

## Parcial - Desarrollo de Sitio Web Interactivo

**Materia:** Construcción de Interfaces de Usuario

**Tema:** Sitio Web de una Cafetería

**Objetivo:** Aplicar los conocimientos aprendidos durante el curso para diseñar y desarrollar un sitio web estático e interactivo, respetando buenas prácticas de UX/UI, HTML, CSS, JavaScript, y asincronía.

### Contexto de la consigna

**¡Nos contrató el dueño de una nueva cafetería llamada "Café Aurora"!** ☕  
Él necesita una **página web** sencilla pero atractiva, donde los clientes puedan:

- Conocer la cafetería y su estilo.
- Consultar el menú de productos actualizados (cafés, té, postres).
- Dejar sus datos de contacto para recibir promociones o realizar consultas.

El dueño quiere que el sitio sea **visual**, **fácil de usar**, y **dinámico**, con pequeños efectos interactivos para mejorar la experiencia del usuario.

### Consigna General

Deberás crear un **sitio web completo** para **Café Aurora**, aplicando los temas trabajados en clase.

El sitio debe ser funcional, visualmente atractivo y debe incluir interactividad mediante JavaScript.

# Requisitos del Trabajo

## 1. Estructura del Sitio (HTML)

- Son dadas 3 **páginas para utilizar**:
  - **Inicio** (Presentación de la cafetería).
  - **Menú de Productos** (Listado de cafés, té, postres).
  - **Contacto** (Formulario de contacto o suscripción).
- Si es necesario se puede agregar elementos o propiedades, pero no modificar las propiedades dadas. Por ejemplo: no modificar los id asignados

## 2. Estilos (CSS)

- Es dado un archivo CSS para MEJORAR, ya que no respeta las buenas prácticas del diseño. Se puede modificar todo lo necesario del archivo css
- Utilizar:
  - Fuentes atractivas (puede ser de **Google Fonts**).
  - **Colores consistentes** que respeten una identidad visual (pueden inspirarse en tonos cafeteros: marrones, beige, crema).
  - Efectos de **hover** en botones o imágenes.
- Incorporar un moodboard sencillo como referencia de diseño (opcional si se desea sumar creatividad).

## 3. Interactividad (JavaScript)

Se pide desarrollar:

### - Menu.html:

- **Filtro**: aplicar la logica necesaria para que al ingresar texto en el input con id "filtro", se vaya filtrando de la lista de productos los que tienen el texto ingresado en el nombre de producto. Debe ir filtrando a medida que el usuario va ingresando texto
- **Agregar Producto**: al hacer clic en el boton con id "btnAgregar" se debe visualizar la sección con id "seccionAgregar"
- **Guardar Nuevo Producto**: al hacer clic en el boton con id "btnGuardarNuevo" se debe validar que los campos de la "seccionAgregar" estén correctamente completados y crear el nuevo elemento en la lista de productos con sus datos y botones. Al finalizar debe ocultar la "secciónAgregar"
- **Quitar y Modificar**: cada elemento de la lista de productos debe tener dos botones "btnQuitar" y "btnModificar" respectivamente. Se debe desarrollar la logica para

eliminar el producto en el que se presionó el botón quitar, y permitir modificar el precio al presionar "Modificar"

- **Lista de productos:** debe iluminarse la fila del producto cuando se pasa por arriba con el mouse
- **Contador de productos seleccionados:** Cada producto de la lista debe incluir un checkbox asociado. Agregar un elemento HTML con `id="contadorSeleccionados"` donde se mostrará la cantidad total de productos seleccionados. Cada vez que el usuario marque o desmarque un checkbox, el contador debe actualizarse automáticamente para reflejar el número de productos seleccionados. Utilizar el evento `change` para detectar los cambios de selección.

#### - Contacto.html:

- **Formulario de contacto:** que valide en tiempo real los valores ingresados en los campos dados (por ejemplo, que advierta si el email no es válido antes de enviar). Y solo permita enviar el formulario en caso que todo este correcto. La validación la tienen que hacer con Javascript.

## 4. Asincronía

Simular una carga de datos asincrónica usando `setTimeout`, `fetch` con datos falsos o una Promesa.

Ejemplo: cargar "recomendaciones de la casa" después de unos segundos de entrar a la página de menú.

## 6. Buenas Prácticas de UX/UI

- Diseño limpio y organizado.
- Navegación intuitiva (barra de navegación visible en todas las páginas).
- Formularios accesibles (etiquetas correctas en los campos).

**Framework CSS:** Pueden utilizar un framework CSS (como Bootstrap, Material UI o Tailwind) para estilizar la aplicación.

## Estructura de Carpetas Sugerida

cafeteria/		
—	index.html	--> Página de Inicio
—	menu.html	--> Página del Menú de Productos
—	contacto.html	--> Página de Contacto
—	css/	
	estilos.css	--> Archivo de estilos principal
—	js/	
	scripts.js	--> Archivo principal de JavaScript
—	img/	
	logo.png	--> (opcional) Logo de la cafetería
	fondo.jpg	--> (opcional) Imagen de fondo
	productos/	--> (opcional) Carpeta para imágenes de cafés, postres, etc.
	cafe1.jpg	
	cafe2.jpg	
	postre1.jpg	
—	README.txt	--> Un archivo con una breve descripción del proyecto