



**UNIVERSIDAD VALLE DEL MOMBOY; VICERRECTORADO
FACULTAD DE INGENIERÍA; PERIODO 2024B
CARRERA INGENIERÍA EN COMPUTACIÓN
FRONT END I; SECCIÓN VIV/FI
UNIDAD III; CSS & UX/UI**

e-actividad 3.2:

Proyecto: Diseño Web con HTML & CSS

Utilizando el Diseño Web Responsive

Informe Explicativo

Integrantes:

Jonathan Antonio Arellano Márquez C.I: 24.190.278

José Enmanuel Cardoza Ferrer C.I: 31.590.138

Luis Fernando Araujo Giardinella C.I. 26.482.894

Tutor: Profesor Yerson González

Contenido

Introducción.....	3
Desarrollo del Proyecto	4
Tecnologías y Dependencias Utilizadas	4
Módulos de Node.js Utilizados.....	4
Estructura del Proyecto	5
Diseño HTML y CSS	6
HTML	6
CSS	6
Lógica de Programación	7
Carrusel de Imágenes.....	7
Efecto Parallax.....	7
Configuración de Webpack.....	8
Entrada (entry).....	8
Salida (output)	8
Reglas (module.rules).....	8
Comentarios de Desarrollo.....	9
Créditos.....	9

Introducción

El presente proyecto tiene como objetivo la creación de un sitio web corporativo para la empresa ficticia "GRUPO ARGICA INC.", dedicada a ofrecer consultas sobre software y hardware. Este proyecto se ha desarrollado con fines educativos, buscando proporcionar una experiencia práctica en el desarrollo de una página web moderna y funcional utilizando tecnologías y herramientas actuales.

El proyecto abarca desde el diseño visual y la estructura HTML del sitio hasta la implementación de estilos CSS y la lógica interactiva con JavaScript. Además, se utiliza Webpack como herramienta de compilación para “compilar”, es decir, optimizar y empaquetar los recursos del proyecto. De igual manera, también se utilizaron técnicas para desarrollo de diseño web Responsive.



Imagen: Página web desarrollada "GRUPO ARGICA INC.".

Desarrollo del Proyecto

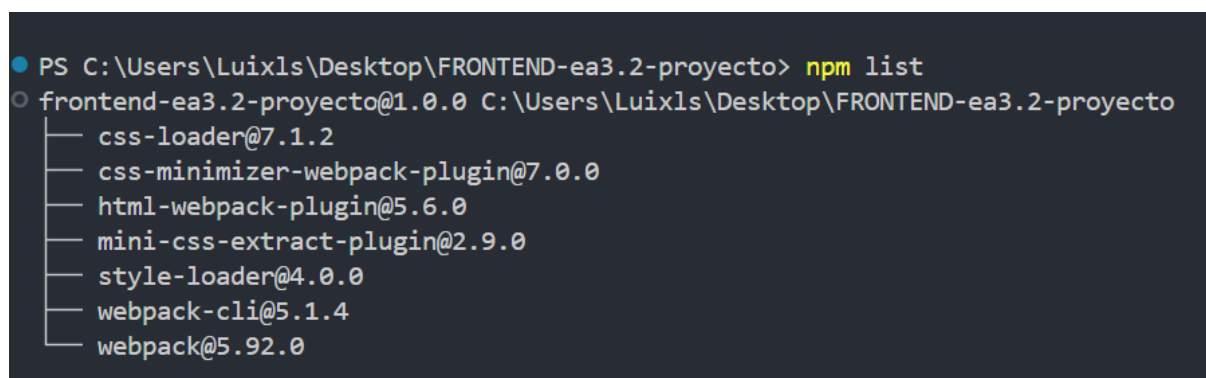
Tecnologías y Dependencias Utilizadas

Para llevar a cabo el desarrollo del sitio web se utilizaron las siguientes tecnologías y módulos:

- **HTML:** Para definir la estructura del sitio web.
- **CSS:** Para estilizar y mejorar la presentación visual del sitio.
- **Node.js (JavaScript):** Para implementar la lógica interactiva, como el carrusel de imágenes y el efecto parallax.
- **Webpack:** Para compilar y empaquetar los archivos del proyecto.

Módulos de Node.js Utilizados

- `css-loader@7.1.2` – Parte de Webpack, interpreta las importaciones.
- `css-minimizer-webpack-plugin@7.0.0` – Minificador para archivos CSS.
- `html-webpack-plugin@5.6.0` – Plugin/Dependencia de Webpack.
- `mini-css-extract-plugin@2.9.0` – Parte de Webpack, extrae código CSS.
- `style-loader@4.0.0` – Inyecta código CSS.
- `webpack-cli@5.1.4` – Webpack. Versión obviada.
- `webpack@5.92.0` – Webpack. Versión utilizada.



```
PS C:\Users\Luixls\Desktop\FRONTEND-ea3.2-proyecto> npm list
frontend-ea3.2-proyecto@1.0.0 C:\Users\Luixls\Desktop\FRONTEND-ea3.2-proyecto
├── css-loader@7.1.2
├── css-minimizer-webpack-plugin@7.0.0
├── html-webpack-plugin@5.6.0
├── mini-css-extract-plugin@2.9.0
├── style-loader@4.0.0
├── webpack-cli@5.1.4
└── webpack@5.92.0
```

Imagen: Módulos utilizados listados con el comando “npm list”.

Estructura del Proyecto

Es importante mantener el directorio del proyecto en orden para la correcta visualización de la página web del proyecto y el correcto funcionamiento de la compilación a través de Webpack. La estructura del directorio es la siguiente:

```
FRONTEND-ea3.2-proyecto/
|
├─ dist/                # Archivos compilados
├─ node_modules/        # Dependencias de Node.js
├─ src/                 # Archivos fuente (source)
│   └─ imgs/            # Imágenes del proyecto
│   └─ estilos.css      # Estilos CSS
│   └─ index.html       # Archivo HTML principal
│   └─ index.js         # Punto de entrada para Webpack, lógica
├─ package.json          # Configuración del proyecto y dependencias
├─ package-lock.json     # Configuración del proyecto y dependencias
└─ webpack.config.js     # Configuración de Webpack
```

Texto: Directorio del proyecto.

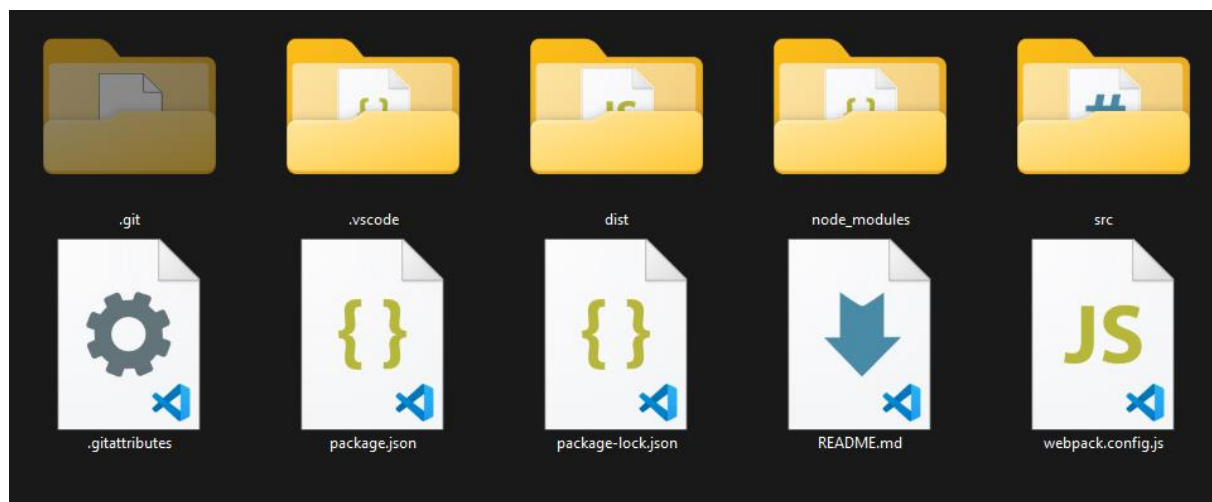


Imagen: Directorio del proyecto visto desde el explorador de archivos.

Diseño HTML y CSS

HTML

El archivo ***index.html*** proporciona la estructura básica del sitio web y contiene las siguientes secciones principales:

- Header: Incluye el nombre de la empresa y la barra de navegación con enlaces a las diferentes secciones.
- Sobre Nosotros: Presenta información general sobre la empresa.
- Carrusel: Un carrusel de imágenes que se desplazan automáticamente.
- Servicios: Describe los servicios ofrecidos por la empresa.
- Nuestro Equipo: Muestra información sobre los miembros del equipo.
- Contáctanos: Proporciona información de contacto.

CSS

El archivo ***estilos.css*** define los estilos para el sitio web. A continuación, se destacan algunos aspectos importantes del diseño:

- Fuentes y Colores: Se utilizan fuentes de Google Fonts y una paleta de colores que combina tonos azules y blancos para una apariencia profesional y moderna.
- Carrusel y Parallax: Se implementan estilos específicos para el carrusel de imágenes y el efecto parallax en las imágenes de fondo.
- Responsividad: Se utilizan media queries para asegurar que el sitio sea responsive y se vea bien en dispositivos de diferentes tamaños.
- Diseño: Se aprovechan las ventajas de las posibilidades que ofrece CSS para ajustar tamaños, especificar colores y tomar decisiones de diseño.

Lógica de Programación

El archivo `index.js` contiene la lógica necesaria para el funcionamiento de ciertos aspectos del proyecto.

Carrusel de Imágenes

- 1) Carga del DOM: La función principal se ejecuta cuando el contenido del DOM está completamente cargado.
- 2) Índice de Imagen: Inicializa una variable **"índice"** para seguir la imagen actual mostrada.
- 3) Selección de Imágenes: Obtiene todas las imágenes del carrusel mediante **`querySelectorAll`** y almacena la cantidad total de imágenes en **`totalImgs`**.
- 4) Mostrar Imagen: La función **`mostrarImg`** quita la clase `active` de todas las imágenes y la añade solo a la imagen cuyo índice coincide con el índice actual.
- 5) Siguiente Imagen: La función **`siguienteImg`** incrementa el índice y lo ajusta al rango usando el operador módulo (%). Luego llama a **`mostrarImg`** para actualizar la imagen mostrada.
- 6) Intervalo de Cambio: Usa **`setInterval`** para llamar a **`siguienteImg`** cada 5 segundos, asegurando que las imágenes cambien automáticamente.

Efecto Parallax

- 1) Escuchar Evento de Scroll: Se añade un **`listener`** al evento **`scroll`** de la ventana para detectar cuando el usuario se desplaza por la página.
- 2) Seleccionar Elementos Parallax: Se seleccionan todos los elementos con la clase **`parallax`** mediante **`querySelectorAll`**.
- 3) Actualizar Posición de Fondo: Para cada elemento **`parallax`**, se calcula la nueva posición de fondo en función de la cantidad de desplazamiento vertical de la página (**`window.pageYOffset`**). La posición de fondo se ajusta para crear el efecto de **`parallax`**, multiplicando el desplazamiento por 0.5 para un efecto más sutil.

Configuración de Webpack

El archivo de configuración de Webpack (***webpack.config.js***) se diseñó para compilar y empaquetar eficientemente los recursos del proyecto. Incluye configuraciones para manejar archivos CSS e imágenes, y plugins para generar el archivo HTML final y extraer CSS en un archivo separado.

Entrada (entry)

entry: *"./src/index.js"*: Define el punto de entrada del proyecto. Webpack empieza a construir su grafo de dependencias desde index.js.

Salida (output)

filename: *"bundle.js"*: El nombre del archivo de salida generado por Webpack.

path: *path.resolve(__dirname, "dist")*: La ruta de la carpeta de salida donde se almacenarán los archivos compilados.

Reglas (module.rules)

CSS

test: */\.css\$/i*: Aplica esta regla a todos los archivos CSS.

use: *[MiniCssExtractPlugin.loader, "css-loader"]*: Usa

MiniCssExtractPlugin.loader para extraer el CSS en un archivo separado y css-loader para interpretar las importaciones CSS.

Imágenes

test: */\.(png|jpe?g|gif|svg)\$/i*: Aplica esta regla a todos los archivos de imagen.

type: *"asset/resource"*: Trata los archivos de imagen como recursos y los copia a la carpeta de salida.

generator.filename: *"imgs/[name][ext]"*: Define el nombre del archivo y la carpeta donde se guardarán las imágenes procesadas.

Comentarios de Desarrollo

Los más grandes desafíos se trataron sobre la implementación del carrusel de imágenes y el efecto parallax al hacer scroll, esto debido a que son implementaciones que se trabajan en todos los archivos: el HTML, el CSS y el JS. De igual manera, la correcta compilación del proyecto a través de Webpack tomó mucha prueba y error para lograr exportar y minificar los archivos correctamente.

Créditos

El repositorio puede ser accedido a través de la siguiente URL a GitHub:

<https://github.com/Luixls/Frontend-ea3.2-proyecto>

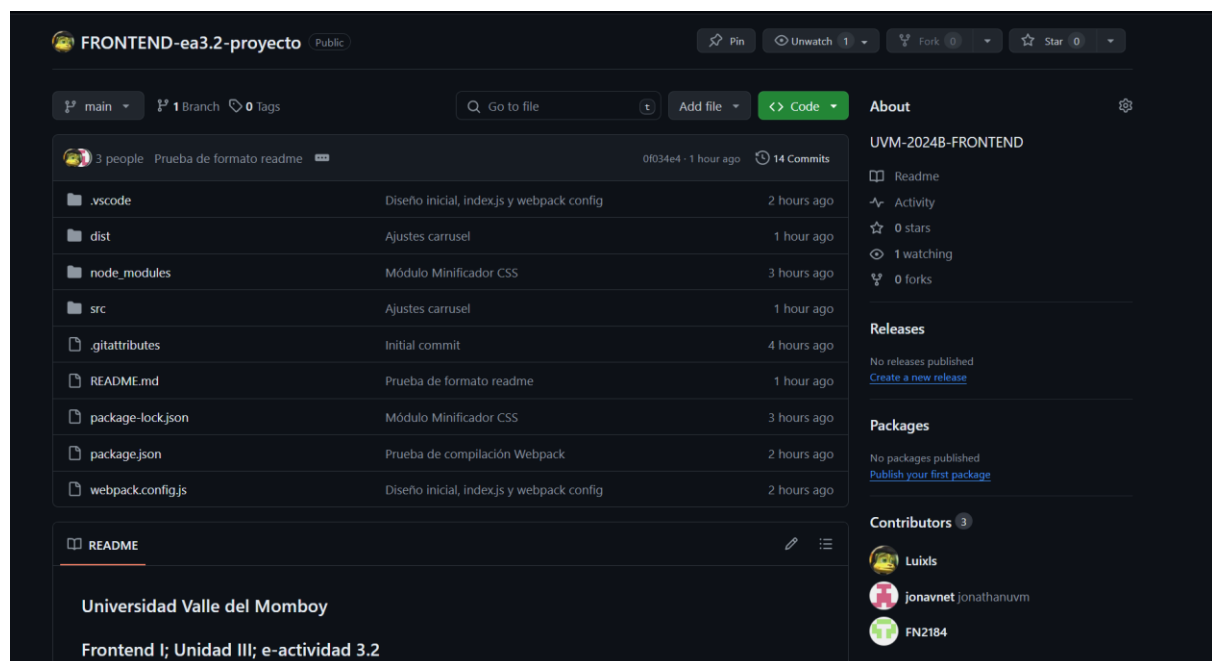


Imagen: Repositorio del proyecto en GitHub.

Por otro lado, los commits pueden ser accedidos desde el siguiente enlace:

<https://github.com/Luixls/Frontend-ea3.2-proyecto/commits/main/>

A continuación, se nombran los integrantes del equipo con su nombre de usuario correspondiente de GitHub:

@jonathanuvm - Jonathan Antonio Arellano Márquez C.I: 24.190.278

@FN2184 - José Enmanuel Cardoza Ferrer C.I: 31.590.138

@Luixls - Luis Fernando Araujo Giardinella C.I. 26.482.894