

# Diseño de Interfaces We



**Creación de páginas web HTML5+CSS3.**

**Proyecto**

# Diseño de Interfaces We

## ÍNDICE

**Carrousel y Lightbox..... 3**

**Pestañas..... 7**

**Acordeon..... 9**

**Pagina y Imagenes Responsive.....12**

# Diseño de Interfaces Web

## Localización de los puntos de la Rubrica

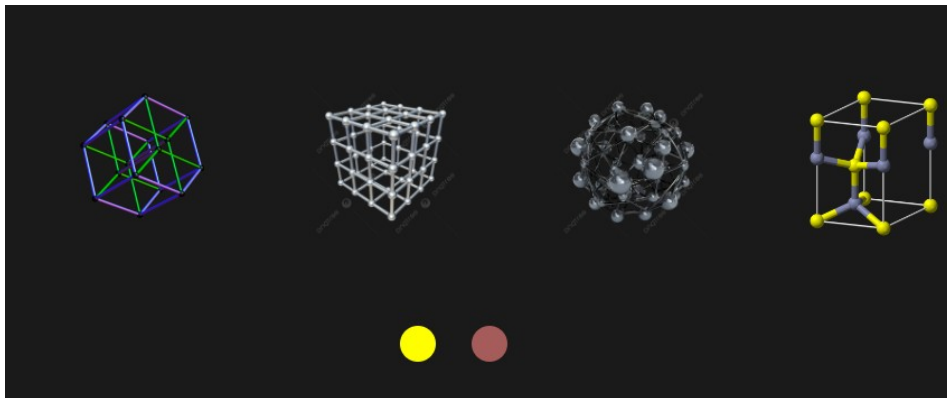
Todos los elementos pedidos por la actividad estarán dentro de Tutoriales.html.

## Carrousel y Lightbox

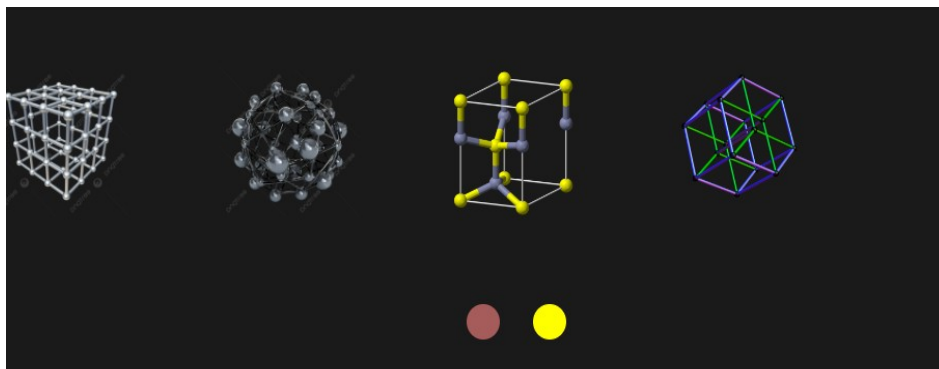
En mi proyecto he decidido usar ambos elementos al mismo tiempo.

La finalidad de esta es mostrar mas elementos y la posibilidad de ver mas información de este al hacer click en el.

## Aspecto del elemento



Al hacer click en el botón izquierdo el cual esta elegido por defecto las imágenes se centraran.



Si se pulsa el derecho el carrusel se desplazara a la derecha mostrando mas objetos si los hay

# Diseño de Interfaces We

Si se hace click en alguno de los elementos se expandirá como un Lightbox



## Codigo

### html

```
<section class="corrousellLightbox">    You, ayer • tarea
  <ul class="">
    <li class=""><a href="#" class="" title="Imagen 1"></a></li>
    <li class=""><a href="#" class="" title="Imagen 2"></a></li>
    <li class=""><a href="#" class="" title="Imagen 3"></a></li>
    <li class=""><a href="#" class="" title="Imagen 4"></a></li>
    <li class=""><a href="#" class="" title="Imagen 1"></a></li>
  </ul>
  <ul class="">
    <li class=""></li>
    <li class=""></li>
  </ul>

  <div class="">
    
    <button class="">Cerrar</button>
  </div>
</section>
```

# Diseño de Interfaces We

## CSS

```
/* LightBox */

.corrouselllightbox{
  background-color: rgba(0, 0, 0, .2);
  border: 1px solid rgba(227, 119, 119, 0.7);
  padding: 2em;
}

.ul {
  width: 80%;
  display: grid;
  grid-template-columns: repeat(8, 1fr);
  gap: 1em;
  margin: auto;
  margin-top: 1em;
  list-style: none;
}

.ul .li {
  padding: 2em;
  align-items: center;
  text-align: center;
}

.ul .li:hover .img {
  transform: scale(1);
}

.ul .img {
  height: 15vh;
  width: calc(100% / 4 - 2em);
  transition: all 1s ease;
  transform: translateX(0%);
}

.lightbox {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, .5);
  z-index: 2;
  display: none;
  margin: auto;
  padding: auto;
}

.lightbox .cerrar {
  width: 5em;
  height: 3em;
  background: rgba(227, 119, 119, 0.7);
  margin: 0 0 1em;
  color: white;
}

.lightbox .grande {
  width: 50%;
  height: 80%;
  transition: all .5s ease;
  margin: auto;
  padding: auto;
}

.lightbox.activo {
  display: flex;
  flex-direction: column;
  margin: auto;
  align-items: center;
  justify-content: flex-start;
}

/* Corrousel */

.puntos {
  width: 100%;
  padding: 1em;
  display: flex;
  flex-flow: row nowrap;
  justify-content: center;
  align-items: center;
  padding: .4em;
  margin: 1em 0;
  list-style: none;
}

.punto {
  width: 2em;
  height: 2em;
  background-color: rgba(227, 119, 119, 0.7);
  margin: 1em;
  border-radius: 50%;
}

.punto.activo {
  background-color: yellow;
}
```

# Diseño de Interfaces We

## JavaScript

El cual le dará la funcionalidad a este al añadir o quitar la clase activo al CSS

```
/* LightBox */
const enlaces = document.querySelectorAll('.ul .a')
const lightbox = document.querySelector('.lightbox')
const grande = document.querySelector('.grande')
const cerrar = document.querySelector('.cerrar')

enlaces.forEach(( cadaEnlace , i ) =>{
  enlaces[i].addEventListener('click', (e)=>{
    e.preventDefault()
    let ruta = cadaEnlace.querySelector('.img').src

    lightbox.classList.add('activo')
    grande.setAttribute('src', ruta)
  })
})

cerrar.addEventListener('click', ()=>{
  lightbox.classList.remove('activo')
  grande.setAttribute('src', '')
})

/* Carousel */
const puntos = document.querySelectorAll('.punto');
const imagenes = document.querySelector('.ul');

puntos.forEach((cadaPunto, i) => {
  puntos[i].addEventListener('click', () => {
    let posicion = i;
    let operacion = posicion * -50;

    imagenes.style.transform = `translateX(${operacion}%)`;

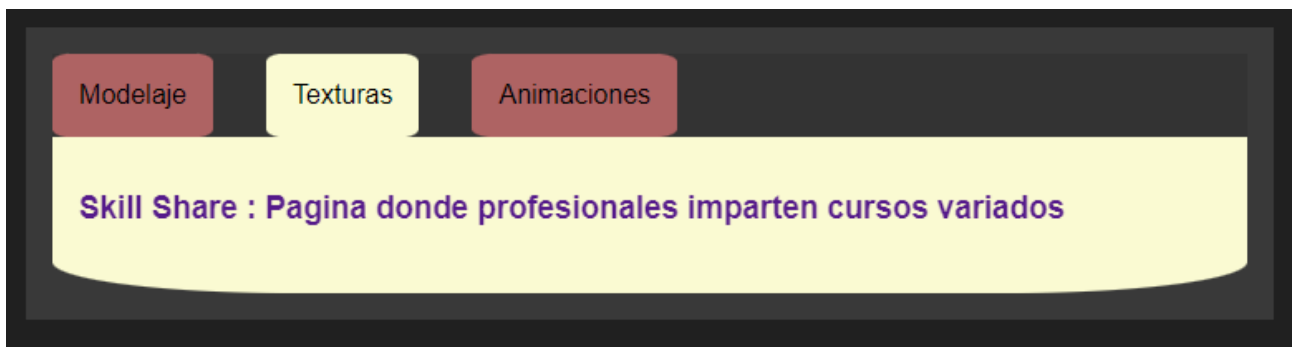
    puntos.forEach((cadaPunto, j) => {
      puntos[j].classList.remove('activo');
    });
    puntos[i].classList.add('activo');
  });
});
```

# Diseño de Interfaces We

## Pestañas (Tab)

En las pestañas he añadido un par de enlaces de paginas que se pueda aprender como modelar

### Aspecto del elemento



### Codigo

#### html

```
<section class="tab"> You, ayer + tarea
  <ul class="">
    <li class="">Modelaje</li>
    <li class="">Texturas</li>
    <li class="">Animaciones</li>
  </ul>

  <div class="">
    <div class="">
      <a href="https://www.udemy.com/es/topic/3d-modeling/free/">
        <h3>Udemy : Pagina de Cursos Relacionados con el modelaje 3D</h3>
      </a>
    </div>
    <div class="">
      <a href="https://www.skillshare.com/es/browse/3d-texturing">
        <h3>Skill Share : Pagina donde profesionales imparten cursos variados</h3>
      </a>
    </div>
    <div class="">
      <a href="https://cei.es/cursos-video-vfx-motion-graphics/?gad_source=1&gclid=EAIaIQobChMI0094tL2gmVFD4GAB3VAAI6EAAAYASAAEGLWH_D_BwE">
        <h3>CEI : Centro de Inovacion y diseño, dirigido a aprender sobre el campo digital</h3>
      </a>
    </div>
  </div>
</section>
```

# Diseño de Interfaces We

## CSS

```
/* Tabs */
.tabs{
  width: 90%;
  max-width: 120em;
  padding: 1em;
  margin: auto;
  margin-top: 2em;
  background-color: rgba(133, 133, 133, 0.25);
}

.tabs .ulT{
  width: 100%;
  background-color: rgba(0, 0, 0, .1);
  display: flex;
  justify-content: flex-start;
  align-items: center;
  list-style: none;
  margin: auto;
  border: 10%;
}

.tabs .liT{
  background-color: rgba(227, 119, 119, 0.7);
  padding: 1em;
  margin: 0 2em 0 0;
  transition: all .5 ease;
  border-radius: 10%;
}

.tabs .liT.activo{
  background-color: lightgoldenrodyellow;
  color: black;
}

.tabs .subcontenedor{
  background-color: lightgoldenrodyellow;
  position: relative;
  min-height: 10vh;
  border-bottom-left-radius: 20%;
  border-bottom-right-radius: 20%;
}

.tabs .bloqueT{
  margin: 1em;
  position: absolute;
  top: 1em;
  opacity: 0;
  transition: all .5s ease;
}

.tabs .bloqueT.activo{
  opacity: 1;
}
```

## Javascript

```
/* Tabs */

const liT = document.querySelectorAll('.liT')
const bloqueT = document.querySelectorAll('.bloqueT')

liT.forEach((cadaLiT, i) => {
  liT[i].addEventListener('click', () => {
    liT.forEach((cadaLiT, i) => {
      liT[i].classList.remove('activo')
      bloqueT[i].classList.remove('activo')
    })

    liT[i].classList.add('activo')
    bloqueT[i].classList.add('activo')
  })
})
```



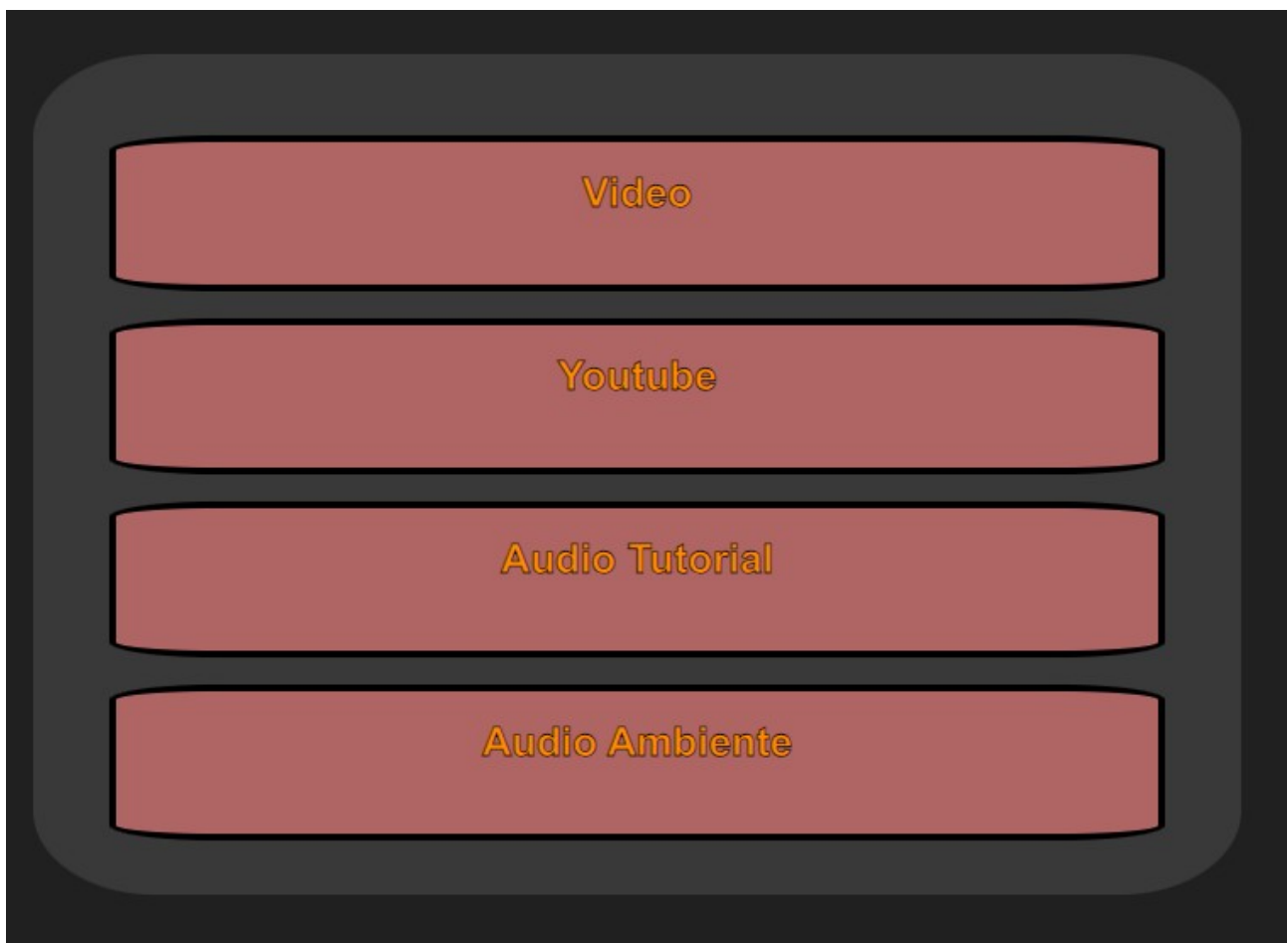
# Diseño de Interfaces We

## Acordeón

Para el acordeón lo que he echo es aprovechar el contenido original de la pagina y ordenarlo con este.

Así el usuario no tendrá una sobrecarga de información al ver tanto contenido multimedia

### Aspecto del elemento



# Diseño de Interfaces We

## Codigo

## html

```
<section class="Tutoriales">      You, el mes pasado • Añadido multimedia
  <!-- Etiqueta Video -->
  <div class="***">
    <div class="***">
      <h1 class="***">Video</h1>
      <div class="***">
        <video class="***" poster="./img/poster.jpg" controls preload="none" width="100%">
          <source src="./multimedia/video.mp4" type="video/mp4">
        </video>
      </div>
    </div>
  </div>
  <!-- Videos Incrustado-->
  <div class="***">
    <h1 class="***">Youtube</h1>
    <div class="***">
      <iframe class="***" src="https://www.youtube.com/embed/0IMb9E00d9I?si=-CdSEhs0uyuzuDwS"
        title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write;
        encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>
      <iframe class="***" " src="https://www.youtube.com/embed/O-tV7uBf5LI?si=hPEYEWfCeWvkRo9q"
        title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write;
        encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>
    </div>
  </div>
  <!-- Audio -->
  <div class="***">
    <h1 class="***">Audio Tutorial</h1>
    <div class="***">
      <audio class="***" controls preload="none">
        <source src="./multimedia/audio.mp3" type="audio/mp3">
      </audio>
    </div>
  </div>
  <div class="***">
    <h1 class="***">Audio Ambiente</h1>
    <div class="***">
      <audio class="***" controls loop volume="0.2" src="./multimedia/audioAmbiental.mp3" type="audio/mp3"></audio>
    </div>
  </div>
</div>
</section>
```

# Diseño de Interfaces We

## CSS

```
/* Acordeon */

.acordeon {
  background-color: rgba(133, 133, 133, 0.25);
  width: 90%;
  max-width: 800px;
  padding: 2em;
  margin: auto;
  border-radius: 10%;
  display: flexbox;
}

.acordeon .bloque{
  background-color: rgba(227, 119, 119, 0.7);
  margin: auto;
  margin-top: 1em;
  padding: 1em;
  width: 90%;
  border-radius: 10%;
  border: 4px solid #000;
}

.acordeon .tituloAco{
  margin: 0 0 1em;
  color: darkorange;
  -webkit-text-stroke-width: 0.8px;
  -webkit-text-stroke-color: black;
}

.acordeon .contenido{
  height: 0;
  padding: 0em;
  overflow: hidden;
  transition: all 0s;
}

.acordeon .bloque.activo .contenido{
  width: auto;
  max-width: 700px;
  height: auto;
  margin: auto;
  padding: auto;
  display: inline-flexbox;
}
```

## Javascript

```
/* Acordeon */

const bloque = document.querySelectorAll(".bloque");
const titulo = document.querySelectorAll(".tituloAco");

titulo.forEach((cadaTitulo, i) => {
  titulo[i].addEventListener("click", () => {
    bloque.forEach((cadaBloque, i) => {
      bloque[i].classList.remove("activo");
    });

    bloque[i].classList.add("activo");
  });
});
```

# Diseño de Interfaces We

## Pagina e Imagenes Responsive

### Pagina

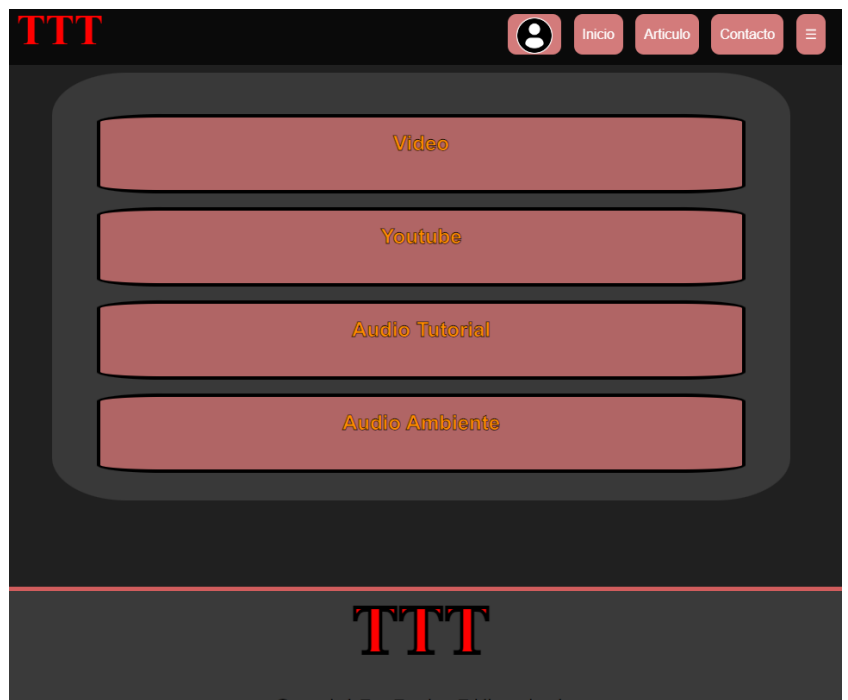
Uno de los puntos positivos de haber usado el flex desde el principio de la maquetación, es que toda la pagina es responsive en si, su tamaño cambia en base a los pixeles que tiene

### Ejemplos:

#### Pequeña

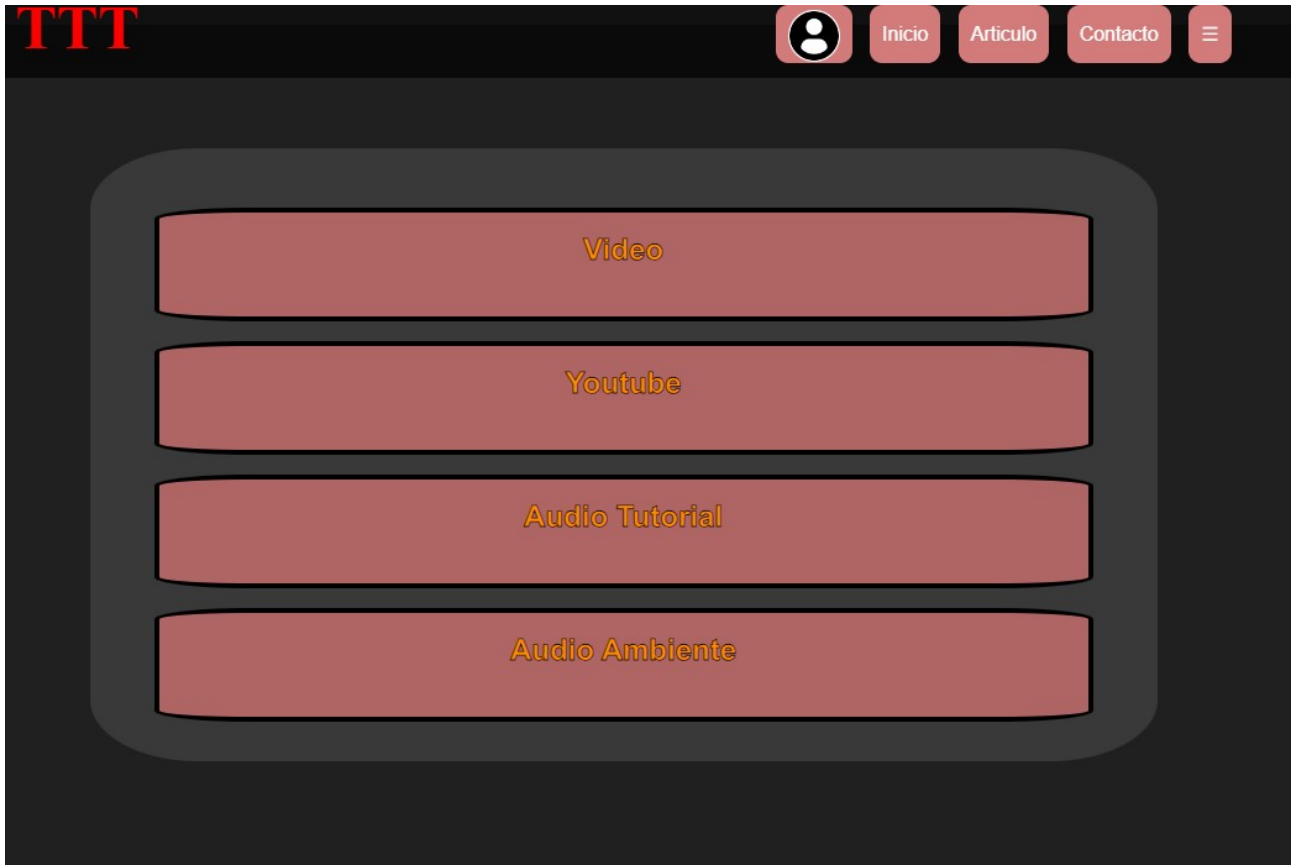


#### Mediana



# Diseño de Interfaces We

Grande



# Diseño de Interfaces We

## Imagen Responsive

Para la imagen responsive , cree una imagen sencilla y la converti a varios formatos distintos.

Tamaño pequeño → webp [ ligera y de poco peso pero calidad baja ]

Tamaño medio → jpg [ formato conocido por su calidad ]

Tamaño grande → jpg.2000 [ De alta calidad pero de mayor peso que el jpg ]

La decisión de esos formatos es en base del tipo de usuario que tendrá esas pantallas

Las mas pequeñas serán de móviles, los cuales prefieren cargas rápidas.

Las medias pueden ir de Tablets a ordenadores.

Y las grandes están pensadas para sobremesa que pueden mostrar mas calidad de imagen y que las carga de material no debería suponer un problema para estos usuarios

## Aspecto del elemento



# Diseño de Interfaces We

## Codigo

### html

```
<section class="imgAdaptable"> You, la semana pasada • 16 de Enero, Interfaces
  <picture>
    <source media="(max-width:699px)" type="image/webp" srcset="./img/tutoriales.webp"/>
    <source media="(max-width:1023px)" type="image/jpg" srcset="./img/tutoriales.jpg"/>
    <source type="image/jpg2" srcset="./img/tutoriales.jpg2"/>
    
  </picture>
</section>
```

### CSS

```
/* Imagenes Adaptables */

.imgAdaptable {
  display: flex;
  align-items: center;
  text-align: center;
  margin: auto;
  padding-top: 5em;
  max-width: 100%;
  height: auto;
  object-fit: contain;
}

.imgAdaptable picture {
  flex: 1;
  margin: auto;
}

.imgAdaptable picture source {
  width: 90%;
  object-fit: contain;
  margin: auto;
}

.imgAdaptable picture img {
  width: 70%;
  max-width: 1024px;
  object-fit: contain;
  margin: auto;
  border-radius: 10%;
  border: 6px solid #c0392b;
}
```