Repair Cats

Maikel Villegas

Entrega de Informe

Facultad de Ingeniería, Universidad “Valle del Momboy”

Front-End

Brian Santeliz

# Consideraciones

* Se empezó a indagar con diferentes bases de datos para ver cual era mas optima al proyecto
* Se decidió trabajar con Visual Studio Code, Github Desktop, Node JS, MySQL y XAMPP
* Las rutas permiten obtener los datos de entrada de parte del usuario, para poder almacenarlos en una variable que será usada en la función del controlador y de la fuente
* Los controladores estarían en la carpeta controllers, en los controladores se obtiene las funciones de SRC con la variable que declaramos al principio que será la que llame a las funciones, además de servir para evaluar ciertos parámetros antes de llamar a la función
* Las fuentes (src), tienen la conexión con la base de datos y son las que tienen las funciones que permitan hacer cambios dentro de la base de datos, así mismo obtendrá los datos de entrada del usuario (a partir de la ruta y controlador) para hacer los cambios o agregados a la BD
* Se hizo un Front-End funcional con la mayoría de aplicaciones de Back-End que tenemos actualmente
* El localhost es del puerto 3306
* Se trabajaron con diversas dependencias como underscore, nodemon, Morgan, http error, express, bcrypt, jwt y dotenv

La conexión con los routers de cada archivo, los middlewares, soportes de archivos y sistema de error están en la carpeta de app.js-

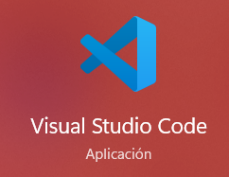
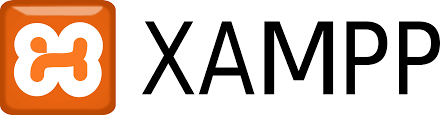
* La creación del servidor se encuentra en bin/www.js

Diseño Propuesto:

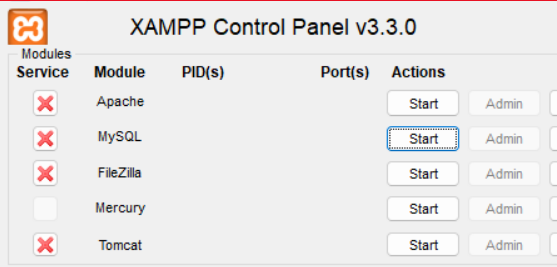
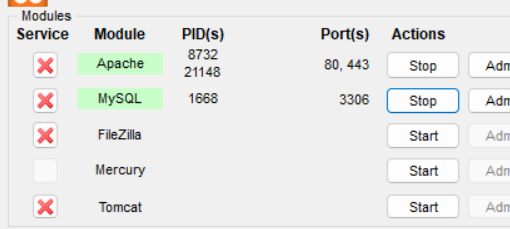
Repair Cats, será un sitio web en donde construiremos una Rest API, la cual podrá hacer inicio de sesión y registro de usuario con JWT y Bcrypt, tendrá la capacidad de conectar los productos del Back-End con el Front-End, hará conexiones de Back con Front de forma segura. A su vez, poseerá un índex, una tienda virtual, una página de presentación de la compañía y una página de Contacto. Además, será un web con la capacidad de tener un estilo responsive, así como ser empaquetado con webpack para mejorar sus procesos

# Instrucciones de Uso

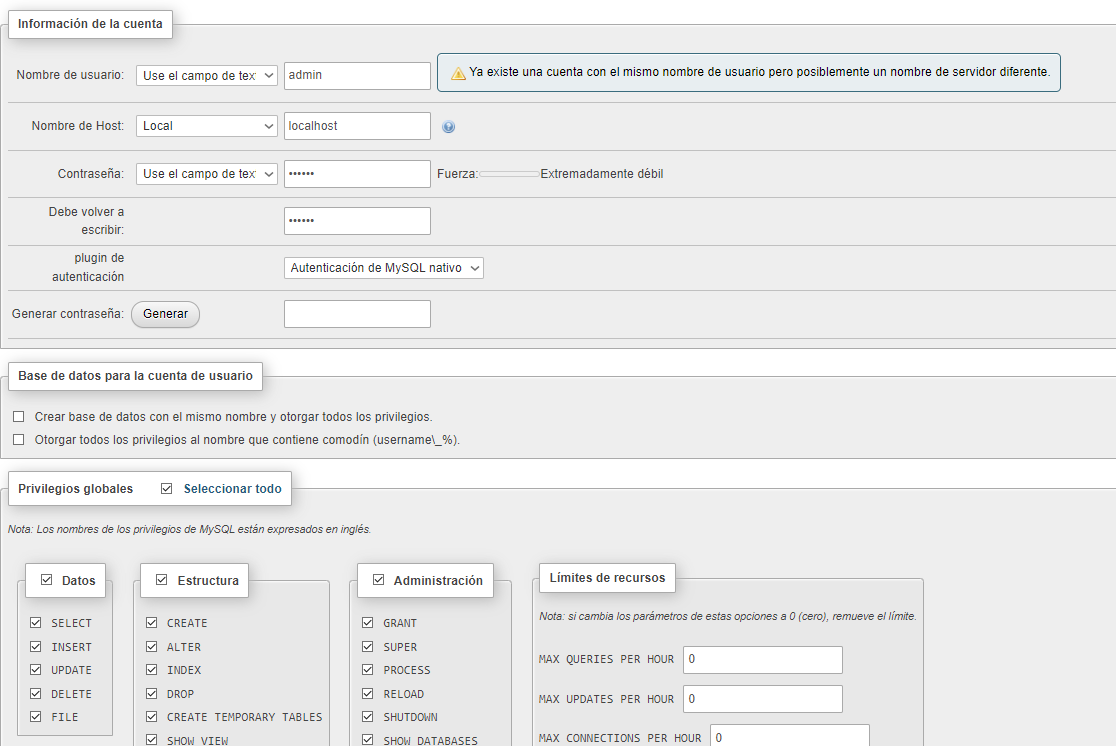
1. Instalar los siguientes programas para tener lo necesario para inicializar el servidor y usarlo



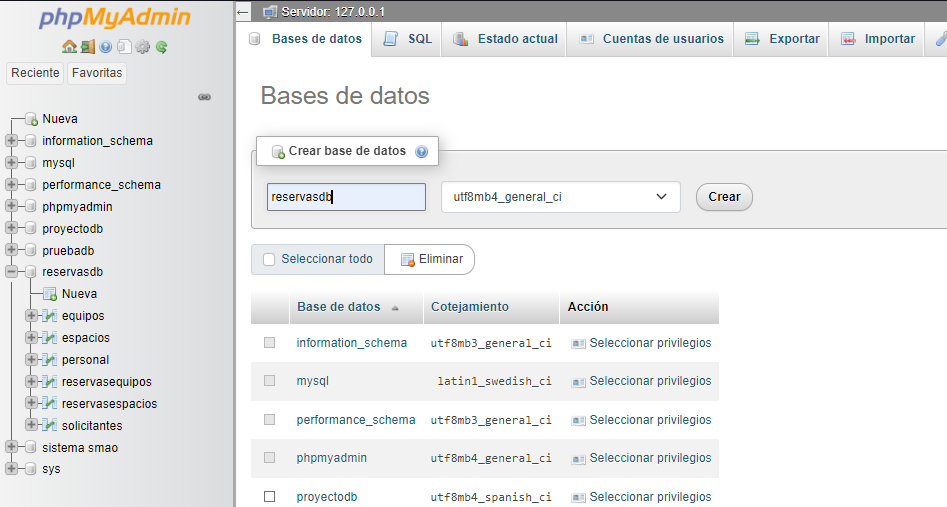
1. Ir a los siguientes links https://github.com/CrimvaelVR/BackEndRepairCats-MaikelVR-30302836, <https://github.com/CrimvaelVR/RepairCats-MaikelVillegas-30302836> y descargar ambas carpetas de nuestro programa desde el botón de Code/Local y luego a Download ZIP
2. Al haber descargado el ZIP, lo extraemos y obtenemos dos carpetas
3. Teniendo MySQL y XAMPP instalado, ir a XAMPP y abrir los puertos de Apache y MySQL con los botones de START, al haberse activado los puertos y servicios se vera en color verde y nos notificará en consola el cambio de estatus y que se encuentra activo, entonces le damos a ADMIN en el servicio de MySQL y así nos abrirá el **phpmyadmin** (para administrar nuestras BD)



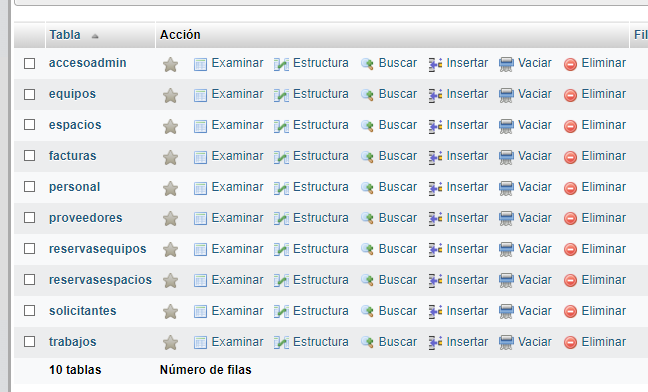
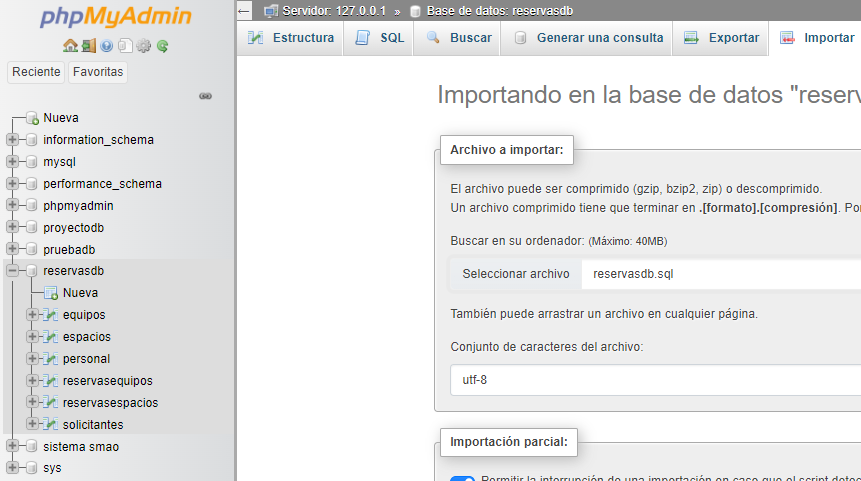
1. Vamos a necesitar crear un usuario desde PHPMyadmin, nos vamos a Cuentas de Usuario, nos saldrá agregar cuentas de usuario (en la parte inferior) y le damos a ese botón, colocaremos los siguientes datos: usuario(root); nombre de host (local, localhost); contraseña (123456); y le damos en privilegios globales a seleccionar todo, luego de esto le damos a continuar. (Ejemplo dado de un proyecto pasado)



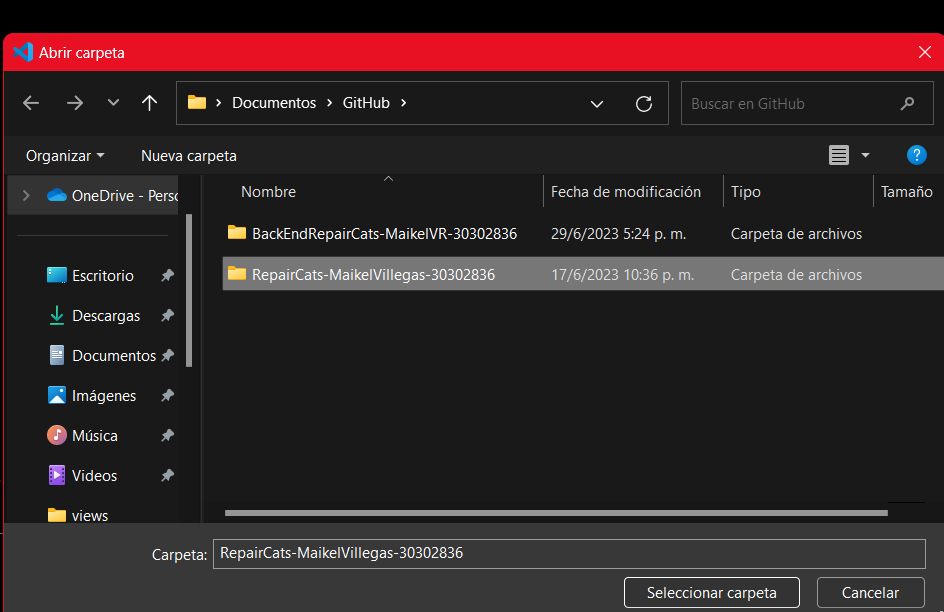
1. Lo siguiente es crear nuestra base de datos, pueden ver que en la barra vertical izquierda hay un botón de “Nueva”, le damos a ese botón y creamos una base de datos de esta forma, colocando de nombre repaircats y le damos a crear, luego de esto nos aparecerá nuestra base de datos, y podemos observarlo desde la barra vertical izquierda, además de mandarnos a la base de datos, mostrándonos que no hay tablas actualmente



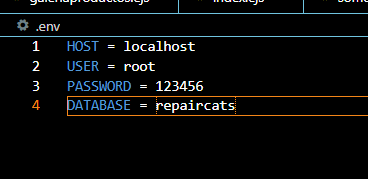
1. Luego tenemos que importar los datos que tenemos en el archivo SQL que se encuentra en la carpeta del proyecto (la que descargamos), lo que haremos es ir a nuestra base de datos, seleccionarla, y luego en la barra superior, le damos a importar. Entonces, nos saldrán diferentes opciones, pero lo único que queremos es importar de nuestro archivo, así que seleccionamos el archivo SQL de la carpeta del proyecto, y luego le damos al botón de importar (se encuentra abajo). Al hacerlo nos notificara de una gran cantidad de comandos realizados, y ahora podrán ver sus tablas y los datos de ejemplos de la Base de Datos (Ejemplo dado de un proyecto pasado)



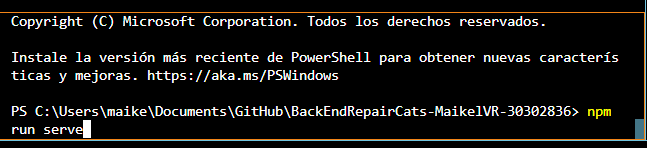
1. Procedemos a extraer los archivos RAR, mezclar los archivos que hay en ambos en una sola carpeta y luego abrir esa carpeta en Visual Studio Code



1. Creamos un “.env” dentro de nuestra carpeta principal del proyecto y colocamos los siguientes datos



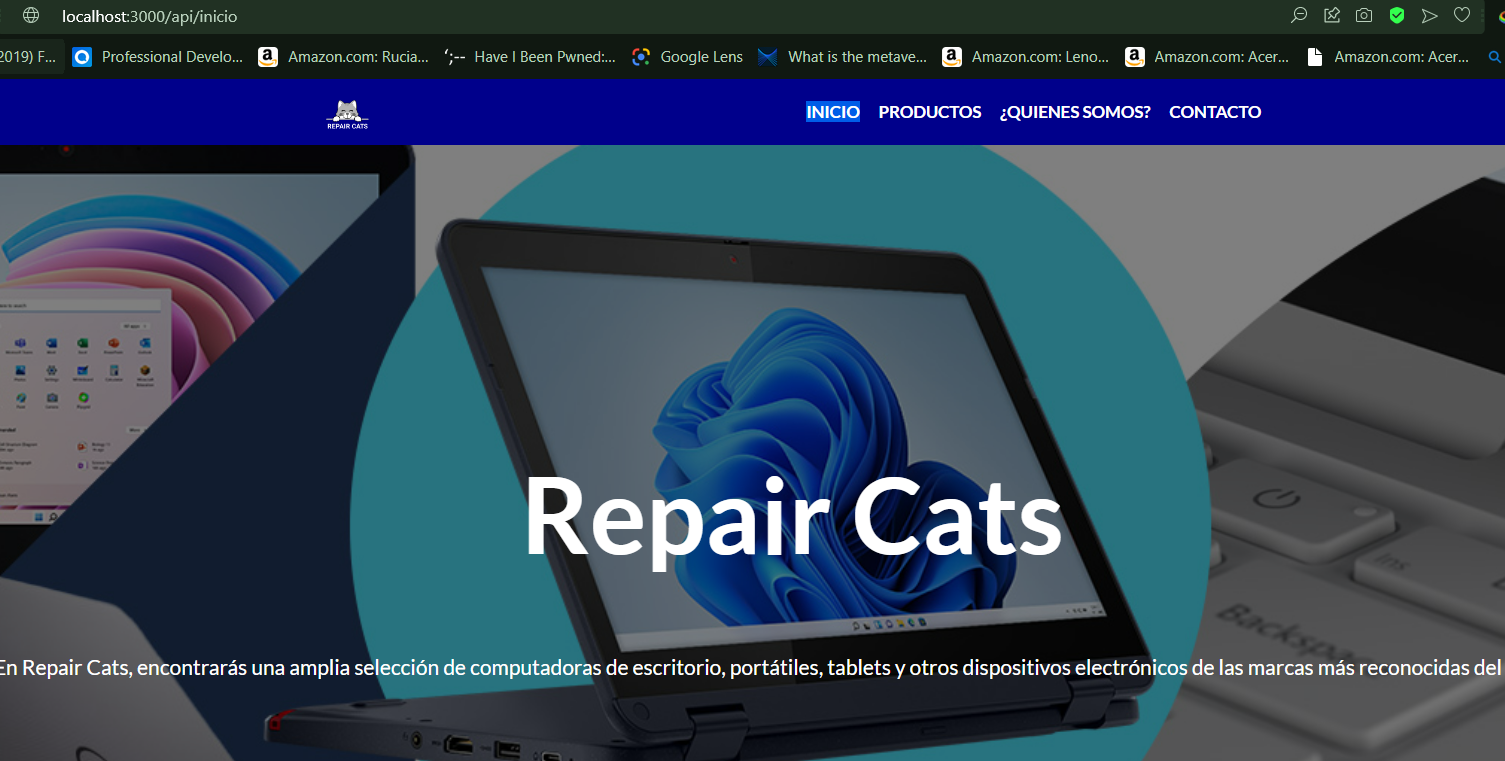
1. Iniciamos el programa con el comando “npm run serve”



1. Tenemos las siguientes rutas para nuestro sistema y las colocamos en nuestro navegador

* localhost:3000/api/inicio
* <http://localhost:3000/api/galeria>
* <http://localhost:3000/api/somos>
* <http://localhost:3000/api/contacto>

1. Así debería verse una de las paginas y con esto ya tendremos nuestro programa listo para usar



# Explicación de Codigo y Uso

A continuación, explicare cuales son algunos de los archivos mas importantes del código y como se usan en el sitio web

## App.js:

* El archivo "sqlconexion.js" contiene la función "conexion()" que se utiliza para conectarse a una base de datos SQL.
* El archivo "app.js" comienza importando la función "conexion()" del archivo "sqlconexion.js" y ejecutándola para establecer una conexión con la base de datos.
* El archivo define una instancia de Express con la variable "app".
* La aplicación utiliza el motor de vistas "ejs" y establece la carpeta "views" como la ubicación de las vistas en el servidor.
* La aplicación utiliza el middleware "morgan" para mostrar mensajes de registro en la consola de la aplicación.
* La aplicación utiliza el middleware "express.json()" para permitir al servidor entender los datos que se envían en formato JSON.
* La aplicación utiliza el middleware "express.urlencoded()" para permitir al servidor entender los datos que se envían desde formularios HTML.
* La aplicación utiliza el middleware "express.static()" para servir archivos estáticos como archivos CSS y JavaScript.
* La aplicación utiliza las rutas definidas en los archivos "formcontacto.js", "index.js", "somos.js" y "galeria.js" para manejar las solicitudes de los usuarios.
* La aplicación utiliza la función "app.listen()" para iniciar el servidor en el puerto 3000.
* La aplicación define un middleware "next(createError(404))" para manejar errores 404 (páginas no encontradas).
* La aplicación define un middleware de gestión de errores para mostrar mensajes de error en caso de que se produzca un error en el servidor.

Repair Cats cuenta con 4 páginas utilizables, en las cuales podrán ver las siguientes secciones:

## Index:

El index posee las siguientes características:

* Head: esta sección contiene información sobre la página web, como el título, la codificación de caracteres y las fuentes de estilo externas que se están utilizando.
* Header: esta sección representa la cabecera de la página web, que contiene el logotipo de la tienda, el menú de navegación y un botón para abrir o cerrar el menú de navegación. El menú de navegación es un elemento desplegable que se muestra u oculta al hacer clic en el botón del menú.
* Section Banner: esta sección es el banner de la página principal, que contiene un mensaje de bienvenida y una breve descripción de la tienda.
* Section Principal Recientes: esta sección muestra los productos más recientes de la tienda. Utiliza JavaScript para cargar los datos de los productos desde el servidor y crear dinámicamente los elementos HTML correspondientes. En particular, crea una lista de hasta tres productos, cada uno representado por un elemento de "div" que contiene una imagen, el nombre del producto, la descripción y el costo, así como un enlace a los detalles del producto.
* Sección de Productos Recientes: esta sección utiliza JavaScript para crear de manera dinámica una lista de hasta tres productos recientes de la tienda. Utiliza el objeto JSON para obtener los datos de los productos, y luego crea elementos HTML para mostrar la imagen, el nombre, la descripción y el costo de cada producto. Los elementos HTML se agregan al contenedor "Datostabla", que se define en el HTML.
* Sección de Videos de los Usuarios: esta sección utiliza la biblioteca de JavaScript Swiper para crear un slider de videos de YouTube. Cada video se muestra en un elemento de "div" que contiene el título del video y un "iframe" que muestra el video de YouTube. El slider es controlado por JavaScript, que utiliza la biblioteca Swiper para agregar la paginación y controlar la reproducción automática y la repetición del slider.
* Sección de Ubicación y Horarios: esta sección probablemente contiene información sobre la ubicación física de la tienda y sus horarios de atención al público. Sin embargo, no hay código HTML o JavaScript visible en esta sección, por lo que no se puede proporcionar una explicación detallada.
* La sección de ubicación muestra la dirección de la tienda y un mapa de Google Maps que se muestra en un "iframe" incrustado en la página.
* La sección de horarios muestra los horarios de atención al público de la tienda en una lista.
* El footer de la página contiene información de contacto y enlaces a las redes sociales de la tienda.

## Galería de Productos:

La galería de productos posee las siguientes características:

* Head: esta sección contiene información sobre la página web, como el título, la codificación de caracteres y las fuentes de estilo externas que se están utilizando.
* Header: esta sección representa la cabecera de la página web, que contiene el logotipo de la tienda, el menú de navegación y un botón para abrir o cerrar el menú de navegación. El menú de navegación es un elemento desplegable que se muestra u oculta al hacer clic en el botón del menú.
* Sección de Categorías Principales: esta sección utiliza la biblioteca de JavaScript Swiper para crear un slider de categorías principales. Cada categoría se muestra en un elemento "div" que contiene una etiqueta "p" con el nombre de la categoría y una etiqueta "span" para "Ver más". El slider se controla con JavaScript, que utiliza la biblioteca Swiper para agregar la paginación y controlar la reproducción automática y la repetición del slider.
* Sección de Productos en Venta: esta sección muestra una lista de productos en venta. Cada producto se muestra en un elemento "div" que contiene una imagen del producto, su nombre, precio y un botón para "Ver más detalles". Los elementos HTML se agregan al contenedor "div" con la clase "tarjetas".
* La sección de Categorías Principales y la sección de Productos en Venta están contenidas dentro de un elemento "section" con la clase "container".
* El código HTML también incluye un elemento "script" que importa la biblioteca Swiper y el código JavaScript para inicializar el slider.
* La sección "tabla-gpu" utiliza HTML para crear una tabla con cuatro columnas: Nombre, Descripción, Costo e Imagen.
* El código HTML incluye un elemento "p" con la clase "invisible" que contiene una variable JSON. Esta variable es utilizada por el código JavaScript para agregar datos a la tabla.
* El código JavaScript utiliza el método "JSON.parse()" para convertir la variable JSON en un objeto JavaScript y luego utiliza un bucle "for" para iterar a través de cada objeto en el objeto JavaScript.
* Dentro del bucle "for", el código JavaScript agrega una nueva fila a la tabla utilizando "innerHTML" y agrega los valores de cada objeto como celdas de la fila.
* El código JavaScript también agrega una etiqueta "img" a la celda de la imagen para mostrar la imagen correspondiente al producto.
* El footer de la página contiene información de contacto y enlaces a las redes sociales de la tienda.

## Somos:

La sección de “¿Quiénes Somos?” posee las siguientes características:

* Head: esta sección contiene información sobre la página web, como el título, la codificación de caracteres y las fuentes de estilo externas que se están utilizando.
* Header: esta sección representa la cabecera de la página web, que contiene el logotipo de la tienda, el menú de navegación y un botón para abrir o cerrar el menú de navegación. El menú de navegación es un elemento desplegable que se muestra u oculta al hacer clic en el botón del menú
* La sección "banner" utiliza HTML para crear un banner con un título y un párrafo que describe la tienda en línea y su objetivo de proporcionar productos de alta calidad a precios competitivos.
* La sección "historia" utiliza HTML para crear una sección de historia con una imagen y un párrafo que describe la creación de la tienda en línea por un grupo de amigos apasionados por la tecnología.
* La sección "slider" utiliza HTML y CSS para crear un slider de imágenes de los fundadores de la empresa. Cada imagen se muestra en una diapositiva con el nombre y el papel del fundador en la empresa. En sí, este código HTML incluye elementos "input" con el atributo "type" definido como "radio" para controlar el slider y el código CSS utiliza "position: absolute" para superponer las diapositivas y los controles.
* La sección "services" utiliza HTML y Font Awesome para crear tres servicios que ofrece la tienda en línea: Envío Gratuito, Devoluciones sin Problemas y Atención al Cliente 24/7. Cada servicio se muestra en un elemento "div" que contiene un icono de Font Awesome, un título y una descripción.
* El código HTML utiliza la clase "fa" para agregar iconos de Font Awesome a cada servicio.
* La sección "cta" utiliza HTML para crear un llamado a la acción que invita a los usuarios a contactar a la tienda en línea para obtener más información sobre sus servicios. La sección incluye un título, un párrafo y un botón con un enlace a la página de contacto de la tienda en línea. El código HTML utiliza la clase "btn-cta" para dar estilo al botón de llamado a la acción.
* El footer de la página contiene información de contacto y enlaces a las redes sociales de la tienda.

## Contacto:

La sección de Contacto posee las siguientes características:

* Head: esta sección contiene información sobre la página web, como el título, la codificación de caracteres y las fuentes de estilo externas que se están utilizando.
* Header: esta sección representa la cabecera de la página web, que contiene el logotipo de la tienda, el menú de navegación y un botón para abrir o cerrar el menú de navegación. El menú de navegación es un elemento desplegable que se muestra u oculta al hacer clic en el botón del menú
* La sección "banner" utiliza HTML para crear un banner con un título que indica que esta sección es para contactar a la empresa.
* La sección "container-form" utiliza HTML para crear dos secciones, la primera es un formulario de contacto y la segunda es una sección que muestra los datos de la empresa.
* El formulario de contacto utiliza etiquetas "label" para describir cada campo del formulario y utiliza etiquetas "input" y "textarea" para que el usuario pueda ingresar información. El formulario también incluye un botón de envío para enviar el mensaje.
* El código HTML utiliza la clase "fa" para agregar iconos de Font Awesome a los datos de la empresa.
* El código JavaScript utiliza la biblioteca jQuery para agregar una función que hace que un mensaje emergente se muestre por unos segundos cuando el usuario envía con éxito un mensaje a la empresa.
* El código HTML incluye un elemento "p" con el id "Emergente" que se utiliza para mostrar el mensaje emergente cuando el usuario envía un mensaje.
* El footer de la página contiene información de contacto y enlaces a las redes sociales de la tienda.