

¡Excelente! Profundicemos en los ejercicios de Front-End para tu proyecto de venta de comidas. A continuación, te presento cada ejercicio detallado, con ejemplos y desafíos adicionales para que puedas llevar tus habilidades al siguiente nivel.

## 1. Menú de comida interactivo

- **Objetivo:** Crear un menú de comida atractivo y funcional utilizando HTML, CSS y JavaScript.
- **Estructura HTML:**
  - Utiliza etiquetas semánticas como `<section>`, `<article>`, `<figure>`, `<figcaption>`, etc. para estructurar el menú.
  - Cada plato debe tener una imagen ( `<img>` ), un título ( `<h3>` ), una descripción ( `<p>` ) y un precio.
  - Considera utilizar una lista ( `<ul>` o `<ol>` ) para organizar los platos.
- **Diseño CSS:**
  - Utiliza Flexbox o Grid para crear un diseño responsive y adaptable a diferentes tamaños de pantalla.
  - Establece estilos para las imágenes, los títulos, las descripciones y los precios.
  - Agrega efectos visuales sutiles al pasar el mouse sobre los platos (por ejemplo, un ligero cambio de opacidad o una sombra).
- **Interactividad con JavaScript:**
  - Al hacer clic en un plato, muestra una ventana emergente o un panel con más detalles (ingredientes, información nutricional, etc.).
  - Implementa un carrito de compras básico:
    - Crea un array para almacenar los platos seleccionados.
    - Al hacer clic en "Agregar al carrito", añade el plato al array.
    - Actualiza el carrito de compras (ya sea un contador o una lista) con los platos añadidos.
    - Calcula y muestra el total del pedido.
- **Desafíos adicionales:**
  - **Filtros:** Agrega filtros para que el usuario pueda ver platos por categoría (entradas, platos principales, postres, etc.).
  - **Buscar:** Implementa un buscador para encontrar platos por nombre o descripción.
  - **Animaciones:** Utiliza CSS o JavaScript para agregar animaciones sutiles al agregar platos al carrito o al cambiar entre categorías.

## 2. Formulario de contacto con validación

- **Objetivo:** Crear un formulario de contacto funcional y seguro utilizando HTML, CSS y JavaScript.
- **Estructura HTML:**
  - Utiliza etiquetas `<form>`, `<label>`, `<input>`, `<textarea>`, `<select>`, `<option>`, etc. para crear el formulario.
  - Incluye campos para nombre, correo electrónico, mensaje y otros que consideres necesarios.
  - Agrega un campo para adjuntar archivos ( `<input type="file">` ).
- **Diseño CSS:**
  - Diseña el formulario de manera que sea visualmente atractivo y fácil de usar.

- Utiliza estilos para las etiquetas, los campos de entrada y el botón de envío.
- **Validación con JavaScript:**
  - Antes de enviar el formulario, utiliza JavaScript para validar los campos:
    - Verifica que todos los campos obligatorios estén llenos.
    - Asegúrate de que el correo electrónico tenga un formato válido.
    - Valida la longitud de los campos de texto (por ejemplo, el mensaje).
    - Para el campo de archivo, puedes verificar el tipo de archivo y el tamaño máximo permitido.
  - Muestra mensajes de error claros y específicos para cada campo que no cumpla con los requisitos.
- **Desafíos adicionales:**
  - **Validación en tiempo real:** Valida los campos a medida que el usuario los escribe para brindar retroalimentación inmediata.
  - **Envío del formulario:** Utiliza AJAX para enviar el formulario sin necesidad de recargar la página.
  - **Diseño responsive:** Asegúrate de que el formulario se adapte correctamente a diferentes tamaños de pantalla.

### 3. Galería de imágenes con efectos

- **Objetivo:** Crear una galería de imágenes atractiva y optimizada utilizando HTML, CSS y JavaScript.
- **Estructura HTML:**
  - Utiliza etiquetas `<img>` para mostrar las imágenes.
  - Considera usar elementos como `<figure>` y `<figcaption>` para agregar títulos y descripciones a las imágenes.
  - Puedes organizar las imágenes en una cuadrícula utilizando Flexbox o Grid.
- **Diseño CSS:**
  - Asegúrate de que las imágenes se vean bien en diferentes tamaños de pantalla.
  - Agrega efectos visuales al pasar el mouse sobre las imágenes:
    - **Zoom:** Utiliza la propiedad `transform: scale()` para hacer zoom en la imagen.
    - **Opacidad:** Cambia la opacidad de la imagen con `opacity`.
    - **Transiciones:** Utiliza `transition` para crear animaciones suaves.
- **JavaScript (opcional):**
  - Si quieres implementar un carrusel de imágenes, necesitarás JavaScript para controlar la navegación y la animación.
- **Desafíos adicionales:**
  - **Optimización de imágenes:** Utiliza imágenes de alta calidad pero optimizadas para la web para que la página cargue rápido.
  - **Lightbox:** Al hacer clic en una imagen, muestra una versión más grande en una ventana emergente (lightbox).
  - **Animaciones:** Agrega animaciones sutiles al cargar las imágenes o al cambiar entre ellas.

### 4. Animaciones con CSS

- **Objetivo:** Crear animaciones CSS atractivas y funcionales para elementos de tu página web.
- **Tipos de animaciones:**

- **Transiciones:** Se utilizan para crear animaciones sencillas al cambiar propiedades CSS (por ejemplo, al pasar el mouse sobre un elemento).
- **Transformaciones:** Permiten modificar la apariencia de un elemento (por ejemplo, rotar, escalar, desplazar).
- **Keyframes:** Se utilizan para crear animaciones más complejas y controladas en el tiempo.
- **Propiedades CSS:\***
  - `opacity` , `transform` , `transition` , `animation` , `animation-duration` , `animation-timing-function` , `animation-delay` , etc.
- **Ejemplos de animaciones:\***
  - Botones que cambian de color o se hacen más grandes al pasar el mouse.
  - Encabezados que se deslizan o aparecen con una transición suave.
  - Menús desplegados con animaciones al abrir y cerrar.
  - Elementos que se desvanecen o se desplazan al hacer scroll en la página.
- **Desafíos adicionales:\***
  - **Animaciones interactivas:** Crea animaciones que se disparen al hacer clic en elementos específicos o al realizar alguna acción.
  - **Animaciones de carga:** Diseña animaciones que se muestren mientras la página web está cargando.
  - **Performance:** Asegúrate de que las animaciones sean fluidas y no afecten negativamente el rendimiento de la página.

## Recomendaciones adicionales

- **Diseño responsive:** Asegúrate de que todos los elementos se adapten correctamente a diferentes tamaños de pantalla utilizando media queries en CSS.
- **Accesibilidad:** Ten en cuenta la accesibilidad al diseñar tu página web. Utiliza colores con suficiente contraste, asegúrate de que las imágenes tengan atributos `alt` descriptivos y utiliza etiquetas semánticas HTML.
- **Optimización:** Optimiza el código HTML, CSS y JavaScript para que la página cargue rápido y funcione de manera eficiente.

¡Espero que estos ejercicios detallados te sean de gran ayuda para prepararte para tu proyecto integrador! Recuerda que la práctica constante y la exploración de nuevas técnicas son clave para convertirte en un experto en Front-End. Si tienes alguna pregunta o necesitas más orientación, no dudes en consultarme. ¡Mucho éxito en tu proyecto!