

CSS3: galería con título y efectos con Grid CSS

Autor: Marina Vega Vázquez **Fecha:** 13/01/20223

- [CSS3: galería con título y efectos con Grid CSS](#)
 - [Cuadrículas CSS responsivas, en tres líneas](#)
 - [Efecto hover \(al colocar el puntero del mouse\)](#)
 - [Superposición del título](#)
 - [Estilizamos el título](#)
 - [Oculta el título](#)
 - [Línea desenfocadas](#)
 - [El problema con las pantallas táctiles](#)

Lo normal antes con **CSS 2.1** era crear los elementos o cajas con una altura fija. Pero esto tiene varios problemas: El responsive no vale, ya que los elementos al ir teniendo diferentes anchos, la altura también va cambiando. Y siempre dependeríamos del contenido máximo de una caja para insertar la altura sin que se nos desbordara...

Ahora es cuando entra en escena la propiedad **Grid**, para los que no lo conozcáis es un modelo de caja de **CSS3**. A diferencia de FlexBox, el cuál es un sistema unidimensional, Grid CSS nos permite trabajar al mismo tiempo con ambos elementos (filas y columnas), por lo que crear retículas de maquetación es extremadamente sencillo y rápido.

Vamos a crear un típico ejemplo de diseño de galería de cajas con foto y sobre la foto de donde es.

Para crear la galería creamos una lista desordenada y cada li contendrá otro contenedor con la imagen y el texto a mostrar tras esta.

```
<main class="contenedor-grid">
  <article class="sitio">
    <a class="titulo-sitio"
      href="https://www.visittheusa.mx/experience/san-francisco-california-turismo-compras-y-restaurantes-en-la-bahia-de-la-ciudad"
      target="_blank">San Francisco </a>
    <div class="imagen-sitio">
      <a href="#"> </a>
    </div>
  </article>

  <article class="sitio">
    <a class="titulo-sitio" href="https://www.londres.es/"
target="_blank">Londres</a>
    <div class="imagen-sitio">
      <a href="#"> </a>
    </div>
  </article>

  .....

```

Cuadrículas CSS responsivas, en tres líneas

Con solo unas cuantas reglas podemos convertir nuestras miniaturas en una cuadrícula:

```
.contenedor-grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));  
  grid-gap: 1em;  
}
```

Estas sencillas líneas son:

- **display: grid;** - que convierte a todos los elementos secundarios directos en elementos de la cuadrícula y los establece.
- **grid-gap: 1em;** - que define la separación entre elementos.

La ligera complejidad viene del valor que hemos dado a la propiedad **grid-template-columns**, que define nuestras columnas. Normalmente podrías esperar ver algo como **repeat(3, 200px)** que nos daría tres columnas de 200px. En este caso hemos utilizado la palabra clave **auto-fill** para **repeat()**, y luego algunos valores. **auto-fill** - establece tantas columnas con un mínimo de 300px (indicada con **minmax**) y un máximo de 1fr como quepan en el contenedor de la cuadrícula.

```
.contenedor-grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));  
  grid-gap: 1em;  
}
```

Lo completaremos indicando que como máximo emplee el 90% del viewport, que centre el contenido y añadimos un margen superior.

```
.contenedor-grid {  
  max-width: 90vw;  
  margin: 0 auto;  
  margin-top: 2rem;  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));  
  grid-gap: 1em;  
}
```

Un último detalle que necesitarás añadir:

```
img {  
  width: 100%;  
  height: auto;  
}
```

Esto asegurará que las imágenes llenen sus contenedores de manera adecuada, recuerda que al poner ancho 100% y alto auto, la imagen se ajuste proporcionalmente al contenedor en el que se encuentra.

Efecto hover (al colocar el puntero del mouse)

Vamos a utilizar unos títulos superpuestos, revelándolos al colocar el puntero del ratón. También vamos a dar a las imágenes un efecto rojo, y un ligero desenfoque para ayudar a la legibilidad del texto superpuesto.

Superposición del título

Para superponer el título necesitamos posicionarlo, así que empezaremos haciendo la posición del artículo relativa (position: relative;) y la posición del título absoluta (position: absolute;) . También le daremos un fondo rojo y haremos que llene el espacio disponible:

```
.sitio {  
  position: relative;  
}  
  
.titulo-sitio {  
  height: 100%;  
  width: 100%;  
  top: 0;  
  left: 0;  
  position: absolute;  
  background: rgba(90, 0, 10, 0.4);  
  
  opacity: 0;  
  
  /* estilo tipográfico */  
  color: white;  
  font-size: 1.5rem;  
  font-weight: bold;  
  text-decoration: none;  
  z-index: 1;  
  
  /* posicionamos el texto en el centro */  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

Hay una o dos cosas que mejorar, incluyendo ese espacio extra en la parte inferior (el título es ligeramente más grande que la miniatura). Esto puede resolverse eliminando cualquier line-height en el contenedor de la imagen:

```
.imagen-sitio {  
  line-height: 0;  
}
```

Estilizamos el título

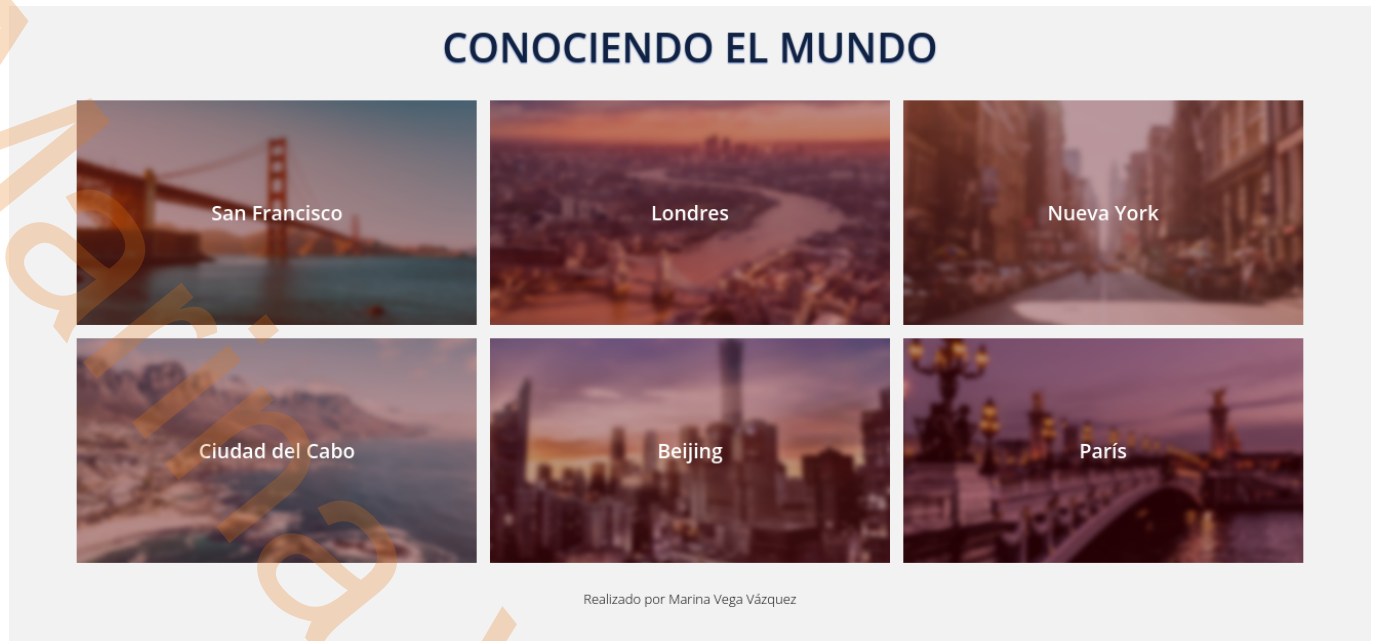
Algunos estilos tipográficos mejorarán la apariencia de nuestro título como son:

- Asignar color blanco al texto.
- Tamaño de fuente 1.5rem.
- Texto en negrita.
- Eliminar la decoración de los enlaces (el título del sitio es un enlace).

Debes realizar tu estos cambios.

```
.titulo-sitio {  
  height: 100%;  
  width: 100%;  
  top: 0;  
  left: 0;  
  position: absolute;  
  background: rgba(90, 0, 10, 0.4);  
  
  /* estilo tipográfico */  
  .....  
  
  /* posicionamos el texto en el centro */  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

Ahora ya podemos ver el siguiente resultado que buscábamos:



Ahora solo nos faltan ajustes para que el título solo aparezca al pasar sobre el div con la clase `.imagen-sitio`.

Ocultar el título

Ahora vamos a ocultar el título eliminando su opacidad para que solo podamos verlo al posicionar el puntero del ratón sobre el div con la clase `.imagen-sitio`. Un par de reglas más debajo de las que ya hemos añadido a nuestro `.titulo-sitio` deberían efectuarlo:

- Hacer el título totalmente transparente con la propiedad `opacity`.
- En este punto también establecemos una regla `transition` para que cuando recuperemos la opacidad ocurra de manera gradual (en el transcurso de 0,5 segundos).

Ahora, para poder hacer visible el título al pasar sobre el div con la clase `.imagen-sitio`, debemos poner la opacidad al máximo.

¡Excelente! Ahora nuestro título aparece de nuevo al colocar el puntero del ratón sobre la imagen de Londres

CONOCIENDO EL MUNDO



Realizado por Marina Vega Vázquez

Línea desenfocadas

Ya hemos creado un efecto de hover con una excelente apariencia, pero llevemos las cosas un paso más allá, ¿de acuerdo? También añadamos un filtro de desenfoque a la imagen. Empezaremos ajustando el filtro de desenfoque a un estado normal, para tener algo desde donde iniciar la transición. Luego desenfocaremos las cosas en 2px para el estado de hover (haz este número tan extremo como desees, pero creo que 2px nos da un buen efecto visual):

```
.imagen-sitio img {  
  filter: blur(0px);  
  transition: filter 0.3s ease-in;  
}  
  
/* Al pasar sobre cada article hacemos que la imagen se desenfoque */  
.sitio:hover .imagen-sitio img {  
  filter: blur(2px);  
}
```

Esto es lo que nos da:



Un par de aspectos a tener en cuenta:

El título ha desaparecido porque el navegador ahora muestra el gráfico desenfocado en la parte superior. El efecto desenfocado se ve bien, pero también nos da un borde borroso (podemos hacerlo mejor). Arreglar el título oculto es tan sencillo como añadir `z-index: 1;` a `.titulo-sitio`.

Arreglar el borde borroso es un poco más complicado. Para empezar, escalamos la imagen para hacerla un poco más grande, luego establecemos `overflow: hidden;` en el contenedor de la imagen (`.sitio`) de manera que cuando la imagen sea más grande se desenfoque sus bordes y se recorten de manera eficaz. Aquí están las propiedades terminadas para los dos elementos en cuestión:

```
.imagen-sitio img{
  filter: blur(0px);
  transition: filter 0.3s ease-in;
  transform: scale(1.1);
}

.sitio {
  position: relative;
  overflow: hidden;
}
```

La regla `transform: scale(1.1);` escala nuestra imagen en todas las direcciones a 1.1 (donde 1.0 mantendría las cosas exactamente del mismo tamaño). Aquí está el resultado mucho más ordenado:

El problema con las pantallas táctiles

Ahora tenemos atractiva cuadrícula de imágenes con un efecto desenfocado al colocar el puntero del ratón en cada miniatura. Lo único es que los títulos están ocultos a cualquiera que no pueda colocar o pasar el puntero del ratón por encima de las imágenes (un buen número de tabletas y teléfonos inteligentes no emulan el estado `hover`), lo cual no es muy accesible.

Por suerte, CSS tiene algunas media queries de interacción muy útiles que nos pueden ayudar (y también gozan de un soporte para navegadores bastante decente). Estas «queries» detectarán el mecanismo de entrada de un navegador –la calidad del dispositivo señalador, la capacidad para el evento `hover` y algunas

otras definiciones especiales– de modo que podamos determinar con bastante precisión si nuestras miniaturas se están viendo en un dispositivo de pantalla táctil.

Tomemos esta media query como ejemplo (hace exactamente lo que podrías esperar):

```
@media (hover: none) { }
```

Como reglas para la media querie indicamos los estilos que deseamos que se apliquen para un navegador que no pueda manejar :hover. Especificaremos que para los dispositivos en los que el evento hover es imposible- [CSS3: galería con columnas de misma altura, autoajustable según contenido](#)

- [CSS3: galería con título y efectos con Grid CSS](#)
 - [Cuadrículas CSS responsivas, en tres líneas](#)
 - [Efecto hover \(al colocar el puntero del mouse\)](#)
 - [Superposición del título](#)
 - [Estilizamos el título](#)
 - [Oculta el título](#)
 - [Línea desenfocadas](#)
 - [El problema con las pantallas táctiles](#)

```
/* para dispositivos con pantalla táctil */  
@media (hover: none) {  
  .titulo-sitio {  
    opacity: 1;  
  }  
  
  .imagen-sitio img {  
    filter: blur(2px);  
  }  
}
```