



PÁGINA RESPONSIVE

⚙ Estado	Hecho
📅 Fecha límite	@1 de mayo de 2024
☰ profesor	Rául Rodriguez (Lenguaje de marcas)

TRABAJO GRUPAL DE UNA PÁGINA WEB RESPONSIVE.

Hemos decidido hacer una web sobre una empresa de Ciberseguridad, nos ha parecido una buena idea ya que desde este módulo tenemos una buena visión de las actividades que se realizan y así poder cumplimentar sin problemas todo el contenido de la web

Hemos dividido el proyecto en:

- El Home con un index.html
- Quienes somos / empresa (el enlace va a la misma página las dos)
- Área de actuación
- Noticias
- Publicaciones
- Contacto / localización que decidimos ponerlo junto.

Nos hemos dividido el trabajo así:

Victor Antón realiza la parte de INDEX Y QUIENES SOMOS.

Martín González Varela realiza la parte de ÁREA DE ACTUACIÓN y CONTACTO.

Martín Marco realiza la parte de NOTICIAS y PUBLICACIONES.

Por lo que cada una de las partes te la explicamos cada uno con su PDF.

PÁGINA PRINCIPAL INDEX.HTML

El código html que se ha realizado es la estructura de del contenido de la página principal del proyecto.

En el head de la misma he vinculado el CSS styles.css que dará la forma visual de ver la página web.

Así mismo se han vinculado las fuentes de google fonts para utilizarlas en el diseño.

```
<!-- google fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Inconsolata:wgh
<!-- google font -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Inconsolata:wgh
```

Comenzamos por el body en el que se verá el cuerpo de la página, para después utilizar un div con la clase contenedor que agrupa todo el contenido.

El header define la cabecera y después del mismo se ha creado un navegador que contiene los enlaces a las demás páginas:

```

<body>
  <div class="contenedor">
    <header>
      <nav>
        <ul class="contenido-ul">
          <li class="columna"><a href="#">area de actuación</a>
          <li class="columna"><a href="quienes_somos.html">qui
            <ul class="columna1">
              <li><a href="quienes_somos.html">empresa</li>
              <li><a href="#">Noticias</li></a>
              <li><a href="#"> PUBLICACIONES</li></a>
            </ul>
          </li>
          <li><a href="#">CONTACTO</li></a>
        </ul>
      </nav>
    </header>
  </div>

```

Se describe a continuación lo realizado en el css del navegador:

Comenzamos importando las fuentes de google fonts para usarlas en todo el documento con @import.

Con el selector * damos varios estilos globales con el margin y padding a 0 hacemos que todos los elementos se peguen a la izquierda de la página y con el box-sizing: border-box; hacemos que los elementos se posicionen dentro de la caja y ponemos en mayúsculas toda la página.

Ponemos los h1 y p del documento siendo el margin 0 y padding 10px para evitar saltos cuando se haga el responsive, siendo los "reseteos" explicados.

```

*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  text-transform: uppercase;
}

/* reseteos */

h1, p{

```

```
margin: 0px;  
padding: 10px;  
}
```

La clase contenedor le ponemos un width: 90%; para que deje un margen, como un marco y centre los demás elementos.

El header usa display:flex que sirve para organizar los elementos horizontalmente así como un justify-content: space-between; para que dejen un espacio uniforme entre ellos.

```
.contenedor{  
  font-family: "Teko", sans-serif;  
  margin: 20px auto;  
  width: 90%;  
}  
header{  
  align-items: center;  
  display: flex;  
  justify-content: space-between;  
}
```

Seguimos con el navegador nav el cual se ha realizado un menú desplegable muy sencillo en el que en la parte de Quienes somos es posible desplegarlo y aparecen tres apartados para ir a otras páginas. Es el menú explicado en clase.

```
nav{  
  background-color: rgb(57, 99, 55, 0.8);  
  width: 100%;  
  display: flex;  
  justify-content: right;  
}  
.contenido-ul {  
  display: flex;
```

```

    align-items: center;
    justify-content: space-between;
}

li {
    list-style: none;
    padding: 1.2rem;
    transition: all 0.3s ease 0s;
}

li a {
    text-decoration: none;
    color: whitesmoke;
}

```

Se ha utilizado el Display Flex ya explicado para poner en horizontal los elementos y se han centrado a la derecha de la página.

Dentro de los li se ha utilizado el elemento transitorio: all 0.3s base 0s; para establecer una transición suave cuando se despliegue el menú, así como se eliminan los puntos de la listad con List-style: none;

```

.columna {
    position: flex;
}

.columna1 {
    display: none;
    position: absolute;
    margin-top: 1.5vw;
}

.columna1>li {
    width: 100%;
    transform: translateX(-20px);
    border-bottom: 1px solid #fff;
}

```

```

    text-align: center;
    background-color: #24252a;
}

.columna1>li:hover {
    background-color: rgb(57, 99, 55, 0.8);
}

```

Con las clases columna columna 1 definimos las posiciones del desplegable, al ponerle una `position: absolute;` el cual se coloca en relación con el contenedor padre y la propiedad `transform: translateX(-20px);` se utiliza para mover el desplegable en su posición X.

Llegamos al contenido principal de la página:

```

<div class="contenido-principal">
  <section>
    <header>
      <div class="imagen-position">
        <!-- 
        
      </div>
      <div class="texto">
        <p>ofrecemos soluciones integrales de seguridad para proteger :
      </div>
    </header>
  </section>

  <aside>
    <h1>ciberseguridad <br>all pentenST</h1><br>
  </aside>

```

En el que hemos dividido la misma con un `section` que ocupa el 60% de la página con una imagen y un párrafo donde explica muy brevemente a que nos dedicamos y un `aside` que ocupa el resto donde podemos ver el nombre de la empresa.

la clase `.contenido-principal` se ha realizado con un `display: flex;` y un `flex-wrap: wrap;` el cual te permite controlar cómo se distribuyen los elementos dentro de un contenedor flexible.

```

.contenido-principal{
    display: flex;
    flex-wrap: wrap;
}

.contenido-principal section{
    width: 60%;
    height: 100%;
}

.contenido-principal section header{
    display: flex;
    flex-wrap: wrap;
    width: 60%;
    height: 45rem;
}

```

Luego diseñamos el contenedor section con un ancho del 60%, que a su vez contiene la cabecera.

```

.contenido-principal section header .imagen-position{
    position: relative;
    display: flex;
    height: 50%;
    width: 100%;
}

.contenido-principal section header .texto{
    width: 100%;
    align-items: center;
    /* padding: 10px; */
    text-align: center;
    font-size: 2vw;
}

```

```
font-family: "Teko", sans-serif;
font-weight: 5vw;
color: whitesmoke;
}
```

con `.imagen-position` y `.texto` posicionamos la fotografía con un `position: relative;` y `display: flex;` para así colocar donde más me ha gustado la imagen. Llegamos al `aside` el cual lo hemos puesto con el ancho restante de la página un 40%.

```
aside {
  width: 40%;
}
aside h1 {
  position: relative;
  display: flex;
  right: 10rem;
  top: 12rem;
  font-size: 4.5rem;
  font-weight: 2rem;
  font-family: "Teko", sans-serif;
  color: whitesmoke;
  text-align: center;
  font-style: italic;
}

aside h1:hover {
  margin: 60px;
  color: rgb(57, 99, 55);
  transition: 3s;
}
```

El título un `h1` de la página se ha puesto con una posición relativa para poder ponerlo donde nos gustaba, siendo que le hemos dado un `hover` del color verde del navegador y una transición de 3 segundos.

finalmente el footer para terminar la página.

@MEDIA SCREEN

```
@media screen and (max-width: 767px){
  nav{
    width: 100%;
    margin: 0 auto;
  }

  li{
    margin-right: 0;
    margin-bottom: 15px;
    width: 100%;
    text-align: center;
  }
  .columna{
    display: block;
  }
  .columna1{
    display: none;
  }

  .contenido-principal .imagen-position{
    width: 25%;
  }
  aside h1{
    position: absolute;
    display: block;
    right: 200px;
    font-size: 30px;
    width: 40%;
    height: 10%;
    top: 520px;
  }
  footer{
    position: absolute;
    display: inline-block;
```

```

        width: 55%;
        left: 200px;
    }
}

```

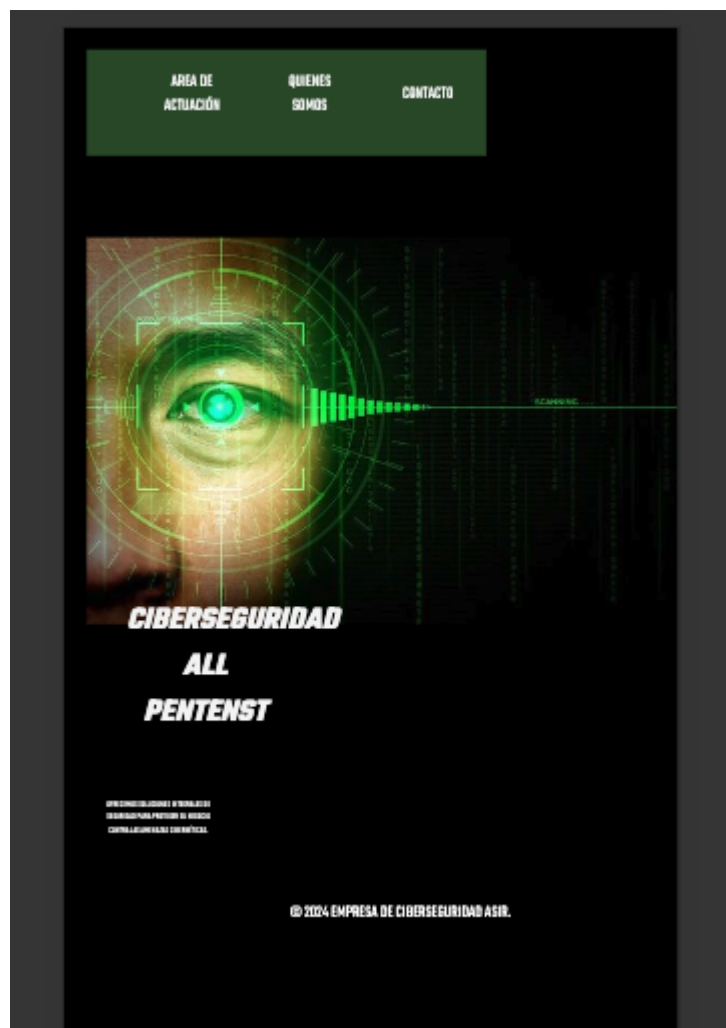
Esta @media es para pantallas de dispositivos móviles en vertical on un ancho de pantalla igual o menor de 767px.

Se han aplicado los siguientes estilos, en el nav dejamos un ancho de 100% y el argón lo dejamos 0 auto

En los elementos de navegación li centra el texto, elimina el margen derecho, y hace un espaciado inferior.

El que he movido de sitio es el aside para que cambie de posición en el teléfono y el footer para que aparezca centrado en la página.

IPHONE.



```

@media screen and (min-width: 768px) and (max-width: 1024px)
  nav{
    width: 100%;
    /* ordena los elemento en fila */
    flex-direction: row;
    /* permite que los objetos salten */
    flex-wrap: wrap;
  }
  .contenido-ul{
    /* alinea los elementos en vertical */
    align-items:center;
  }
  li{
    margin-right: 20;
    margin-bottom: 15px;
    width: 50%;
    text-align: center;
  }
  .columna1{
    display: none;
  }

  .contenido-principal section{
    width: 100%;
  }
  .contenido-principal .imagen-position{
    width: 45%;
  }
  aside h1{
    position: absolute;
    display: block;
    font-size: 40px;
    width: 20%;
    height: 10%;
    right: 200px;
  }

```

}

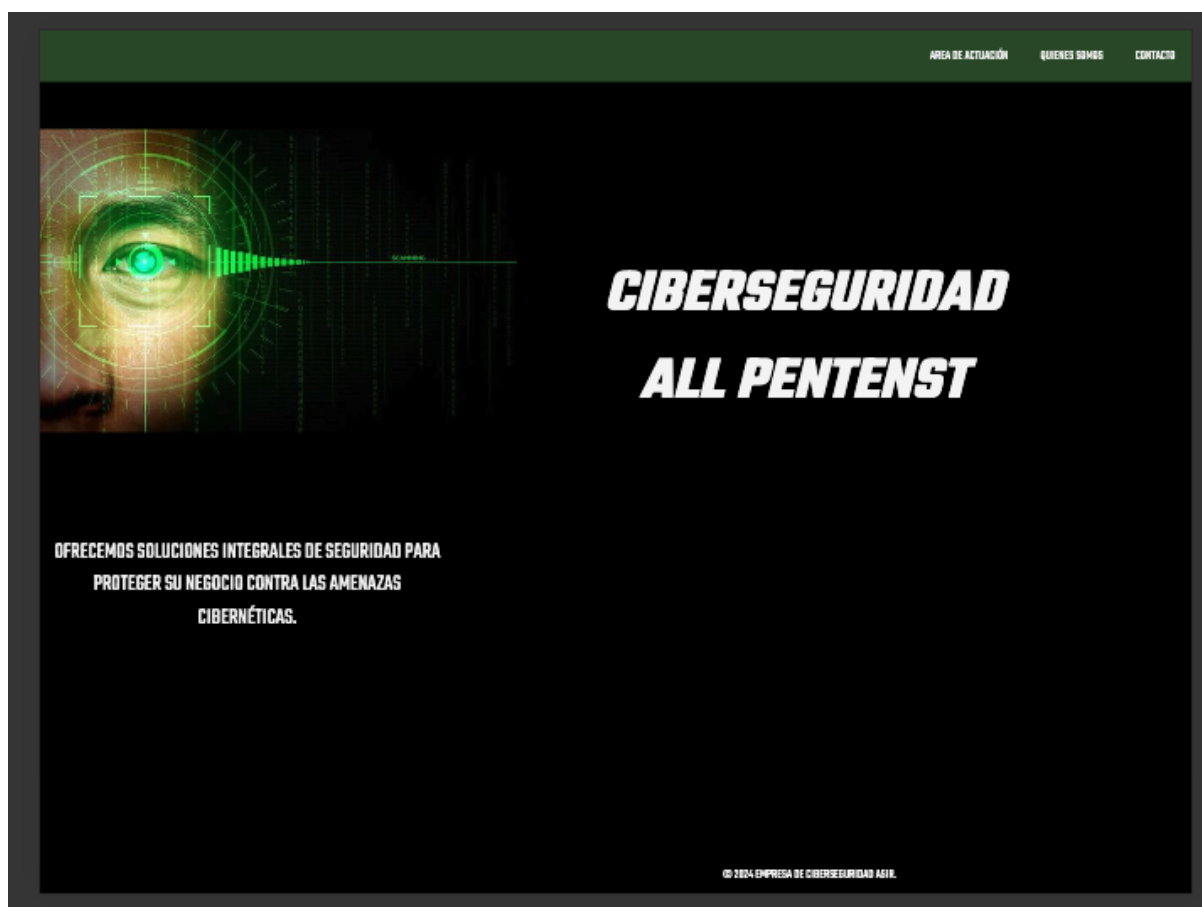
}

Aplica los siguientes estilos solo cuando el tamaño de la pantalla está **entre 768 píxeles y 1024 píxeles de ancho**. Como teléfonos en horizontal y tablets.

Aquí en el nav le he puesto la propiedad `flex-direction: row`; que organiza los elementos en una fila horizontal y el `flex-wrap: wrap`; para que los elementos de navegación se distribuyan en varias líneas si no caben todos en una sola.

Por lo demás como el anterior he ido tocando cada parte que descuadraba y centrando en los distintos tipos de pantallas:

IPAD



```

@media screen and (min-width: 1300px) {
    .contenedor{
        width: 100%;
        margin: 0 auto;
    }
    footer{
        position: relative;
        display: flex;
        width: 50%;
        top: 200px;
    }
}

```

En la última pantalla de más de 1300px he dejado el ancho al 100% y el margen 0 auto, mientras que los demás elementos no los he tenido que tocar ya que se adaptaba perfecto a una gran pantalla, el footer si lo he tenido que posicionar para que se viera.

QUIENES SOMOS

Aquí he realizado una página bastante simple con un nav que contiene un enlace a la página de inicio y un section donde he colocado un div con la clase container que engloba 5 fotografías con su texto indicando lo que ha realizado nuestra empresa, como los más de 100 clientes satisfechos (Mi hija quería poner 1000), me encantaría tener esta empresa la verdad!!

```

<body>
  <div class="contenedor">
    <nav class="navbar">
      <a href="index.html">Inicio</a>
    </nav><br>
    <section class="fotos">
      <div class="container">
        <div><span>Somos <br> Seguridad.</span></div>
        <div><span>+ 100 <br> clientes <br> satisfechos.</span></div>
        <div><span>2 Filiales en España.</span></div>
        <div><span>Somos un equipo multidisciplinar.</span></div>
        <div><span>+ de 30 Países.</span></div>
      </div>
    </section>
    <footer>
      <p>&copy; 2024 Empresa de ciberseguridad ASIR.</p>
    </footer>
  </div>

```

En el css no hay mucho que reseñar simplemente he colocado una imagen de fondo la cual la he puesto Fixed para que sea estática mientras todo lo demás se mueve.

y un height de 100vh para que ocupe toda la altura del viewport y tenga un diseño responsivo.

En el section en el div class="container" he añadido varias cosas para que las fotos queden equitativas y correlacionadas:

```

.container {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-around;
}

.container div {
  width: 45%;
  margin-bottom: 20px;
  text-align: center;
}

.container div img {
  width: 100%;
  height: 80%;
}

.container div span {
  color: whitesmoke;
}

```

```

    font-size: 30px;
    padding: 10px;
    display: block;
}
.container div:hover span {
    background-color: rgb(50, 53, 58);
    transition: 1s;
    transform: scale(1.1);
}

```

El display:Flex; hace que las fotos se muestren en línea, y con el flex-wrap: wrap; hace que si no caben todas se baje a la siguiente línea, con el justify-content:space-around; haya un espacio entre ellas.

En .container div el ancho es del 45% así me entran dos fotos por ancho, dependiendo de las pantallas

En el .container div:over span hay una propiedad que me ha gustado mucho que es transform: scale (1.1); para cuando pasas el ratón por encima del texto del span como que se hace un poco más grande y se lee mejor.

@MEDIA SCREEN

```

@media screen and (max-width: 767px) {
    .container div {
        width: 90%;
    }
}

@media screen and (min-width: 768px) and (max-width: 1024px)
    .container div{
        width: 100%;
    }
}

```

```
@media screen and (min-width: 1300px) {
    .contenedor {
        width: 100%;
        margin: 0 auto;
    }
}
```

Los medias screen en esta parte lo he asociado todo al .contenedor ya que me había salido una página bastante responsive, pongo algunos ejemplos:

Teléfono:



IPAD horizontal

INICIO



SOMOS
SEGURIDAD.



+ 100
CLIENTES
SATISFECHOS.



Estoy satisfecho con el trabajo realizado, he encontrado 100 maneras de no hacer una página web, por lo que he aprendido una que más o menos he podido presentar, un saludo.