

# CSS GRID RESPONSIVE

## 0. Conceptos previos

Solemos llamar “hero section” a una sección que acapara casi toda la atención. En este ejemplo, sería el “Hello, world!”

<https://getbootstrap.com/2.3.2/examples/hero.html>

Podemos leer más sobre el tema en este artículo:

<https://www.justinmind.com/blog/10-inspiring-hero-image-websites/>

En el que dan la siguiente definición:

*“A hero image is a large or oversized web banner image that is pinned to the header section of a webpage, usually towards the top of the page. Because of its prominent place in the site’s visual hierarchy, the hero header is often the first thing users see when arriving on a website.”*

Por otro lado llamamos “cards” a las imágenes que pueden ser artículos o links, acompañadas de cierto texto debajo, y que aparecen en forma de galería. Por ejemplo, una lista de artículos.

Aquí podemos ver distintos diseños de “cards”:

<https://freefrontend.com/css-cards/>

# 1. CSS Grid responsive

## 1.1. Contenido “hero” y lista de artículos

Vamos a tomar como ejemplo un layout o diseño de web común: una sección “hero”, con un grid de cartas debajo.

Ambos elementos deben responder a redimensionamientos de la pantalla adaptándose consecuentemente (reducirse de tamaño las cartas, y en determinados puntos saltar de línea).

Pensemos por un momento cómo lo haríamos...

Pues bien, vamos a hacerlo con sólo 6 líneas de código de CSS Grid, sin usar ni una sola @media query.

Podemos verlo en el siguiente ejemplo:

<https://codepen.io/imjuangarcia/pen/pGgmbe/>

En el que vemos un HTML y un css básicos, en el que cabe resaltar:

```
grid-template-columns: repeat(auto-fit, minmax(240px, 1fr));
```

Hasta ahora habíamos visto reglas como:

```
grid-template-columns: repeat(4, 1fr);  
grid-template-rows: repeat(2, 300px);
```

En este caso hemos usado minmax() que nos da unos valores mínimo y máximo que queremos que las columnas sean capaces de redimensionarse.

Además, con auto-fit se provoca un “wrap” haciendo que una columna pase a la fila de abajo cuando no hay suficiente espacio en nuestra pantalla para que quepa el valor mínimo de 240 px sin que rebose el contenido. Podríamos profundizar en la diferencia entre auto-fill y auto-fit, en el siguiente artículo:

<https://css-tricks.com/auto-sizing-columns-css-grid-auto-fill-vs-auto-fit/>