Documentación del Código HTML

**Index:**

Este documento detalla la estructura y el contenido del archivo HTML proporcionado, el cual corresponde a una página web de comercio electrónico.

Desarrollo

El archivo HTML sigue una estructura estándar de una página web, con las siguientes secciones principales:

Encabezado (<head>): Contiene metadatos y enlaces a recursos externos, como hojas de estilos y bibliotecas de iconos.

Cuerpo (<body>): Contiene el contenido visible de la página, organizado en varias secciones.

Barra de Navegación (<nav>): Incluye enlaces a diferentes secciones de la página y un formulario de búsqueda.

Contenido Principal (<main>): Contiene la sección principal de la página, donde se muestran los productos disponibles para la venta.

Productos (<div class="container-products">): Lista de productos disponibles para la venta, cada uno representado por una tarjeta de producto (<div class="card-product">).

Pie de Página (<footer>): Incluye información de contacto, enlaces útiles, información de la cuenta del usuario y un formulario de suscripción al boletín.

Esta estructura proporciona una experiencia de usuario coherente y facilita la navegación por la página web.

Esta documentación ofrece una visión general del archivo HTML, lo que permite comprender su estructura y contenido de manera eficiente.

**clima:**

Cuerpo (<body>):

Contenedor de la barra de navegación (<div class="container-navbar">):

Se define una barra de navegación que contiene enlaces a diferentes secciones de la página, como "Inicio", "Registrarse", "Inicio Sesión", entre otros.

Se incluye un formulario de búsqueda.

Contenido Principal (<main class="main-content">):

Sección que contiene la información del clima de Santiago para los próximos 5 días.

Se utiliza un contenedor (<div class="container">) para organizar el contenido.

Se muestra el título "Tiempo Santiago" (<h1 class="heading-1">Tiempo Santiago</h1>).

Se muestran los próximos 5 días como subtítulo (<h3 class="heading-1">proximo 5 dias </h3>).

Se utiliza un contenedor (<div id="forecastContainer" class="container mt-4">) para mostrar las tarjetas de pronóstico del clima.

Pie de Página (<footer class="footer">):

Sección que contiene información de contacto, enlaces útiles y un formulario de suscripción al boletín.

Se utilizan varios contenedores para organizar la información, como "Información de Contacto", "Información", "Mi cuenta" y "Boletín informativo".

Script (<script>):

Se incluyen varios scripts para el funcionamiento de la página, como jQuery, Popper.js y Bootstrap.

Se agrega un script personalizado para obtener el pronóstico del clima de Santiago utilizando la API de OpenWeatherMap.

Íconos (<i>):

Se utilizan íconos de FontAwesome para representar elementos como el logo, el carrito de compras, el menú desplegable, entre otros.

**Decoración :**

Cuerpo (<body>):

Contenedor de la barra de navegación (<div class="container-navbar">):

Se define una barra de navegación que contiene enlaces a diferentes secciones de la página, como "Inicio", "Registrarse", "Inicio Sesión", entre otros.

Se incluye un formulario de búsqueda.

Contenido Principal (<main class="main-content">):

Contenedor de Productos (<div class="container-products">):

Se muestran varios productos en tarjetas, cada una con una imagen, título, precio, descuento (si corresponde) y opciones como ver, agregar a favoritos y comparar.

Los productos están dispuestos en un diseño de rejilla utilizando tarjetas (<div class="card-product">).

Pie de Página (<footer class="footer">):

Sección que contiene información de contacto, enlaces útiles y un formulario de suscripción al boletín.

Scripts (<script>):

Se incluyen scripts de FontAwesome (<script src="https://kit.fontawesome.com/81581fb069.js" crossorigin="anonymous"></script>).

**Estilos (css) :**

Estilos Globales:

Variables para colores y fuentes.

Reset de estilos básicos.

Componentes Principales

Header: Logo, menú y usuario.

Navbar: Barra de navegación y búsqueda.

Banner: Imagen con texto superpuesto.

Secciones Específicas

Características: Tarjetas de características.

Categorías: Cuadrícula de categorías.

Productos Destacados: Tarjetas de producto.

Galería: Cuadrícula de imágenes.

Blogs: Tarjetas de blog.

Footer:

Menús y enlaces sociales.

Media Queries

Adaptación de estilos para diferentes tamaños de pantalla.

**For-2 (css) :**

body:

font-family: Define la familia de fuentes utilizada en todo el cuerpo del documento como Arial o una fuente sans-serif alternativa.

margin: Establece márgenes en todos los lados del cuerpo del documento para separarlo del borde del navegador.

padding: Agrega un relleno interno de 20px en todos los lados del cuerpo del documento para separar el contenido de los bordes.

background-color: Define el color de fondo del cuerpo del documento como un tono neutro #f2f2f2.

Componentes Principales:

h1:

text-align: Alinea el texto del encabezado h1 al centro.

color: Define el color del texto del encabezado h1 como un tono oscuro #333.

form:

max-width: Limita el ancho máximo del formulario a 600px para que no se extienda demasiado en pantallas grandes.

margin: Establece un margen de 0 en la parte superior e inferior y automáticamente calcula los márgenes laterales para centrar el formulario horizontalmente en la página.

background-color: Define el color de fondo del formulario como un suave tono verde #eaf7ed.

padding: Agrega un relleno interno de 20px en todos los lados del formulario para separar el contenido de los bordes.

border-radius: Aplica un borde redondeado de 10px a las esquinas del formulario para suavizar su apariencia.

box-shadow: Agrega una sombra ligera alrededor del formulario para darle profundidad y resaltar su contenido.

**Forescat (js) :**

Función: getWeatherForecast

Descripción: Esta función realiza una solicitud a la API de OpenWeatherMap para obtener el pronóstico del tiempo para Santiago, Chile. Luego, muestra el pronóstico en tarjetas de pronóstico en la interfaz de usuario.

Parámetros: Ninguno

Valor de retorno: Ninguno

Evento: DOMContentLoaded

Descripción: Este evento se dispara cuando el HTML y CSS del documento se han cargado y analizado completamente, sin esperar a que se carguen imágenes y hojas de estilo adicionales.

Función asociada: getWeatherForecast

Evento: click (Botón de la API)

Descripción: Este evento se dispara cuando se hace clic en el botón de la API. Llama a la función getWeatherForecast para obtener y mostrar el pronóstico del tiempo actualizado.

Función asociada: getWeatherForecast

Variables:

- apiKey: Clave de la API de OpenWeatherMap utilizada para autenticar las solicitudes.

- city: Nombre de la ciudad para la cual se solicita el pronóstico del tiempo.

- apiUrl: URL de la API de OpenWeatherMap para obtener el pronóstico del tiempo.

Fetch API:

- Realiza una solicitud HTTP GET para obtener el pronóstico del tiempo desde la API de OpenWeatherMap.

- Convierte la respuesta a formato JSON.

- Maneja errores si la solicitud falla.

Procesamiento de Datos:

- Filtra la data para obtener un pronóstico por día (cada 24 horas).

- Itera sobre los datos diarios y crea tarjetas de pronóstico para cada día.

Creación de Tarjetas de Pronóstico:

- Obtiene la fecha, temperatura, descripción y icono del pronóstico diario.

- Crea una tarjeta para mostrar el pronóstico diario con la fecha, temperatura, descripción e icono.

**Formulario-2:**

HTML:

- Estructura básica del documento HTML con etiquetas doctype, html, head y body.

- Metaetiqueta para especificar el conjunto de caracteres y la escala de la vista inicial.

- Título de la página.

- Enlace al archivo de estilos externo (styles.css).

- Encabezado principal (h1) con el título "Formulario de Postulación".

- Formulario con un identificador único "postulacionForm" y varios campos de entrada para la postulación.

- Campo de entrada para RUT con etiqueta "rut", nombre "rut", requisito de longitud y formato, y marcador de posición.

- Campos de entrada para apellido paterno, apellido materno, nombre, fecha de nacimiento, edad, género, email, celular, profesión y motivación, cada uno con etiquetas, nombres, requisitos y marcadores de posición.

- Dos botones dentro del formulario, uno para enviar el formulario y otro para crear una carta de presentación.

- Un área de texto para mostrar la carta de presentación generada, con un identificador único "carta" y atributo de solo lectura.

- Inclusión de las bibliotecas jQuery y el archivo de script carta.js.

- Secuencia de comandos en línea para definir funciones y eventos asociados al formulario.

JavaScript:

- Inicia una función anónima al cargar el documento HTML, utilizando jQuery.

- Dentro de la función, se definen tres funciones adicionales:

- calcularEdad: Calcula la edad en base a la fecha de nacimiento ingresada.

- validarRut: Valida el formato y dígito verificador del RUT ingresado.

- generarCarta: Genera una carta de presentación con los datos del formulario.

- Se asignan eventos a los campos del formulario y a los botones:

- Cambio en la fecha de nacimiento para calcular automáticamente la edad.

- Enfoque en el campo de celular para agregar automáticamente el prefijo +569.

- Clic en el botón "Crear Carta" para generar la carta de presentación.

Styles.css

**HTML:**

**<**body>: Define el cuerpo del documento HTML.

<h1>: Encabezado de nivel 1, centrado en el medio de la página.

<form>: Define un formulario, con un ancho máximo de 600px y centrado en la página.

<div>: Contenedor genérico para agrupar elementos del formulario.

<label>: Etiqueta para describir campos de entrada.

<input>: Entradas de texto, número, correo electrónico, y teléfono, así como áreas de texto. Estos elementos ocupan el 100% del ancho disponible y tienen un relleno de 8px.

<select>: Selector de opciones.

* **<textarea>**: Área de texto para entradas más largas.
* **<input type="submit">** y **<input type="button">**: Botones de envío y botones genéricos respectivamente, con un estilo que incluye un fondo verde, texto blanco, un relleno interno de 10px en la parte superior e inferior y 20px en los lados, sin borde y con un cursor de apuntador.

**CSS:**

body: Establece el estilo del cuerpo del documento, incluido el tipo de fuente, márgenes y relleno.

h1: Define el estilo del encabezado de nivel 1, incluyendo la alineación y el color del texto.

form: Establece el estilo del formulario, incluyendo el ancho máximo, el margen, el color de fondo, el relleno, el radio de borde y la sombra.

textarea: Define el estilo específico para las áreas de texto, incluyendo la altura fija, la restricción de redimensionamiento y el margen superior.

.css: Esto sugiere que hay estilos adicionales en un archivo CSS separado.

**Comentarios:**

Hay un comentario /\* styles.css \*/ al final del CSS, lo que sugiere que este archivo HTML podría estar vinculado a un archivo de estilos externo llamado "styles.css", donde se encontrarían los estilos adicionales no mostrados aquí.

**Observaciones adicionales:**

Se usan colores neutros y suaves para el fondo y el formulario para una apariencia amigable y accesible.

Se han definido los estilos tanto en línea como en un archivo CSS externo, lo que sugiere una práctica de desarrollo organizada y escalable.

Registro.html

Este código HTML representa una página de registro de usuario, junto con un encabezado de navegación y un contenedor de información de la empresa.

**HTML:**

<!DOCTYPE html>: Declaración del tipo de documento.

<html lang="en">: Elemento raíz del documento HTML con atributo de idioma establecido en inglés.

<head>: Sección que contiene metadatos y enlaces a recursos externos.

meta charset="UTF-8": Declara la codificación de caracteres UTF-8.

meta name="viewport" content="width=device-width, initial-scale=1.0": Establece la escala inicial y la anchura del viewport para dispositivos móviles.

title: Título de la página.

<link rel="stylesheet" href="estilos.css">: Enlace al archivo de estilos CSS local.

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">: Enlace a la fuente de iconos de Font Awesome.

<body>: Contenido visible de la página.

.container-hero: Contenedor para el encabezado de la página.

.container-logo, .container-user, .customer-support: Contenedores para elementos específicos del encabezado, como el logotipo, el carrito de compras y el soporte al cliente.

<nav>: Navegación principal de la página.

.container-navbar: Contenedor para la barra de navegación.

<div class="login-container">: Contenedor para el formulario de registro de usuario.

<form>: Formulario para que los usuarios ingresen sus datos de registro.

<input>: Campos de entrada para el nombre, contraseña, correo y teléfono del usuario.

<input type="submit">: Botón para enviar el formulario de registro.

**CSS:**

* Se referencia un archivo de estilos externo **estilos.css**, pero no se proporciona en el código.

**Observaciones:**

Se utilizan iconos de Font Awesome para elementos como el carrito de compras, el usuario y el soporte al cliente.

La página incluye una barra de navegación con enlaces a diferentes secciones del sitio web.

El formulario de registro está centrado en la página y contiene campos para el nombre, contraseña, correo y teléfono del usuario.

No se proporcionan acciones de formulario (action="#") ni métodos de envío (method="post"), lo que indica que el formulario no está completamente funcional en esta implementación.

Perfil.html

Este código HTML representa una página web que incluye un encabezado de navegación, información de perfil de usuario y un pie de página con enlaces y formulario de boletín informativo.

**HTML:**

<!DOCTYPE html>: Declaración del tipo de documento.

<html lang="en">: Elemento raíz del documento HTML con atributo de idioma establecido en inglés.

<head>: Sección que contiene metadatos y enlaces a recursos externos.

meta charset="UTF-8": Declara la codificación de caracteres UTF-8.

meta name="viewport" content="width=device-width, initial-scale=1.0": Establece la escala inicial y la anchura del viewport para dispositivos móviles.

title: Título de la página.

<link rel="stylesheet" href="estilos.css">: Enlace al archivo de estilos CSS local.

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">: Enlace a la fuente de iconos de Font Awesome.

<body>: Contenido visible de la página.

.container-hero: Contenedor para el encabezado de la página.

.container-navbar: Contenedor para la barra de navegación.

.main-content: Contenedor para el contenido principal de la página.

.profile: Contenedor para la información del perfil de usuario, incluyendo imagen de avatar y detalles personales.

.additional-info: Contenedor para información adicional del perfil.

.footer: Contenedor para el pie de página.

.container-footer: Contenedor principal para los elementos del pie de página.

.menu-footer: Contenedor para los enlaces de información y cuenta.

.newsletter: Contenedor para el formulario de boletín informativo.

**Observaciones:**

Se utilizan iconos de Font Awesome para elementos como el avatar, los enlaces de redes sociales y el formulario de boletín informativo.

La página incluye un enlace al archivo JavaScript de Font Awesome para el uso de sus iconos.

El pie de página contiene información de contacto, enlaces de información y cuenta, así como un formulario de boletín informativo para suscripciones.

En algunos casos, hay errores de redacción o atributos faltantes, como el atributo alt para la imagen de avatar y el enlace <img src=""> sin URL especificada.

meteosource.js

Este código es un script JavaScript que permite a los usuarios obtener el pronóstico del tiempo para una ciudad específica utilizando la API de OpenWeatherMap.

**JavaScript:**

const result = document.querySelector('.result');: Selecciona el elemento HTML con la clase "result", que se utiliza para mostrar el resultado del pronóstico del tiempo.

const form = document.querySelector('.get-weather');: Selecciona el formulario HTML con la clase "get-weather".

const nameCity = document.querySelector('#city'); y const nameCountry = document.querySelector('#country');: Selecciona los elementos de entrada de texto con los ID "city" y "country" respectivamente, que representan la ciudad y el país ingresados por el usuario.

form.addEventListener('submit', (e) => {...});: Agrega un event listener al formulario que escucha el evento de envío y ejecuta una función cuando se envía el formulario.

La función de callback para el evento de envío del formulario:

Evita que el formulario se envíe utilizando e.preventDefault();.

Obtiene los valores ingresados por el usuario para la ciudad y el país.

Verifica si se ingresaron valores válidos para la ciudad y el país. Si no, muestra un mensaje de error utilizando la función showError.

Llama a la función callAPI con los valores de ciudad y país ingresados por el usuario.

async function callAPI(city, country) {...}: Una función asincrónica que llama a la API de OpenWeatherMap para obtener el pronóstico del tiempo para la ciudad y el país especificados.

Construye la URL de la API utilizando la clave de API proporcionada y los parámetros de la ciudad, país y unidades métricas.

Utiliza fetch para hacer la solicitud a la API.

Maneja las respuestas de la API y muestra el pronóstico del tiempo llamando a las funciones showWeather y showError.

function showWeather(data) {...}: Una función que muestra el pronóstico del tiempo recibido de la API.

Extrae los datos relevantes (nombre de la ciudad, temperatura actual, temperatura mínima, temperatura máxima y el icono del clima) del objeto de datos recibido.

Crea un nuevo elemento HTML para mostrar esta información y lo agrega al elemento HTML con la clase "result".

* **function showError(message) {...}**: Una función que muestra mensajes de error.
  + Crea un elemento **<p>** con la clase "alert-message" y el mensaje de error recibido.
  + Agrega este elemento al formulario.
  + Borra el mensaje de error después de 3 segundos utilizando **setTimeout**.
* **function clearHTML() {...}**: Una función que limpia el contenido HTML dentro del elemento con la clase "result".

Este script permite a los usuarios obtener el pronóstico del tiempo para una ciudad y país específicos y maneja posibles errores de la API.

jugetes.html

Este código HTML representa una página web de comercio electrónico.

<!DOCTYPE html>: Define el tipo de documento como HTML5.

<html lang="en">: El elemento raíz del documento HTML con el atributo lang que especifica el idioma del contenido, en este caso, inglés.

<head>: Contiene metadatos y enlaces a recursos externos utilizados en la página.

<meta charset="UTF-8">: Define el conjunto de caracteres como UTF-8 para admitir caracteres especiales.

<meta name="viewport" content="width=device-width, initial-scale=1.0">: Configura la vista del contenido para dispositivos móviles.

<title>Login</title>: Establece el título de la página.

Enlaces a hojas de estilo CSS y fuentes de iconos.

<body>: Contiene todo el contenido visible de la página.

container-hero: Una sección que contiene información de contacto y un logotipo de la tienda.

container-navbar: Una sección que contiene la barra de navegación de la tienda.

navbar: La barra de navegación con elementos de menú y un formulario de búsqueda.

main: La sección principal de la página.

container-products: Contenedor para los productos disponibles en la tienda.

card-product: Cada tarjeta representa un producto.

container-img: Contiene la imagen del producto y opciones de interacción.

content-card-product: Contiene detalles del producto como título, precio y opciones de compra.

footer: El pie de página de la página web.

container-footer: Contenedor para los elementos del pie de página.

Secciones para información de contacto, enlaces útiles, información de cuenta del usuario y un formulario de boletín informativo.

copyright: Información de derechos de autor y créditos del desarrollador.

<script>: Enlace al archivo JavaScript que contiene lógica dinámica para la página.

Se utilizan íconos de FontAwesome para agregar iconos a diferentes partes de la página, como botones de acción, elementos de menú y enlaces de redes sociales.

En general, este código define una página web que incluye elementos de navegación, productos para la venta, información de contacto y un pie de página con enlaces útiles y opciones de suscripción.

js--2.js

Este código jQuery se encarga de realizar varias funciones en un formulario de solicitud de empleo:

Función para calcular la edad: calcularEdad(fechaNacimiento) calcula la edad en base a la fecha de nacimiento ingresada en el campo correspondiente. Utiliza la fecha actual para realizar el cálculo.

Función para validar el RUT: validarRut(rut) verifica si un RUT (Rol Único Tributario, un identificador fiscal en Chile) es válido. Utiliza una expresión regular y el algoritmo de validación de RUT chileno.

Función para generar la carta de presentación: generarCarta() recopila los datos del formulario y genera una carta de presentación. Verifica que todos los campos estén completos, valida el RUT, y verifica que la edad esté dentro del rango requerido (entre 18 y 35 años). Si algún campo no está completo o el RUT es inválido, muestra una alerta correspondiente. Si todo está correcto, genera la carta de presentación y la muestra en un área de texto.

Eventos de escucha:

Al hacer clic en el botón con el id #crearCartaBtn, se llama a la función generarCarta().

Cuando cambia el valor del campo de fecha de nacimiento (#fechaNacimiento), se calcula automáticamente la edad y se actualiza el campo #edad.

Cuando se enfoca en el campo de celular (#celular), se agrega automáticamente el prefijo "+569" si el campo está vacío.

En resumen, este código agrega funcionalidades dinámicas a un formulario de solicitud de empleo, como la validación del RUT, el cálculo automático de la edad y la generación de una carta de presentación. Además, proporciona una experiencia de usuario mejorada al agregar automáticamente el prefijo al número de celular.

Principio del formulario

inicioSesion.html

### Encabezado (Head)

* **Metaetiquetas**: Definen la codificación de caracteres y la configuración de la vista en dispositivos móviles.
* **Título**: Es el título de la página que se mostrará en la pestaña del navegador.
* **Enlaces CSS**: Se vincula a una hoja de estilos externa para aplicar estilos adicionales a la página.
* **Enlace a Font Awesome**: Se vincula a la biblioteca de iconos Font Awesome para usar iconos en la página.

### Cuerpo (Body)

#### Contenedor de Hero (Header)

* Contiene elementos como el número de soporte al cliente, el logotipo y el carrito de compras.
* Utiliza iconos de Font Awesome para el número de soporte y el logotipo de la tienda.

#### Barra de navegación (Navbar)

* Contiene enlaces a diferentes páginas de la aplicación, como "Inicio", "Registrarse", "Mi Perfil", etc.
* Incluye un formulario de búsqueda.

#### Contenedor de Inicio de Sesión

* Contiene un formulario de inicio de sesión con dos campos: usuario y contraseña.
* Se utiliza el método POST para enviar los datos del formulario a un script PHP llamado "guardar\_datos.php".

#### Script JavaScript

* Se encarga de manejar el evento de envío del formulario.
* Obtiene los valores ingresados por el usuario para el nombre de usuario y la contraseña.
* Realiza una validación de las credenciales.
* Si las credenciales son correctas, redirige al usuario a diferentes páginas según el tipo de usuario.
* Si las credenciales son incorrectas, muestra una alerta al usuario.

### Notas Adicionales

* Se recomienda que el formulario de inicio de sesión y el script de validación estén protegidos contra ataques de seguridad, como la inyección SQL y la validación de entrada.
* El redireccionamiento a páginas específicas debe ser adaptado según la estructura de tu aplicación y las necesidades específicas del flujo de trabajo de inicio de sesión.

Index.html

### Encabezado (Head)

* **Metaetiquetas**: Define la codificación de caracteres y la configuración de la vista en dispositivos móviles.
* **Título**: Es el título de la página que se mostrará en la pestaña del navegador.
* **Enlaces CSS**: Se vincula a una hoja de estilos externa para aplicar estilos adicionales a la página.
* **Enlace a Font Awesome**: Se vincula a la biblioteca de iconos Font Awesome para usar iconos en la página.

### Cuerpo (Body)

#### Sección Principal (Main)

* **Contenedor de Características**: Muestra varias características destacadas con iconos y descripciones.
* **Categorías Principales**: Muestra las principales categorías de productos con enlaces.
* **Productos Destacados**: Presenta una selección de productos destacados con imágenes, precios y opciones de interacción (vista, favoritos, añadir al carrito).
* **Galería de Imágenes**: Muestra una galería de imágenes con enlaces a recursos externos.
* **Productos Especiales**: Presenta una selección de productos especiales con imágenes, precios y opciones de interacción.
* **Enlace Externo**: Proporciona un enlace a un recurso externo.

#### Pie de Página (Footer)

* **Menú de Pie de Página**: Contiene secciones de información de contacto, información general, cuentas de usuario y suscripción al boletín.
* **Derechos de Autor**: Muestra la información de derechos de autor del sitio web.
* **Scripts Externos**: Se incluyen scripts externos, como Bootstrap y jQuery, para funcionalidades adicionales.

### Script JavaScript (API del Clima)

* Utiliza la API de OpenWeatherMap para obtener el pronóstico del tiempo de Santiago, Chile.
* Al hacer clic en un botón con id "apiButton", se realiza una solicitud a la API.
* Se filtra la respuesta para obtener el pronóstico diario y se muestra en tarjetas en la página.

### Notas Adicionales

* Se utilizan iconos de Font Awesome para mejorar la experiencia visual.
* Se recomienda asegurar la validez y la seguridad de los enlaces externos y scripts.
* Se utiliza Bootstrap para la maquetación y la funcionalidad de la página, lo que proporciona un diseño receptivo y funcionalidades adicionales.