

## **FORMATO DE GUÍAS DE LAS PRÁCTICAS/LABORATORIO Y CENTROS DE SIMULACIÓN**

CARRERA: INGENIERÍA EN SISTEMAS DE INFORMACIÓN

ASIGNATURA: DESARROLLO BASADO EN PLATAFORMAS

NÚMERO DE LA PRÁCTICA: 5

DOCENTE: DAMIÁN NICOLALDE

NIVEL: 3VO.

TEMA DE LA PRÁCTICA: DISEÑO DE UNA PÁGINA WEB ADAPTABLE CON HTML 5 Y CSS3

### **OBJETIVOS DE LA PRÁCTICA**

- Diseñar una página web adaptable con HTML5 y CSS3

### **MARCO TEÓRICO**

#### **DESARROLLO WEB**

- Sitios Web
- Lenguajes
- Herramientas

#### **HTML**

- Estructura
- Contenido
- Formularios
- Diseño web
- Diseño web adaptable
  - Media Queries
  - Puntos de Interrupción
  - Área de visualización
  - Flexibilidad
  - Box-sizing
  - Fijo y flexible
  - Texto, imágenes

#### **CSS**

- Estilos
- Referencias
- Propiedades

### **RECURSOS, MATERIAL Y EQUIPO**

- Computadores
- Visual Studio Code
- Acceso a Internet
- Una cuenta en <https://neocities.org/>

### **ENUNCIADO, INSTRUCCIONES, ACTIVIDADES POR DESARROLLAR Y/O REGISTRO DE DATOS**

Crear un sitio web adaptable exige combinar todas las técnicas que hemos estudiado. Algunas se deben aplicar varias veces y, normalmente, se debe declarar más de una Media Query para

adaptar el diseño a múltiples dispositivos. Pero el sitio en el que establecemos los puntos de interrupción y cómo se implementan estas técnicas depende del diseño de nuestro sitio web y del modelo de caja que implementemos. La estructura básica de este documento incluye una cabecera, una barra de navegación, dos columnas creadas con los elementos `<section>` y `<aside>`, y un pie de página. Lo primero es definir los estilos por defecto que se necesitan para transformar estos elementos en elementos flexibles.

Como hemos mencionado anteriormente, para volver a un elemento flexible en el modelo de caja tradicional, tenemos que definir sus anchos con valores en porcentaje. Los siguientes son los estilos por defecto requeridos. Los tamaños se declaran en porcentaje y usamos la propiedad `max-width` para especificar un ancho máximo para el contenido de la página.

```
* {
  margin: 0px;
  padding: 0px;
}
#cabeceralogo {
  width: 96%;
  height: 150px;
  padding: 0% 2%;
  background-color: #0F76A0;
}
#cabeceralogo > div {
  max-width: 960px;
  margin: 0px auto;
  padding-top: 45px;
}
#cabeceralogo h1 {
  font: bold 54px Arial, sans-serif;
  color: #FFFFFF;
}
#menuprincipal {
  width: 96%;
  height: 50px;
  padding: 0% 2%;
  background-color: #9FC8D9;
  border-top: 1px solid #094660;
  border-bottom: 1px solid #094660;
}
#menuprincipal > div {
  max-width: 960px;
  margin: 0px auto;
}
#menuprincipal li {
  display: inline-block;
  height: 35px;
  padding: 15px 10px 0px 10px;
```

```
margin-right: 5px;
}
#menuprincipal li:hover {
background-color: #6FACC6;
}
#menuprincipal a {
font: bold 18px Arial, sans-serif;
color: #333333;
text-decoration: none;
}
main {
width: 96%;
padding: 2%;
background-image: url("fondo.png");
}
main > div {
max-width: 960px;
margin: 0px auto;
}
#articulosprincipales {
float: left;
width: 65%;
padding-top: 30px;
background-color: #FFFFFF;
border-radius: 10px;
}
#infoadicional {
float: right;
width: 29%;
padding: 2%;
background-color: #E7F1F5;
border-radius: 10px;
}
#infoadicional h1 {
font: bold 18px Arial, sans-serif;
color: #333333;
margin-bottom: 15px;
}
.recuperar {
clear: both;
}
article {
position: relative;
padding: 0px 40px 20px 40px;
}
```

```
article time {
display: block;
position: absolute;
top: -5px;
left: -70px;
width: 80px;
padding: 15px 5px;
background-color: #094660;
box-shadow: 3px 3px 5px rgba(100, 100, 100, 0.7);
border-radius: 5px;
}
.numerodia {
font: bold 36px Verdana, sans-serif;
color: #FFFFFF;
text-align: center;
}
.nombredia {
font: 12px Verdana, sans-serif;
color: #FFFFFF;
text-align: center;
}
article h1 {
margin-bottom: 5px;
font: bold 30px Georgia, sans-serif;
}
article p {
font: 18px Georgia, sans-serif;
}
figure {
margin: 10px 0px;
}
figure img {
max-width: 98%;
padding: 1%;
border: 1px solid;
}
#pielogo {
width: 96%;
padding: 2%;
background-color: #0F76A0;
}
#pielogo > div {
max-width: 960px;
margin: 0px auto;
background-color: #9FC8D9;
```

```
border-radius: 10px;
}
.seccionpie {
float: left;
width: 27.33%;
padding: 3%;
}
.seccionpie h1 {
font: bold 20px Arial, sans-serif;
}
```

El siguiente es el documento HTML.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Este texto es el título del documento</title>
  <meta charset="utf-8">
  <meta name="description" content="Este es un documento HTML5">
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <link rel="stylesheet" href="misestilos.css">
</head>
<body>
  <header id="cabeceralogo">
    <div>
      <h1>Este es el título</h1>
    </div>
  </header>
  <nav id="menuprincipal">
    <div>
      <ul>
        <li><a href="index.html">Principal</a></li>
        <li><a href="fotos.html">Fotos</a></li>
        <li><a href="videos.html">Videos</a></li>
        <li><a href="contacto.html">Contacto</a></li>
      </ul>
    </div>
  </nav>
  <main>
    <div>
      <section id="articulosprincipales">
        <article>
          <h1>Título Primer Artículo</h1>
          <time datetime="2016-12-23" pubdate>
            <div class="numerodia">23</div>
            <div class="nombredia">Viernes</div>
          </time>
        </article>
      </section>
    </div>
  </main>
</body>
</html>
```

```
</time>
<p>Este es el texto de mi primer artículo</p>
<figure>
  
</figure>
</article>
<article>
  <h1>Título Segundo Artículo</h1>
  <time datetime="2016-12-7" pubdate>
    <div class="numerodia">7</div>
    <div class="nombredia">Miércoles</div>
  </time>
  <p>Este es el texto de mi segundo artículo</p>
  <figure>
    
  </figure>
</article>
</section>
<aside id="infoadicional">
  <h1>Información Personal</h1>
  <p>Cita del artículo uno</p>
  <p>Cita del artículo dos</p>
</aside>
<div class="recuperar"></div>
</div>
</main>
<footer id="pielogo">
  <div>
    <section class="seccionpie">
      <h1>Sitio Web</h1>
      <p><a href="index.html">Principal</a></p>
      <p><a href="fotos.html">Fotos</a></p>
      <p><a href="videos.html">Videos</a></p>
    </section>
    <section class="seccionpie">
      <h1>Ayuda</h1>
      <p><a href="contacto.html">Contacto</a></p>
    </section>
    <section class="seccionpie">
      <address>Toronto, Canada</address>
      <small>&copy; Derechos Reservados 2016</small>
    </section>
    <div class="recuperar"></div>
  </div>
</footer>
```

```
</body>  
</html>
```

Mirar en resultados imagen 1 y 2.

Las reglas anteriores (CSS) vuelven flexibles los elementos estructurales, de modo que el sitio web se adapta al espacio disponible, pero cuando la pantalla es demasiado pequeña, el diseño se rompe, algunos elementos se muestran de forma parcial y el contenido se vuelve imposible de leer, tal como muestra la Imagen 2.

Los cambios en el diseño se tienen que introducir gradualmente. Por ejemplo, en nuestro diseño, cuando el tamaño del área de visualización es de 1120 píxeles o inferior, el elemento `<time>` que usamos para representar la fecha en la que el artículo se ha publicado se queda fuera de la ventana. Esto nos indica que nuestro diseño necesita un punto de interrupción a 1120 píxeles para mover este elemento a una posición diferente o reorganizar el contenido. En este caso, decidimos mover la fecha de vuelta dentro del área ocupada por el elemento `<article>`.

```
@media (max-width: 1120px) {  
  article time {  
    position: static;  
    width: 100%;  
    padding: 0px;  
    margin-bottom: 10px;  
    background-color: #FFFFFF;  
    box-shadow: 0px 0px 0px;  
    border-radius: 0px;  
  }  
  .numerodia {  
    display: inline-block;  
    font: bold 14px Verdana, sans-serif;  
    color: #999999;  
    padding-right: 5px;  
  }  
  .nombredia {  
    display: inline-block;  
    font: bold 14px Verdana, sans-serif;  
    color: #999999;  
  }  
  article h1 {  
    margin-bottom: 0px;  
  }  
}
```

Ver imagen 3

Lo primero que tenemos que hacer para reposicionar la fecha es restaurar el modo de posicionamiento del elemento `<time>`. Las reglas por defecto introducidas en la hoja de estilo le otorgan una posición absoluta al elemento para moverlo al lado derecho del área ocupada por el elemento `<article>`, pero cuando la pantalla no es lo suficientemente grande para mostrarlo en esa posición, tenemos que moverlo nuevamente a su ubicación natural en el documento, debajo

del elemento `<h1>`. Esto se logra asignando el valor `static` a la propiedad `position` (`static` es el modo por defecto). Ahora, el elemento `<time>` se coloca debajo del título del artículo, pero aún tenemos que ubicar la fecha y el día en la misma línea. Para este ejemplo, decidimos convertir los elementos en elementos `inline-block`, por lo que se ubicarán uno al lado del otro en la misma línea (ver imagen 3).

Otro momento en el que se debe modificar el diseño es cuando las dos columnas se vuelven demasiado pequeñas para mostrar el contenido apropiadamente. Dependiendo de las características del contenido, podemos optar por ocultarlo, reemplazarlo con un contenido diferente o reorganizar las columnas. En este caso, decidimos convertir el diseño de dos columnas en un diseño de una columna moviendo el elemento `<aside>` debajo del elemento `<section>`.

Existen varias maneras de lograr este objetivo. Por ejemplo, podemos asignar el valor `none` a la propiedad `float` para prevenir que los elementos floten a los lados, o simplemente definir el ancho de los elementos con los valores `100 %` o `auto`, y dejar que el navegador se encargue de ubicarlos uno por línea. Para nuestro ejemplo, decidimos establecer un diseño de una sola columna cuando el área de visualización tiene un ancho de 720 píxeles o menos usando la segunda opción.

```
@media (max-width: 720px) {  
  #articulosprincipales {  
    width: 100%;  
  }  
  #infoadicional {  
    width: 90%;  
    padding: 5%;  
    margin-top: 20px;  
  }  
}
```

Cada vez que el documento se muestre en una pantalla de un tamaño de 720 píxeles o inferior, el usuario verá el contenido organizado en una sola columna (ver imagen 4).

En este momento, el documento se ve bien en ordenadores personales y tablets, pero aún debemos realizar varios cambios para adaptarlo a las pequeñas pantallas de los teléfonos móviles. Cuando esta página se muestre en un área de visualización de 480 píxeles o inferior, las opciones del menú no tendrán espacio suficiente para visualizarse en una sola línea, y el contenido del pie de página puede que no tenga espacio suficiente para mostrarse por completo. Una manera de resolver este problema es listando los ítems uno encima del otro.

```
@media (max-width: 480px) {  
  #cabeceralogo > div {  
    text-align: center;  
  }  
  #cabeceralogo h1 {  
    font: bold 46px Arial, sans-serif;  
  }  
  #menuprincipal {  
    width: 100%;  
    height: 100%;  
  }  
}
```



```
padding: 0%;  
}  
#menuprincipal li {  
display: block;  
margin-right: 0px;  
text-align: center;  
}  
.seccionpie {  
width: 94%;  
text-align: center;  
}  
}
```

Con las reglas anteriores modificamos los elementos para forzar al navegador a mostrarlos uno por línea y centrar sus contenidos. Las primeras dos reglas centran el contenido del elemento cabeceralogo y asignan un nuevo tamaño al título de la página para que se muestre mejor en pantallas pequeñas. A continuación, definimos el tamaño del elemento menuprincipal (el contenedor del menú) para que tenga el máximo ancho posible y una altura determinada por su contenido (height: 100%). También declaramos los elementos <li> dentro del elemento menuprincipal como elementos Block para mostrar las opciones del menú una por línea. Finalmente, los anchos de las tres secciones dentro del pie de página también se extienden para forzar al navegador a mostrarlas una por línea. La imagen 5 ilustra cómo afectan estos cambios a algunos de los elementos.

Después de aplicar estas reglas, el pie de página se ve bien, pero las opciones del menú de la parte superior de la pantalla desplazan el contenido relevante hacia abajo, forzando al usuario a desplazar la página para poder verlo. Una alternativa es reemplazar el menú con un botón y mostrar las opciones solo cuando se pulsa el botón. Para este propósito, tenemos que agregar una imagen al documento que ocupará el lugar del menú cuando el ancho del área de visualización sea de 480 píxeles o inferior (ver imagen 6).

```
<nav id="menuicono">  
    
</nav>
```

La primera modificación que tenemos que introducir en nuestra hoja de estilo es una regla que oculta el elemento menuicono porque solo lo queremos mostrar en pantallas pequeñas. Existen dos formas de hacerlo: definir la propiedad visibility con el valor hidden o declarar el modo de visualización como none con la propiedad display. La primera opción no muestra el elemento al usuario, pero el elemento aún ocupa un espacio en la página, mientras que la segunda le dice al navegador que debe mostrar la página como si el elemento se hubiera incluido en el documento y, por lo tanto, esta última es la opción que tenemos que implementar para nuestro menú.

```
#menuicono {  
display: none;  
width: 95%;  
height: 38px;  
padding: 12px 2% 0px 3%;  
background-color: #9FC8D9;
```

```
border-top: 1px solid #094660;  
border-bottom: 1px solid #094660;  
}
```

El siguiente paso es mostrar el botón y ocultar el menú cuando el ancho del área de visualización es de 480 píxeles o inferior. Las siguientes son las modificaciones que tenemos que introducir en este punto de interrupción.

```
@media (max-width: 480px) {  
  #menuprincipal {  
    display: none;  
    width: 100%;  
    height: 100%;  
    padding: 0%;  
  }  
  #menuprincipal li {  
    display: block;  
    margin-right: 0px;  
    text-align: center;  
  }  
  #menuicono {  
    display: block;  
  }  
  .seccionpie {  
    width: 94%;  
    text-align: center;  
  }  
  #cabeceralogo > div {  
    text-align: center;  
  }  
}
```

Asignando el valor none a la propiedad display del elemento menuprincipal hacemos que el menú desaparezca. Si el ancho del área de visualización es de 480 píxeles o inferior, el elemento menuicono y su contenido se muestran en su lugar (ver imagen 7).

En este momento, tenemos un menú que se adapta al espacio disponible, pero el botón no responde. Para mostrar el menú cuando el usuario pulsa o hace clic en el botón, tenemos que agregar a nuestro documento un programa que responda a esta acción. Estas acciones se llaman eventos y son controladas por código escrito en JavaScript. Estudiaremos JavaScript y eventos en el siguiente capítulo, pero la tarea que debemos realizar aquí es sencilla. Tenemos que volver visible al elemento menuprincipal cuando el usuario pulsa el botón. La siguiente es nuestra implementación para este ejemplo.

```
<script>  
  var visible = false;  
  function iniciar() {  
    var elemento = document.getElementById("menu-img");
```

```
        elemento.addEventListener("click", mostrarMenu);
    }
    function mostrarMenu() {
        var elemento = document.getElementById("menuprincipal");
        if (!visible) {
            elemento.style.display = "block";
            visible = true;
        } else {
            elemento.style.display = "none";
            visible = false;
        }
    }
    window.addEventListener("load", iniciar);
</script>
```

Como veremos más adelante, una forma de insertar código JavaScript dentro de un documento HTML es por medio del elemento `<script>`. Este elemento se ubica normalmente dentro de la cabecera (el elemento `<head>`), pero también se puede ubicar en cualquier otra parte del documento.

El código JavaScript, como cualquier otro código de programación, está compuesto por una serie de instrucciones que se procesan de forma secuencial. El código anterior primero obtiene una referencia al elemento `menu-img` y agrega una función que responderá al evento `click` de este elemento. Luego, cuando el elemento recibe el clic del usuario, el código cambia el valor de la propiedad `display` del elemento `menuprincipal` dependiendo de la condición actual. Si el menú no es visible, lo hace visible, o viceversa (explicaremos cómo funciona este código en el siguiente capítulo). La imagen 8 muestra lo que vemos cuando se pulsa el botón.

## REGISTRO DE CÁLCULOS Y/O RESULTADOS OBTENIDOS



Imagen 1 (cuando el tamaño es grande todos los contenidos se ven correctamente)



Imagen 2 (cuando la pantalla se hace pequeña algunos contenidos de la página web se rompen)



Imagen 3



Imagen 4



Imagen 5



Imagen 6



Imagen 7



Imagen 8

**CUESTIONARIO (responder el cuestionario)**

- ¿Qué son los media queries?
- ¿Qué son los puntos de interrupción?
- ¿Qué es el área de visualización?
- ¿Qué es la flexibilidad?
- ¿Qué son los box-sizing?
- ¿Cómo funciona la adaptabilidad fija y flexible?

## CONCLUSIONES Y RECOMENDACIONES (escribir conclusiones y recomendaciones)

### BIBLIOGRAFÍA

| Bibliografía (usar en normas APA)   | Código Biblioteca   | Nro. ejemplares |
|---|---------------------|-----------------|
| J.D Gauchat (2014), El gran libro de HTML5, CSS3 y JavaScript, 2ra. Edición                             | 006.74/G231g/2014   | 1               |
| J.D Gauchat (2017), El gran libro de HTML5, CSS3 y JavaScript, 3ra. Edición                             | No hay              | 0               |
| Amaro Soriano, José Enrique (2013), El gran libro de programación avanzada con Android                  | 005.43/Am13g        | 1               |
| Nolasco Valenzuela, Jorge Santiago (2016), Desarrollo de aplicaciones móviles con Android, 2da. Edición | 005.4321/N711d/2016 | 1               |
| Montero Miguel, Roberto (2012), Desarrollo de aplicaciones para Android                                 | 005.4321/M764d      | 1               |
|   |                     |                 |

### MATRIZ DE EVALUACIÓN DE CONOCIMIENTOS Y DESTREZAS DEL ESTUDIANTE EN LA PRÁCTICA (RÚBRICA)

| PARÁMETROS  | CRITERIOS DE EVALUACIÓN DE LOS PARÁMETROS  |   |   |  | TOTAL PARCIAL |
|---|--|---|---|--|---------------|
|   |  |   |   |  |               |
| Desempeño en el laboratorio (toma de datos, realización de cálculos, realización de programa, obtención de resultados, obtención de un producto, aplicación de una herramienta, realización de un procedimiento para experimento, etc.) | El estudiante demuestra solvencia al desempeñarse durante la práctica (4 puntos) | El estudiante demuestra parcialmente solvencia al desempeñarse durante la práctica (2 puntos) | El estudiante no demuestra solvencia al desempeñarse durante la práctica (0 puntos) |  |               |
| Cuestionario  | El estudiante contesta el cuestionario de forma correcta (3 puntos)              | El estudiante contesta la mitad del cuestionario de forma correcta (2 puntos)                 | El estudiante contesta el cuestionario de forma incorrecta (1 punto)                |  |               |
| Conclusiones y Recomendaciones  | Todas las conclusiones son adecuadas, objetivas y aplicables (3 puntos)          | Las conclusiones son parcialmente adecuadas, objetivas y aplicables (2 puntos)                | Las conclusiones no son adecuadas, objetivas y aplicables (1 punto)                 |  |               |
| TOTAL   |  |   |   |  |               |

|                                     |                  |                       |  |               |                     |
|-------------------------------------|------------------|-----------------------|--|---------------|---------------------|
| Fecha de elaboración del documento: |                  | 15 de febrero de 2019 |  |               |                     |
| Elaborado por:                      | Damián Nicolalde | Revisado por:         |  | Aprobado por: | Damián Nicolalde    |
| Cargo:                              | Docente          | Cargo:                |  | Cargo:        | Director de Escuela |
| Firma:                              |                  | Firma:                |  | Firma:        |                     |

*Nota: El archivo de las prácticas deberá guardarse con el siguiente formato:*

*Práctica\_NroXX\_abreviacióndeasignatura\_inicialesdeldocente*

*Donde:*

*NroXX: es el número de la práctica de laboratorio.*

*Abreviacióndeasignatura: es el nombre de la asignatura abreviado en máximo 6 caracteres.*

*Inidicialesdeldocente: Es la primera letra del nombre y primera letra del apellido.*