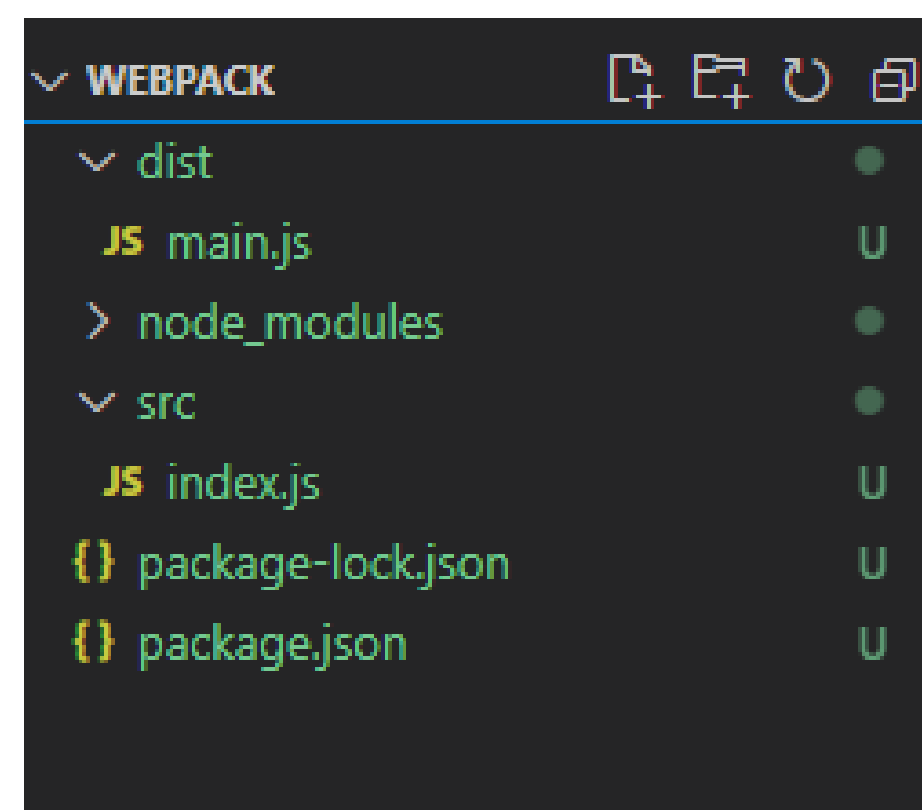
# Actividad guiada

## Parte 2

**Duración: 75 minutos.**

### Ejercicio 2

1. Prepara tu primer proyecto con webpack:
  - a. En la carpeta que hayas destinado para guardar tus trabajos de esta sesión y desde la terminal, crea la carpeta webpack.
  - b. Ingresa a la carpeta.
  - c. Crea un repositorio local con **git init**.
  - d. Inicializa NPM con la opción **-y** para que tome todos los valores por defecto.
  - e. Abre el proyecto en el editor de código, si estas utilizando el Visual Studio Code, puedes utilizar **code**.
  - f. En el directorio raíz de tu aplicación, crea la carpeta **src**.
  - g. Dentro de este directorio, crea el archivo **index.js**.
  - h. Desde la terminal (dentro o fuera de VSC), instala con NPM el paquete **Webpack** y el paquete **Webpack-cli** utilizando el parámetro **-D** para guardarlo como una dependencia de desarrollo.
  - i. Instala **webpack** de manera global con el comando **npm install -g webpack**.
  - j. Observa que estos comandos crearon la siguiente estructura:





## Actividad guiada

### Parte 2

2. Con el ambiente instalado, ahora observa qué hace webpack en el modo desarrollo. Para eso, ejecuta el comando **npx webpack --mode development**.
  - a. Observa los cambios registrados en el archivo **main.js**.
3. Ejecuta el mismo comando, pero con el modo **production**.
  - a. Observa ahora el contenido del archivo **main.js**.
4. Agrega más código al proyecto:
  - a. En la carpeta **src**, crea la carpeta **utils**.
  - b. Dentro de ella, crea un archivo llamado **sum.js**.
    - i. Crea una función flecha llamada **sum** y pásale los argumentos **a** y **b**.
    - ii. Dentro del cuerpo de la función, retorna la suma de **a** y **b**.
    - iii. Exporta la función **sum** por defecto.
  - c. En el archivo **index.js**, en la parte superior, importa la función **sum** del archivo **sum.js**.
  - d. Imprime en la consola una instancia de la función **sum** en donde le pases como parámetro 2,2.
5. Ejecuta nuevamente el **npx** con el modo **development**.
  - a. Observa que el archivo **main** incluye la nueva funcionalidad.
6. Ejecuta ahora el comando en modo **production**.
  - a. Observa como aparece una autoejecución con la nueva funcionalidad.
  - b. Observa que webpack optimiza el código para no ser redundante en cosas que no son tan necesarias.

### Ejercicio 3

Vas a construir un proyecto a partir de otro que ya se encuentra en GitHub y lo vas a preparar con webpack para mandarlo a producción.

1. Crea una copia del repositorio <https://github.com/trinuxmon/js-portfolio> en tu cuenta GitHub, realízalo haciendo un **fork** en GitHub.
2. Una vez que se encuentre el código en tu GitHub, clónalo en tu espacio de trabajo:
  - a. Dentro de la carpeta que elegiste para tus ejercicios, clona tu repositorio utilizando **git clone [url\_de\_tu\_repositorio].git**.
  - b. Ingresa al directorio de tu proyecto recientemente creado.
  - c. Navega en el directorio con VSC y observa que solo existen archivos html, css y js.

## Actividad guiada

### Parte 2

3. Optimiza el código con webpack:
  - a. Instala Webpack y Webpack-cli en el proyecto, a través de NPM, utilizando **y -D** porque esto es una dependencia de desarrollo.
  - b. Ejecuta webpack en modo producción y observa que el código se encuentra minificado.
4. Ajusta el archivo de configuración de webpack:
  - a. En la raíz del proyecto, crea el archivo de configuración llamado **webpack.config.js**.
  - b. En ese archivo, crea la constante **path** y asígnale un **require('path')**
  - c. Crea el objeto que se va a exportar con el objeto de la configuración deseada.

```
module.exports = {
  entry: './src/index.js',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'main.js',
  },
  resolve: {
    extensions: ['.js']
  }
}
```

- d. Ejecuta el siguiente comando para preparar el proyecto:  
**npx webpack --mode production --config webpack.config.js**
  - e. Una vez que termina el proceso, puedes ver en el VSC que ya está la carpeta **dist**, aunque por el momento solo se encuentra ahí el archivo **main.js**.
5. Para hacer más amigable la ejecución del comando, agrega en el archivo **package.json** un script que ayude a tal fin:
  - a. Dentro de la sección scripts, agrega una entrada llamada "build" y pasa el comando de webpack "webpack --mode production".
  - b. Y ahora solo será necesario ejecutar **npm run build**.

## Actividad guiada

### Parte 2

6. Instala el paquete para que webpack pueda hacer manejo del html ejecutando el comando:  
`npm install html-webpack-plugin -D`
7. Añade el elemento al archivo de configuración de webpack:
  - a. En la parte superior del archivo, añade la constante **HtmlWebpackPlugin** y asígnale el **require** que recibe como parámetro 'html-webpack-plugin'.
  - b. Dentro del módulo que se va a exportar, después de **resolve**, agrega los *plugins* que se estarán utilizando, en este caso, el **HtmlWebpackPlugin**. Hazlo de la siguiente manera:

```
plugins: [  
  new HtmlWebpackPlugin({  
    inject: true,  
    template: './public/index.html',  
    filename: './index.html'  
  })  
]
```

8. Modifica el archivo **index.html** y elimina el script src que se incluía en el mismo.
9. Ejecuta el **npm run build** y observa que aparece el archivo **index.html** de la carpeta **dist** y dentro del archivo **index.html** aparece una nueva línea de script **src**.
10. Agrega un nuevo script al archivo **package.json** que se llame dev y el comando a ejecutar es “webpack --mode development”.
11. Ejecuta el comando **npm run dev** y observa que el archivo **index.html** de la carpeta **dist** es más legible (no está minificado).
12. Instala ahora las dependencias para poder reconocer el **css**:
  - a. Ejecuta el comando **npm install mini-css-extract-plugin css-loader -D**.



## Actividad guiada

### Parte 2

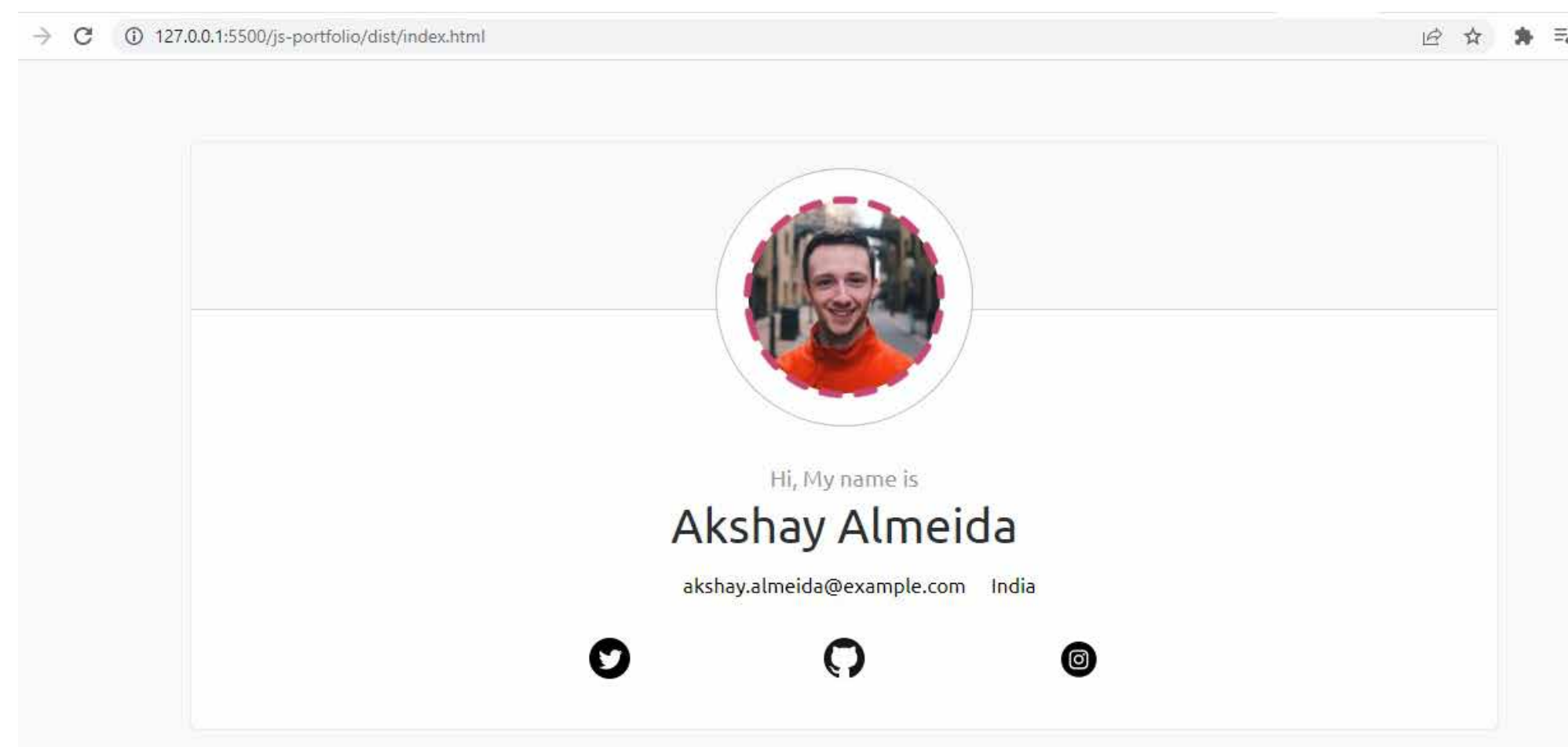
13. Prepara los archivos para que sea reconocido **css**:
  - a. Elimina del archivo **public/index.html** el link al archivo **css**.
  - b. Modifica el archivo **src/index.js** y añade los siguientes estilos:
    - i. Importa los estilos desde **./styles/main.css**.
  - c. Modifica el archivo de configuración de webpack, agrégale la constante **MiniCssExtractPlugin** y asígnale el require que recibe como parámetro el recurso recién instalado 'mini-css-extract-plugin'.
  - d. En el módulo que se exporta, crea la nueva regla después de **resolve** y antes de plugin.

```
module:{
  rules: [
    {
      test: /\.css$/i,
      use: [MiniCssExtractPlugin.loader,
        'css-loader'
      ],
    }
  ],
},
```

- e. En los plugin, inserta la llamada al nuevo método **MiniCssExtractPlugin()**.
14. Compila con **dev** y **build** y valida que aparezca un nuevo archivo **main.css**.

## Actividad guiada Parte 2

15. Ejecuta el proyecto en Live Server y deberías tener tu aplicación funcionando así:



### Consideraciones:

**Instructor:** En el contenido de la práctica se colocaron partes de código que no se explicaron a fondo, en caso de ser necesario, ayuda a los aprendedores a entender de mejor manera esos fragmentos.

**Aprendedor:** Utiliza tu cuenta de GitHub para poder realizar el proyecto base y posteriormente actualiza el repositorio con las modificaciones realizadas.

## Cierre

### **Duración: 10 minutos.**

En esta sesión guiada pudiste ver el uso de NPM y webpack, que son dos de las tecnologías más utilizadas en el mercado laboral. También se repasaron temas de versionamiento, configuración de archivos HTML y CSS. Con esta información podrías, por iniciativa propia, publicar el resultado en Netlify y tener un proyecto más para tu portafolio

Es importante que practiques la instalación de los paquetes y pongas especial atención en la importación y uso de dichos paquetes en tu código.



La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.