SERVICIO NACIONAL DE APRENDIZAJE — SENA

SENA

Técnico en Programación de Software

Taller 2

HTML5 y CSS

Menús de navegación · Galerías de imágenes · Formularios interactivos

Separación de contenido (HTML) y presentación visual (CSS)

Programa:	Programación de Software — Código 233104 V.2
Competencia:	Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo
Resultado:	Crear componentes front-end del software de acuerdo con las necesidades del cliente.
Duración:	54 horas
Herramientas:	Visual Studio Code, Live Server, Navegador web (Chrome/Firefox)
Instructor:	Elver Siaucho
Prerequisito:	Taller 1 — Etiquetas HTML (completado)

Introducción

En el Taller 1 aprendiste a estructurar el contenido de una página web usando etiquetas HTML. Ahora darás el siguiente paso: dotar de vida visual a esa estructura mediante CSS (Cascading Style Sheets — Hojas de Estilo en Cascada).

CSS es el lenguaje que controla el aspecto visual de una página web: colores, tipografías, espaciados, animaciones, diseño de columnas y mucho más. La gran ventaja es que un solo archivo .css puede estilizar cientos de páginas HTML, logrando consistencia visual y ahorrando trabajo.

Objetivo del taller:
Al finalizar este taller el aprendiz sera capaz de:
- Vincular archivos CSS externos a documentos HTML.
- Aplicar selectores, propiedades y valores CSS para estilizar cualquier elemento.
- Construir menús de navegación horizontales y verticales con efectos hover.
- Diseñar galerías de imágenes con CSS Grid y Flexbox.
- Crear formularios interactivos visualmente atractivos y con validación visual.
- Aplicar variables CSS, transiciones y animaciones básicas.
- Implementar diseño responsivo con Media Queries.

Antes de comenzar:
1. Tener completado el Taller 1 (carpeta Taller1_HTML con todos los ejercicios).
2. Crear una carpeta nueva llamada Taller2_CSS para guardar todos los archivos de este taller.
3. Dentro de Taller2_CSS crear subcarpeta css/ para las hojas de estilo y imagenes/ para las fotos.
4. Todos los ejercicios deben tener un archivo HTML + su archivo CSS externo vinculado.
5. Nombrar cada entregable según las indicaciones de cada ejercicio.

Resumen de ejercicios y distribución de tiempo

#	Ejercicio	Duración	Nivel	Entregable
1	Introducción a CSS: Selectores y propiedades	4 h	Básico	`ejercicio1/`
2	Colores, tipografías y el modelo de caja	5 h	Básico	`ejercicio2/`
3	Flexbox: diseño de layouts modernos	6 h	Básico	`ejercicio3/`
4	CSS Grid: cuadrículas y galerías de imágenes	6 h	Intermedio	`ejercicio4/`
5	Menú de navegación horizontal con hover	6 h	Intermedio	`ejercicio5/`
6	Menú de navegación vertical y sidebar	5 h	Intermedio	`ejercicio6/`
7	Formularios interactivos y validación visual	7 h	Intermedio	`ejercicio7/`

8	Variables CSS y animaciones / transiciones	5 h	Intermedio	`ejercicio8/`
9	Diseño responsivo con Media Queries	5 h	Avanzado	`ejercicio9/`
10	Proyecto integrador: Sitio web estilizado	5 h	Avanzado	`sitio_css/`

Ej. 1	Introducción a CSS: Selectores y Propiedades	4 horas	Básico

Concepto: ¿Cómo funciona CSS?

CSS se vincula al HTML con <link rel="stylesheet" href="css/estilos.css"> dentro del <head>.

Una regla CSS tiene dos partes: el SELECTOR (a quién afecta) y las PROPIEDADES (qué cambia).

selector { propiedad: valor; }

Tipos de selectores más usados:

- Etiqueta: p { color: red; } afecta a todos los <p>
- Clase: .titulo { font-size: 24px; } afecta a elementos con class="titulo"
- ID: #header { background: blue; } afecta al elemento con id="header"
- Descendente: nav a { color: white; } afecta a <a> dentro de <nav>

Ejemplo de referencia — index.html:

```
<!-- index.html -->
<head>
  <link rel="stylesheet" href="css/estilos.css">
</head>
<body>
  <h1 id="titulo-principal">Mi Página con CSS</h1>
  <p class="destacado">Este párrafo tiene estilo especial.</p>
  <p>Este párrafo tiene el estilo por defecto.</p>
</body>
```

Ejemplo de referencia — css/estilos.css:

```
/* Selector de etiqueta */
body {
  font-family: 'Segoe UI', sans-serif;
  background-color: #f0f4f8;
  margin: 0;
  padding: 20px;
}

/* Selector de ID */
#titulo-principal {
  color: #003087;
  font-size: 2rem;
  text-align: center;
}

/* Selector de clase */
.destacado {
  color: #C0392B;
  font-weight: bold;
  background-color: #FDECEA;
  padding: 8px;
  border-left: 4px solid #C0392B;
}
```

Actividad — Carpeta: ejercicio1/

Crea una página HTML llamada index.html y su hoja de estilos css/estilos.css. Debes estilizar:

1. El body con una fuente moderna, color de fondo suave y márgenes cero.
2. Un <h1> centrado con color azul, tamaño grande y una línea inferior decorativa (border-bottom).
3. Tres <p> con diferentes clases (.intro, .cuerpo, .nota) cada uno con color, tamaño y fondo distintos.
4. Una lista <ul> donde cada <li> tenga un color de fondo alterno (usa nth-child).
5. Un <a> sin subrayado que cambie de color al pasar el cursor (:hover).
6. Incluye comentarios CSS explicando cada sección.

Ej. 2	Colores, Tipografías y el Modelo de Caja	5 horas	Básico

Concepto: Box Model (Modelo de Caja)
Todo elemento HTML es una caja rectangular con 4 capas (de adentro hacia afuera):
CONTENT -> PADDING (relleno interior) -> BORDER (borde) -> MARGIN (margen exterior)
Propiedades clave:
width / height : tamaño del contenido
padding: 10px 20px : relleno vertical y horizontal
border: 2px solid red : grosor, estilo y color del borde
margin: 0 auto : centrado horizontal automático
border-radius: 8px : esquinas redondeadas
box-shadow: 2px 2px 8px rgba(0,0,0,0.2) : sombra
box-sizing: border-box : el padding NO aumenta el ancho total (¡siempre úsalo!)

Ejemplo de referencia — css/estilos.css:

```
/* Reset universal */
* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

/* Importar fuente de Google Fonts */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap
');

body {
  font-family: 'Poppins', sans-serif;
  background: #f5f5f5;
}

.tarjeta {
  width: 300px;
  background: #ffffff;
  border-radius: 12px;
  border: 1px solid #ddd;
  padding: 24px;
  margin: 20px auto;
  box-shadow: 0 4px 16px rgba(0, 0, 0, 0.10);
}

.tarjeta h2 {
  color: #003087;
  margin-bottom: 12px;
  border-bottom: 2px solid #1A6FBF;
  padding-bottom: 8px;
}
```

Actividad — Carpeta: ejercicio2/

Diseña una página de "Tarjetas de Productos" para una tienda ficticia. Requiere:

1. Importar una fuente de Google Fonts (fonts.google.com) y aplicarla a toda la página.
2. Crear al menos 4 tarjetas .card con: imagen, nombre, precio y botón de compra.

3. Cada tarjeta debe tener: padding, border-radius, box-shadow y un hover que eleve la tarjeta (translateY).
4. El botón de cada tarjeta debe tener: padding, border-radius, color de fondo, sin borde y cursor:pointer.
5. Usa colores con formato hexadecimal (#), RGB y HSL en distintos elementos.
6. Aplica el reset universal * { box-sizing: border-box; margin: 0; padding: 0; } al inicio del CSS.

Ej. 3	Flexbox: Diseño de Layouts Modernos	6 horas	Básico

Concepto: CSS Flexbox

Flexbox es un sistema de diseño de una dimensión (fila O columna). Se activa con:
display: flex;
Propiedades del contenedor (padre):
flex-direction: row | column — dirección de los elementos
justify-content: center | space-between | flex-end — alineación en el eje principal
align-items: center | flex-start | stretch — alineación en el eje cruzado
flex-wrap: wrap — permite que los items bajen a la siguiente línea
gap: 16px — espacio entre elementos (mucho más limpio que margin)
Propiedades del elemento hijo:
flex: 1 — ocupa el espacio disponible proporcionalmente
flex: 0 0 200px — tamaño fijo sin crecer ni encoger

Ejemplo de referencia — css/estilos.css:

```
/* Layout de página completa con Flexbox */
.pagina {
  display: flex;
  flex-direction: column;
  min-height: 100vh;
}

/* Barra de navegación con Flexbox */
nav {
  display: flex;
  justify-content: space-between;
  align-items: center;
  background: #003087;
  padding: 0 32px;
  height: 60px;
}

/* Tarjetas que se ajustan automáticamente */
.contenedor-tarjetas {
  display: flex;
  flex-wrap: wrap;
  gap: 20px;
  justify-content: center;
  padding: 32px;
}

.tarjeta {
  flex: 0 0 250px;   /* Ancho fijo de 250px */
  background: white;
  border-radius: 10px;
  padding: 20px;
}
```

Actividad — Carpeta: ejercicio3/

Construye una página de portafolio personal usando Flexbox para TODA la estructura. Debes implementar:

1. Header con logo a la izquierda y menú de texto (sin estilos complejos aún) a la derecha usando justify-content: space-between.
2. Sección hero centrada con flex-direction: column y align-items: center.

3. Sección de habilidades con tarjetas que usen flex-wrap: wrap y gap.
4. Sección de dos columnas (contenido + sidebar) con flex: 2 y flex: 1 respectivamente.
5. Footer con 3 columnas iguales usando flex: 1.
6. Cada sección debe tener padding generoso y colores de fondo diferenciados.

Ej. 4	CSS Grid: Cuadrículas y Galerías de Imágenes	6 horas	Intermedio

Concepto: CSS Grid

Grid es un sistema de diseño de DOS dimensiones (filas Y columnas). Se activa con:
  display: grid;
Propiedades clave del contenedor:
  grid-template-columns: repeat(3, 1fr) — 3 columnas de igual ancho
  grid-template-columns: 200px 1fr 1fr — primera fija, las demás flexibles
  grid-template-rows: auto — filas se ajustan al contenido
  gap: 16px — espacio entre celdas
Propiedades del elemento hijo:
  grid-column: 1 / 3 — ocupa desde la línea 1 hasta la 3 (2 columnas)
  grid-row: 1 / 2 — ocupa desde la fila 1 hasta la 2
  grid-column: span 2 — ocupa 2 columnas desde su posición actual

Ejemplo de referencia — css/estilos.css:

```
/* Galería de 3 columnas responsiva */
.galeria {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 12px;
  padding: 24px;
}

.galeria img {
  width: 100%;
  height: 220px;
  object-fit: cover;       /* Recorta la imagen manteniendo proporción */
  border-radius: 8px;
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.galeria img:hover {
  transform: scale(1.04);
  box-shadow: 0 8px 24px rgba(0, 0, 0, 0.25);
  cursor: pointer;
}

/* Imagen destacada que ocupa 2 columnas */
.galeria .destacada {
  grid-column: span 2;
  height: 300px;
}
```

Actividad — Carpeta: ejercicio4/

Crea una galería de fotografías temática (naturaleza, ciudades, tecnología, etc.). Debe contener:

1. Una galería principal con display: grid y grid-template-columns: repeat(3, 1fr) con mínimo 9 imágenes.
2. Al menos 2 imágenes «destacadas» que usen grid-column: span 2 para ocupar más espacio.
3. Efecto hover en cada imagen: escala (transform: scale) y sombra (box-shadow).
4. Un overlay de texto sobre cada imagen al hacer hover (posición absoluta dentro del grid item).

5. Una segunda sección con display: grid; grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); para demostrar cuadrículas automáticas.
6. Todas las imágenes deben tener object-fit: cover para uniformar su tamaño.

Consejo práctico:
Puedes usar imágenes gratuitas de https://picsum.photos/400/300 (cambia los números por el tamaño deseado)
o de https://unsplash.com para fotografías de alta calidad. Guárdalas en tu carpeta imagenes/.

Ej. 5	Menú de Navegación Horizontal con Hover	6 horas	Intermedio

Concepto: Menús horizontales con CSS

Los menús de navegación profesionales se construyen con <nav>
<a>. Con CSS:

- Se elimina el estilo de lista: list-style: none; y padding: 0; margin: 0;
- Los <li> se ponen en fila: display: flex; en el
- Los <a> reciben padding para área de clic amplia: padding: 16px 24px;
- La propiedad :hover cambia el estado al pasar el cursor.
- transition: all 0.3s ease; suaviza cualquier cambio de propiedad CSS.

Los submenús se crean con listas anidadas y position: absolute.

Ejemplo de referencia:

```
/* === MENÚ HORIZONTAL === */
nav {
  background: #003087;
  position: sticky;     /* Se queda fijo al hacer scroll */
  top: 0;
  z-index: 100;
}

nav ul {
  list-style: none;
  margin: 0;
  padding: 0;
  display: flex;
  align-items: center;
}

nav ul li a {
  display: block;
  color: #ffffff;
  text-decoration: none;
  padding: 18px 24px;
  font-weight: 600;
  transition: background 0.25s ease, color 0.25s ease;
}

nav ul li a:hover {
  background: #1A6FBF;
  color: #FFC107;
}

/* Submenú desplegable */
nav ul li { position: relative; }

nav ul li ul {
  display: none;          /* Oculto por defecto */
  position: absolute;
  top: 100%;
  left: 0;
  background: #002060;
  min-width: 180px;
  border-top: 3px solid #FFC107;
}

nav ul li:hover ul {
  display: block;         /* Mostrar al hover del padre */
}

nav ul li ul li a {
```

```
  padding: 12px 20px;
  font-size: 0.9rem;
}
```

Actividad — Carpeta: ejercicio5/

Diseña una página web para una empresa ficticia con un menú de navegación horizontal profesional. Debe incluir:

1. Menú con al menos 5 ítems principales (Inicio, Nosotros, Servicios, Portafolio, Contacto).
2. Al menos 2 ítems del menú deben tener submenús desplegables con 3 opciones cada uno.
3. El menú debe ser sticky (position: sticky; top: 0) para que se fije al hacer scroll.
4. El ítem activo/actual debe tener un estilo diferente (clase .activo con border-bottom o background distinto).
5. Transiciones suaves en todos los cambios de hover (transition: all 0.3s ease).
6. El submenú debe mostrar una línea de color en el top (border-top) para diferenciarse.
7. El resto de la página debe tener contenido de relleno (mínimo 3 secciones) para poder probar el scroll.

Ej. 6	Menú de Navegación Vertical y Sidebar	5 horas	Intermedio

Concepto: Menús verticales y paneles laterales

Los paneles laterales (sidebar) son comunes en dashboards y paneles de administración.
Se implementan con un layout de 2 columnas (Flexbox o Grid) donde:
- El sidebar tiene un ancho fijo (ej: 240px o 260px).
- El contenido principal ocupa el espacio restante (flex: 1).
- position: fixed; height: 100vh; permite un sidebar fijo en pantalla.

Los ítems del menú vertical son bloques completos (display: block) con padding amplio.
Los iconos se pueden agregar con caracteres Unicode o la librería Font Awesome.

Ejemplo de referencia:

```
/* Layout con sidebar fijo */
.app {
  display: flex;
  min-height: 100vh;
}

.sidebar {
  width: 250px;
  background: #1a1a2e;
  height: 100vh;
  position: sticky;
  top: 0;
  display: flex;
  flex-direction: column;
  padding: 0;
  overflow-y: auto;
}

.sidebar .logo {
  padding: 24px 20px;
  font-size: 1.4rem;
  font-weight: 700;
  color: #FFC107;
  border-bottom: 1px solid rgba(255,255,255,0.1);
}

.sidebar ul { list-style: none; padding: 12px 0; margin: 0; }

.sidebar ul li a {
  display: flex;
  align-items: center;
  gap: 12px;
  padding: 14px 20px;
  color: #a0a8b8;
  text-decoration: none;
  font-size: 0.95rem;
  transition: all 0.2s ease;
  border-left: 3px solid transparent;
}

.sidebar ul li a:hover,
.sidebar ul li a.activo {
  background: rgba(255,255,255,0.08);
  color: #ffffff;
  border-left-color: #1A6FBF;
}

.contenido-principal {
```

```
  flex: 1;
  background: #f0f4f8;
  padding: 32px;
  overflow-y: auto;
}
```

Actividad — Carpeta: ejercicio6/

Construye la interfaz de un panel de administración (dashboard) con sidebar vertical. Debe tener:

1. Sidebar con logo/nombre de la app, menú de navegación vertical con 6 ítems (Dashboard, Usuarios, Productos, Pedidos, Reportes, Configuración).
2. El sidebar debe tener fondo oscuro y el ítem activo resaltado con border-left de color.
3. Área de contenido principal que muestre tarjetas de estadísticas (4 tarjetas con número grande y descripción).
4. Una tabla de datos simple estilizada con CSS (zebra striping: filas alternas de color).
5. El sidebar debe mantenerse fijo al hacer scroll con position: sticky; top: 0; height: 100vh.
6. Agregar sección de categorías colapsables (submenús) con :hover o clase CSS dentro del sidebar.

Ej. 7	Formularios Interactivos y Validación Visual	7 horas	Intermedio

Concepto: Formularios estilizados con CSS

Los formularios HTML ganan vida con CSS. Las pseudoclases clave son:

:focus — cuando el campo está seleccionado (clic o tabulación)
:hover — cuando el cursor está encima
:valid — cuando el valor cumple la validación HTML (required, type, pattern)
:invalid — cuando el valor NO cumple la validación
:placeholder-shown — cuando el campo muestra el placeholder

Estilos esenciales para un input profesional:

outline: none; (quita el borde azul del navegador)
border: 2px solid #ccc; + :focus { border-color: #1A6FBF; }
transition: border-color 0.3s ease, box-shadow 0.3s ease;

Ejemplo de referencia:

```
/* Reset de formulario */
input, select, textarea, button {
  font-family: inherit;
  font-size: 1rem;
}

/* Campo de texto estilizado */
.campo {
  margin-bottom: 20px;
}

.campo label {
  display: block;
  font-weight: 600;
  margin-bottom: 6px;
  color: #333;
}

.campo input,
.campo select,
.campo textarea {
  width: 100%;
  padding: 12px 16px;
  border: 2px solid #ddd;
  border-radius: 8px;
  background: #fafafa;
  outline: none;
  transition: border-color 0.3s, box-shadow 0.3s;
}

.campo input:focus,
.campo select:focus,
.campo textarea:focus {
  border-color: #1A6FBF;
  box-shadow: 0 0 0 3px rgba(26, 111, 191, 0.15);
  background: #ffffff;
}

/* Validación visual */
.campo input:valid  { border-color: #1B7A3E; }
.campo input:invalid:not(:placeholder-shown) { border-color: #C0392B; }

/* Botón de envío */
```

```css
.btn-enviar {
  background: #003087;
  color: white;
  border: none;
  padding: 14px 40px;
  border-radius: 8px;
  font-size: 1rem;
  font-weight: 700;
  cursor: pointer;
  transition: background 0.3s, transform 0.1s;
}
.btn-enviar:hover  { background: #1A6FBF; }
.btn-enviar:active { transform: scale(0.97); }
```

Actividad — Carpeta: ejercicio7/

Diseña una página con tres formularios completamente estilizados:

Formulario 1: Inicio de Sesión

1. Campos: usuario/email y contraseña con sus labels. Botón de login.
2. Fondo centrado con sombra (tarjeta de login), bordes redondeados.
3. Ojo para mostrar/ocultar contraseña (solo HTML+CSS, sin JS).

Formulario 2: Registro Completo

4. Campos: nombre, apellido, email, teléfono, fecha de nacimiento, ciudad (select), género (radio), intereses (checkbox), foto (file), comentarios (textarea).
5. Diseño en 2 columnas usando CSS Grid para los campos cortos.
6. Validación visual con :valid (borde verde) e :invalid:not(:placeholder-shown) (borde rojo).
7. Mensajes de error CSS debajo de cada campo usando ::after y el selector :invalid.

Formulario 3: Encuesta con barra de progreso

8. Simular una barra de progreso decorativa con CSS (div con width progresivo).
9. Mínimo 5 preguntas con diferentes tipos de input estilizados.
10. Checkboxes y radio buttons personalizados (sin el estilo del navegador).

Ej. 8	Variables CSS y Animaciones / Transiciones	5 horas	Intermedio

Concepto: Variables CSS y animaciones

Las variables CSS (custom properties) se definen en :root y se reutilizan con var():
:root { --color-primario: #003087; --espaciado: 16px; }
.boton { background: var(--color-primario); padding: var(--espaciado); }
Ventaja: cambiar la variable actualiza TODOS los elementos que la usan.

Las TRANSICIONES suavizan cambios de una propiedad:
transition: propiedad duración función-de-tiempo;
transition: all 0.3s ease;

Las ANIMACIONES definen movimiento con @keyframes:
@keyframes nombre { from { ... } to { ... } }
.elemento { animation: nombre 2s ease infinite; }

Ejemplo de referencia:

```
/* Variables globales */
:root {
  --primario:    #003087;
  --secundario:  #1A6FBF;
  --acento:      #FFC107;
  --texto:       #333333;
  --fondo:       #f5f7fa;
  --radio:       10px;
  --sombra:      0 4px 20px rgba(0,0,0,0.1);
}

/* Cambiar tema oscuro fácilmente */
.modo-oscuro {
  --primario:  #4A9EF5;
  --fondo:     #121212;
  --texto:     #e0e0e0;
}

/* Animación de entrada */
@keyframes entradaArriba {
  from { opacity: 0; transform: translateY(-30px); }
  to   { opacity: 1; transform: translateY(0); }
}

.encabezado {
  animation: entradaArriba 0.6s ease forwards;
}

/* Loader / spinner */
@keyframes girar {
  to { transform: rotate(360deg); }
}

.spinner {
  width: 48px;
  height: 48px;
  border: 5px solid #ddd;
  border-top-color: var(--primario);
  border-radius: 50%;
  animation: girar 0.8s linear infinite;
}
```

Actividad — Carpeta: ejercicio8/

Crea una página de demostración de animaciones CSS que incluya:

1. Un archivo variables.css con mínimo 10 variables CSS (colores, tipografías, espaciados, radios, sombras).
2. Un botón de cambio de tema (claro/oscuro) usando solo CSS: al añadir una clase .oscuro al body deben cambiar todas las variables y por ende todos los colores.
3. 3 animaciones @keyframes diferentes: (a) entrada con fadeIn+translateY, (b) pulso/latido para un ícono, (c) efecto de escritura en un título.
4. Un spinner de carga animado con @keyframes girar.
5. Tarjetas con transición hover que combinen: scale, box-shadow y border-color simultáneamente.
6. Una barra de progreso animada que pase de 0% a 100% al cargar la página.

Ej. 9	Diseño Responsivo con Media Queries	5 horas	Avanzado

Concepto: Diseño adaptable a cualquier pantalla

Un sitio responsivo se adapta a diferentes tamaños de pantalla (móvil, tablet, escritorio).
Las Media Queries aplican estilos solo cuando se cumple una condición de tamaño:
  @media (max-width: 768px) { /* estilos para pantallas <= 768px */ }
  @media (min-width: 1024px) { /* estilos para pantallas >= 1024px */ }
Breakpoints comunes:
  Móvil: max-width: 480px
  Tablet: max-width: 768px
  Laptop: max-width: 1024px
  Escritorio: min-width: 1025px
Regla de oro: diseñar primero para móvil (Mobile First) y luego escalar.
El meta viewport en HTML es obligatorio: <meta name="viewport" content="width=device-width, initial-scale=1.0">

Ejemplo de referencia:

```
/* === MOBILE FIRST === */

/* Estilos base: móvil (todo en columna) */
.contenedor {
  display: flex;
  flex-direction: column;
  gap: 16px;
  padding: 16px;
}

nav ul { flex-direction: column; }

.galeria {
  grid-template-columns: 1fr;   /* 1 columna en móvil */
}

/* Tablet: 768px en adelante */
@media (min-width: 768px) {
  .contenedor { flex-direction: row; }
  .galeria { grid-template-columns: repeat(2, 1fr); }
  nav ul { flex-direction: row; }
}

/* Escritorio: 1024px en adelante */
@media (min-width: 1024px) {
  .galeria { grid-template-columns: repeat(3, 1fr); }
  .sidebar { display: block; }   /* Mostrar sidebar en escritorio */
}

/* Menú hamburguesa (CSS puro) */
.menu-toggle { display: none; }      /* Ocultar checkbox */
.hamburguesa { display: none; }      /* Ocultar ícono en escritorio */

@media (max-width: 767px) {
  .hamburguesa { display: block; cursor: pointer; font-size: 1.8rem; }
  nav ul { display: none; flex-direction: column; }
  .menu-toggle:checked ~ nav ul { display: flex; }
}
```

Actividad — Carpeta: ejercicio9/

Adapta la página del Ejercicio 5 (menú horizontal) para que sea completamente responsiva:

1. Enfoque Mobile First: el CSS base debe ser para móviles (menú vertical, 1 columna).
2. En móvil: implementar un menú hamburguesa funcional usando solo CSS (checkbox hack o :focus-within).
3. Breakpoint tablet (768px): menú horizontal compacto, galería de 2 columnas.
4. Breakpoint escritorio (1024px+): menú completo con submenús, galería de 3 columnas, sidebar visible.
5. El header debe tener el logo a la izquierda y el menú hamburguesa a la derecha en móvil.
6. Las tipografías y espaciados también deben escalar (usa clamp() o diferentes tamaños por breakpoint).
7. Usa las DevTools del navegador para probar en diferentes tamaños. Toma capturas de pantalla de 3 tamaños distintos.

Ej. 10	Proyecto Integrador: Sitio Web Estilizado	5 horas	Avanzado

Descripcion del proyecto final
En este proyecto integrador aplicarás TODOS los conocimientos del Taller 2 para estilizar completamente el sitio web de 4 páginas que creaste en el Taller 1.
Este es tu entregable principal y la evidencia de producto del resultado de aprendizaje: "Crear componentes front-end del software de acuerdo con las necesidades del cliente."

Requisitos obligatorios del proyecto:

Requisito	Detalle
Variables CSS globales	Un archivo variables.css con paleta de colores, tipografías y espaciados del sitio.
Hoja de estilos compartida	Un archivo base.css con reset, body, tipografía y estilos del header/footer aplicado a las 4 páginas.
Menú de navegación	Horizontal con submenús, sticky, efecto hover, ítem activo diferenciado. Responsivo en móvil (hamburguesa).
Galería de imágenes	Mínimo 9 imágenes en CSS Grid con hover overlay y efecto de zoom. En la página de servicios o portafolio.
Formulario de contacto	Completamente estilizado con validación visual :valid/:invalid, transiciones focus y botón animado.
Flexbox + Grid	Uso demostrable de ambos sistemas de layout en al menos 2 secciones diferentes.
Animaciones CSS	Mínimo 2 @keyframes: una de entrada y una interactiva (hover o focus).
Diseño responsivo	Funcional en móvil, tablet y escritorio con al menos 2 breakpoints.
Tema de color	Coherencia visual en todas las páginas. Color primario, secundario y de acento definidos como variables.

Estructura de archivos esperada:

```
sitio_css/
├── index.html
├── nosotros.html
├── servicios.html
├── contacto.html
├── css/
│   ├── variables.css    ← Variables globales del tema
│   ├── base.css         ← Reset + estilos compartidos
│   ├── nav.css          ← Menú de navegación
│   ├── galeria.css      ← Galería de imagenes
│   ├── formulario.css   ← Estilos de formularios
│   └── animaciones.css  ← @keyframes y transiciones
└── imagenes/
    └── (todas las imágenes del sitio)
```

Criterios de evaluacion — Proyecto Integrador

Variables CSS y organización de archivos: 15%
Menú de navegación (escritorio + responsivo): 20%
Galería de imágenes con CSS Grid: 15%
Formulario estilizado con validación visual: 20%
Diseño responsivo (móvil / tablet / escritorio): 15%
Animaciones y transiciones: 10%
Coherencia visual y creatividad: 5%

Referencia Rápida — Propiedades CSS Esenciales

Propiedad	Descripción	Ejemplo
`color`	Color del texto	`color: #003087;`
`background`	Color o imagen de fondo	`background: #f5f5f5;`
`font-family`	Fuente tipográfica	`font-family: 'Poppins', sans-serif;`
`font-size`	Tamaño de fuente	`font-size: 1.2rem;`
`font-weight`	Grosor de fuente	`font-weight: 700;`
`text-align`	Alineación de texto	`text-align: center;`
`text-decoration`	Decoración de texto	`text-decoration: none;`
`margin`	Margen exterior	`margin: 16px auto;`
`padding`	Relleno interior	`padding: 12px 24px;`
`border`	Borde del elemento	`border: 2px solid #ccc;`
`border-radius`	Esquinas redondeadas	`border-radius: 8px;`
`box-shadow`	Sombra del elemento	`box-shadow: 0 4px 16px rgba(0,0,0,.1);`
`width / height`	Tamaño del elemento	`width: 300px; height: auto;`
`display: flex`	Activar Flexbox	`display: flex;`
`justify-content`	Alineación eje principal	`justify-content: space-between;`
`align-items`	Alineación eje cruzado	`align-items: center;`
`gap`	Espacio entre hijos flex/grid	`gap: 16px;`
`flex-wrap`	Permitir salto de línea	`flex-wrap: wrap;`
`display: grid`	Activar Grid	`display: grid;`
`grid-template-columns`	Definir columnas del grid	`grid-template-columns: repeat(3, 1fr);`
`grid-column: span N`	Fusionar N columnas	`grid-column: span 2;`
`position`	Tipo de posicionamiento	`position: sticky;`
`top / left`	Posición desde borde	`top: 0; left: 0;`
`z-index`	Capa de apilamiento	`z-index: 100;`
`overflow`	Desbordamiento de contenido	`overflow: hidden;`
`object-fit`	Ajuste de imagen	`object-fit: cover;`
`transition`	Suavizar cambios CSS	`transition: all 0.3s ease;`
`transform`	Transformaciones 2D/3D	`transform: scale(1.05);`
`animation`	Aplicar animación	`animation: girar 1s linear infinite;`
`@keyframes`	Definir animación	`@keyframes nombre { from{} to{} }`
`var()`	Usar variable CSS	`var(--color-primario)`
`:hover`	Pseudoclase hover	`a:hover { color: red; }`
`:focus`	Pseudoclase focus	`input:focus { border-color: blue; }`
`::before / ::after`	Pseudoelementos	`p::before { content: ''; }`
`@media`	Consulta de medios	`@media (max-width: 768px) {}`

Elaborado por:

Elver Siaucho

Instructor — Programación de Software

SENA — Centro de Gestión y Formación CEGAFE