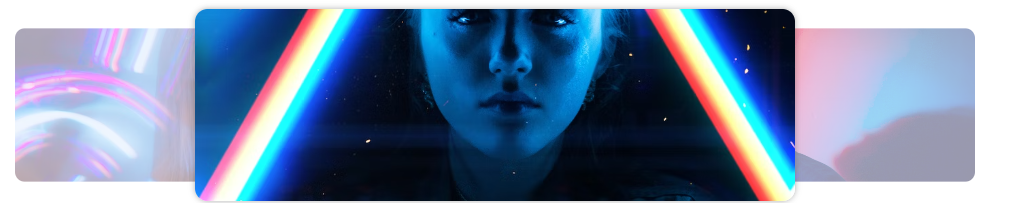
# Plantillas de css

## Carrusel de imágenes:

### Resultado:



### HTML:

<div class="container">

<input type="radio" name="slider" id="item-1" checked>

<input type="radio" name="slider" id="item-2">

<input type="radio" name="slider" id="item-3">

<div class="cards">

<label class="card" for="item-1" id="song-1">

<img src="https://images.unsplash.com/photo-1530651788726-1dbf58eeef1f?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=882&q=80" alt="song">

</label>

<label class="card" for="item-2" id="song-2">

<img src="https://images.unsplash.com/photo-1559386484-97dfc0e15539?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=1234&q=80" alt="song">

</label>

<label class="card" for="item-3" id="song-3">

<img src="https://images.unsplash.com/photo-1533461502717-83546f485d24?ixlib=rb-1.2.1&auto=format&fit=crop&w=900&q=60" alt="song">

</label>

</div>

### CSS:

@import url("https://fonts.googleapis.com/css?family=DM+Sans:400,500,700&display=swap");

\* {

box-sizing: border-box;

}

html, body {

margin: 0;

padding: 0;

width: 100%;

height: 100%;

}

body {

display: flex;

align-items: center;

justify-content: center;

padding: 30px 10px;

transition: background .4s ease-in;

}

input[type=radio] {

display: none;

}

.card {

position: absolute;

width: 60%;

height: 100%;

left: 0;

right: 0;

margin: auto;

transition: transform .4s ease;

cursor: pointer;

}

.container {

width: 100%;

max-width: 800px;

max-height: 600px;

height: 100%;

transform-style: preserve-3d;

display: flex;

justify-content: center;

flex-direction: column;

align-items: center;

}

.cards {

position: relative;

width: 100%;

height: 100%;

margin-bottom: 20px;

}

img {

width: 100%;

height: 100%;

border-radius: 10px;

object-fit: cover;

}

#item-1:checked ~ .cards #song-3, #item-2:checked ~ .cards #song-1, #item-3:checked ~ .cards #song-2 {

transform: translatex(-40%) scale(.8);

opacity: .4;

z-index: 0;

}

#item-1:checked ~ .cards #song-2, #item-2:checked ~ .cards #song-3, #item-3:checked ~ .cards #song-1 {

transform: translatex(40%) scale(.8);

opacity: .4;

z-index: 0;

}

#item-1:checked ~ .cards #song-1, #item-2:checked ~ .cards #song-2, #item-3:checked ~ .cards #song-3 {

transform: translatex(0) scale(1);

opacity: 1;

z-index: 1;

img {

box-shadow: 0px 0px 5px 0px rgba(81, 81, 81, 0.47);

}

}

#item-2:checked ~ .player #test {

transform: translateY(0);

}

#item-2:checked ~ .player #test {

transform: translateY(-40px);

}

#item-3:checked ~ .player #test {

transform: translateY(-80px);

}

# Galería de imgenes:

## 1-Opcion:

### Resultado:

### HTML

//- PREVIOUS VERSION: https://codepen.io/mican/pen/awxmpY

- var placeholder = function(width, height) { return "data:image/svg+xml;charset=utf-8,%3Csvg%20xmlns%3D'http://www.w3.org/2000/svg'%20viewBox%3D'0%200%20" + width + "%20" + height + "'%20%2F%3E" }

- var images = [{ id: '-gS54SWrHMg', width: 1000, height: 500 }, { id: 'AU1rKyKPJco', width: 500, height: 1000, caption: 'Hello from Poland' }, { id: 'AXfDvKOawZQ' }, { id: 'gKlkZrsG\_Pw', width: 1000, height: 500 }, { id: 'DVONaLgCRxo', width: 500, height: 1000 }, { id: 'o7txpYpxNLs', caption: 'Have a nice day' }, { id: 'ZsgUsl8GATg', width: 500, height: 1000, caption: "This is a very very long description to show you it's possible to add something like this" }, { id: 'CkagyZJ88kE' }, { id: 'PpQ4-HOZ\_8U', width: 1000, height: 500 }, { id: 'si7gjqJQj\_8' }, { id: 'u0M0gyuexfE', width: 500, height: 1000 }, { id: 'aQcE3gDSSTY' }, { id: 'GkCafprWKRo', width: 500, height: 1000 }, { id: 'OFlzoTfpRdw' }, { id: 'YlFM0-LdHu8' }, { id: 'c\_Tc9ZELeYw' }];

mixin gallery-item(id, width=500, height=500, caption)

if height > width

- var klass = 'vertical';

else if width > height

- var klass = 'horizontal';

figure.gallery-item(

itemprop="associatedMedia",

itemscope="",

itemtype="http://schema.org/ImageObject",

class=klass

)

a(

href=`https://source.unsplash.com/${id}/${width \* 2}x${height \* 2}`,

itemprop="contentUrl",

data-size=`${width \* 2}x${height \* 2}`

)

img.lazyload.lazypreload.fadein(

src=placeholder(width, height),

data-src=`https://source.unsplash.com/${id}/${width}x${height}`,

itemprop="thumbnail",

alt="Image description"

)

figcaption.gallery-caption(itemprop="caption description")= caption || 'Caption'

.gallery(itemscope="", itemtype="http://schema.org/ImageGallery")

each image in images

+gallery-item(image.id,image.width,image.height, image.caption)

// Root element of PhotoSwipe. Must have class pswp.

.pswp(tabindex="-1", role="dialog", aria-hidden="true")

//

Background of PhotoSwipe.

It's a separate element as animating opacity is faster than rgba().

.pswp\_\_bg

// Slides wrapper with overflow:hidden.

.pswp\_\_scroll-wrap

//

Container that holds slides.

PhotoSwipe keeps only 3 of them in the DOM to save memory.

Don't modify these 3 pswp\_\_item elements, data is added later on.

.pswp\_\_container

.pswp\_\_item

.pswp\_\_item

.pswp\_\_item

// Default (PhotoSwipeUI\_Default) interface on top of sliding area. Can be changed.

.pswp\_\_ui.pswp\_\_ui--hidden

.pswp\_\_top-bar

// Controls are self-explanatory. Order can be changed.

.pswp\_\_counter

button.pswp\_\_button.pswp\_\_button--close(title="Close (Esc)")

button.pswp\_\_button.pswp\_\_button--share(title="Share")

button.pswp\_\_button.pswp\_\_button--fs(title="Toggle fullscreen")

button.pswp\_\_button.pswp\_\_button--zoom(title="Zoom in/out")

// Preloader demo http://codepen.io/dimsemenov/pen/yyBWoR

// element will get class pswp\_\_preloader--active when preloader is running

.pswp\_\_preloader

.pswp\_\_preloader\_\_icn

.pswp\_\_preloader\_\_cut

.pswp\_\_preloader\_\_donut

.pswp\_\_share-modal.pswp\_\_share-modal--hidden.pswp\_\_single-tap

.pswp\_\_share-tooltip

button.pswp\_\_button.pswp\_\_button--arrow--left(

title="Previous (arrow left)"

)

button.pswp\_\_button.pswp\_\_button--arrow--right(

title="Next (arrow right)"

)

.pswp\_\_caption

.pswp\_\_caption\_\_center

### CSS:

%gallery-caption

+absolute(bottom 4rem left 50%)

transform: translate(-50%,0%)

font-size: 12px

+breakpoint($mobile)

font-size: 14px

color: rgba(white,0)

padding: 1.25em 1.5em

transition: all .2s ease

font-weight: 600

// min-width: 10rem

// width: 50%

// max-width: calc(100% - 6rem)

line-height: 1.25

text-align: center

box-sizing: border-box

pointer-events: none

&:before, &:after

content: ''

+absolute(top right left bottom)

background: rgba(black,1)

width: 100%

height: 100%

transition: all .3s ease 0s

z-index: -1

&:before

top: auto

height: 3px

transform: scale(0,1)

transform-origin: bottom left

transition-delay: .6s

&:after

transform: scale(1,0)

transform-origin: bottom

transition-delay: .3s

&.visible

color: rgba(white,1)

text-shadow: 0 0 1px rgba(black,.2)

transition: all .3s ease .3s

&:before

transform: scale(1,1)

transition-delay: 0s

&:after

transform: scale(1,1)

&:empty

display: none

$corner: 1rem

%image-border

+relative

&:before, &:after

content: ''

+absolute(top right left bottom)

border: 0 solid rgba(black,.1)

transition: all .2s

will-change: border

z-index: 10

&.active

&:before

border-width: .5rem

&:after

border-width: 2px

&:after

margin: $corner

border: 2px solid rgba(white,.5)

clip-path: polygon(0 calc(100% - #{$corner}), 0 100%, $corner 100%, $corner 0, 0 0, 0 $corner, 100% $corner, 100% 0, calc(100% - #{$corner}) 0, calc(100% - #{$corner}) 100%, 100% 100%, 100% calc(100% - #{$corner}))

&:hover

&:after

transform: scale(.9)

border-color: rgba(white,1)

%caption-outside

background-color: black

color: white

padding: .75em 1em

display: inline-block

text-align: left

%gallery-grid

+sans-serif-font

width: 100%

display: grid

grid-template-rows: flow

grid-auto-flow: dense

+breakpoint(max-width $mobile - 1px)

grid-template-columns: repeat(2, 1fr)

+breakpoint($mobile $desktop - 1px)

grid-template-columns: repeat(3, 1fr)

+breakpoint($desktop $laptop - px)

grid-template-columns: repeat(4, 1fr)

+breakpoint($laptop $screen - 1px)

grid-template-columns: repeat(5, 1fr)

+breakpoint($screen)

grid-template-columns: repeat(6, 1fr)

%gallery-item

+relative

background-color: rgba(black,.5)

overflow: hidden

img, a

display: block

&.vertical

grid-row: span 2

&.horizontal

grid-column: span 2

.gallery

@extend %gallery-grid

.gallery-item

@extend %gallery-item

a

display: block

@extend %image-border

.lazy-images &.image-lazyloaded, html:not(.lazy-images) &

@extend %image-border.active

.gallery-caption

html:not(.touchevents) &

@extend %gallery-caption

.gallery-item:hover &

@extend %gallery-caption.visible

[class\*=list] &, .gallery-size-thumbnail &

display: none

### JAVA

initPhotoSwipeFromDOM = (gallerySelector) ->

# parse slide data (url, title, size ...) from DOM elements

# (children of gallerySelector)

parseThumbnailElements = (el) ->

thumbElements = el.childNodes

numNodes = thumbElements.length

items = []

figureEl = undefined

linkEl = undefined

size = undefined

item = undefined

i = 0

while i < numNodes

figureEl = thumbElements[i]

# <figure> element

# include only element nodes

if figureEl.nodeType != 1

i++

continue

linkEl = figureEl.children[0]

# <a> element

size = linkEl.getAttribute('data-size').split('x')

# create slide object

item =

src: linkEl.getAttribute('href')

w: parseInt(size[0], 10)

h: parseInt(size[1], 10)

if figureEl.children.length > 1

# <figcaption> content

item.title = figureEl.children[1].innerHTML

if linkEl.children.length > 0

# <img> thumbnail element, retrieving thumbnail url

item.msrc = linkEl.children[0].getAttribute('src')

item.el = figureEl

# save link to element for getThumbBoundsFn

items.push item

i++

items

# find nearest parent element

closest = (el, fn) ->

el and (if fn(el) then el else closest(el.parentNode, fn))

# triggers when user clicks on thumbnail

onThumbnailsClick = (e) ->

e = e or window.event

if e.preventDefault then e.preventDefault() else (e.returnValue = false)

eTarget = e.target or e.srcElement

# find root element of slide

clickedListItem = closest(eTarget, (el) ->

el.tagName and el.tagName.toUpperCase() == 'FIGURE'

)

if !clickedListItem

return

# find index of clicked item by looping through all child nodes

# alternatively, you may define index via data- attribute

clickedGallery = clickedListItem.parentNode

childNodes = clickedListItem.parentNode.childNodes

numChildNodes = childNodes.length

nodeIndex = 0

index = undefined

i = 0

while i < numChildNodes

if childNodes[i].nodeType != 1

i++

continue

if childNodes[i] == clickedListItem

index = nodeIndex

break

nodeIndex++

i++

if index >= 0

# open PhotoSwipe if valid index found

openPhotoSwipe index, clickedGallery

false

# parse picture index and gallery index from URL (#&pid=1&gid=2)

photoswipeParseHash = ->

hash = window.location.hash.substring(1)

params = {}

if hash.length < 5

return params

vars = hash.split('&')

i = 0

while i < vars.length

if !vars[i]

i++

continue

pair = vars[i].split('=')

if pair.length < 2

i++

continue

params[pair[0]] = pair[1]

i++

if params.gid

params.gid = parseInt(params.gid, 10)

params

openPhotoSwipe = (index, galleryElement, disableAnimation, fromURL) ->

pswpElement = document.querySelectorAll('.pswp')[0]

gallery = undefined

options = undefined

items = undefined

items = parseThumbnailElements(galleryElement)

# define options (if needed)

options =

galleryUID: galleryElement.getAttribute('data-pswp-uid')

getThumbBoundsFn: (index) ->

# See Options -> getThumbBoundsFn section of documentation for more info

thumbnail = items[index].el.getElementsByTagName('img')[0]

pageYScroll = window.pageYOffset or document.documentElement.scrollTop

rect = thumbnail.getBoundingClientRect()

{

x: rect.left

y: rect.top + pageYScroll

w: rect.width

}

# PhotoSwipe opened from URL

if fromURL

if options.galleryPIDs

# parse real index when custom PIDs are used

# http://photoswipe.com/documentation/faq.html#custom-pid-in-url

j = 0

while j < items.length

if items[j].pid == index

options.index = j

break

j++

else

# in URL indexes start from 1

options.index = parseInt(index, 10) - 1

else

options.index = parseInt(index, 10)

# exit if index not found

if isNaN(options.index)

return

if disableAnimation

options.showAnimationDuration = 0

# Pass data to PhotoSwipe and initialize it

gallery = new PhotoSwipe(pswpElement, PhotoSwipeUI\_Default, items, options)

gallery.init()

return

# loop through all gallery elements and bind events

galleryElements = document.querySelectorAll(gallerySelector)

i = 0

l = galleryElements.length

while i < l

galleryElements[i].setAttribute 'data-pswp-uid', i + 1

galleryElements[i].onclick = onThumbnailsClick

i++

# Parse URL and open gallery if it contains #&pid=3&gid=1

hashData = photoswipeParseHash()

if hashData.pid and hashData.gid

openPhotoSwipe hashData.pid, galleryElements[hashData.gid - 1], true, true

return

# execute above function

initPhotoSwipeFromDOM '.gallery'

# ---

# generated by js2coffee 2.2.0

## Opción 2:

# Infografia:

[Código html desde google fotos, sin que tengas que descargar la imagen.](https://wwwhatsnew.com/2020/05/17/como-insertar-imagenes-de-google-fotos-en-tu-sitio-web/)

<https://codepen.io/humbl3man/pen/NWEOLy>

<https://www.youtube.com/watch?v=k3wQL85U9s8&ab_channel=divcode>

<https://codepen.io/mican/pen/RyjZgm?editors=1000>

<https://codepen.io/cycosta/pen/wvMeNoJ>

<https://codepen.io/knyttneve/pen/YgZbLO>

<https://codepen.io/alchatti/pen/NOdzLO>

[Menu para RRSS](https://codepen.io/benhatsor/pen/YzwxaQV)

# Lista de videos:

[Lista de reproducción de videos:](https://codepen.io/coreyroth/pen/eeyvMw)

# Para sacar cositas:

<https://plantillashtmlgratis.com/todas-las-plantillas/plantilla/plantilla-html-gratis-central/>

—---------------------------------------------------------------------------------------

[No tiene nada que ver](http://www.auditiait.es/es/lista-de-consolas-msc/)

[Para sacar pantalla de carga:](https://programacion.net/articulo/mostrar_la_tipica_imagen_de_carga_cuando_cargue_la_pagina_utilizando_jquery_y_css_1728)

[Favicon](https://codigonaranja.com/curso-html-como-agregar-un-favicon-icono#1-como-poner-un-icono-en-html)

[Base](https://developer.mozilla.org/es/docs/Web/HTML/Element/base)

[Citas](https://developer.mozilla.org/es/docs/Web/HTML/Element/cite)