

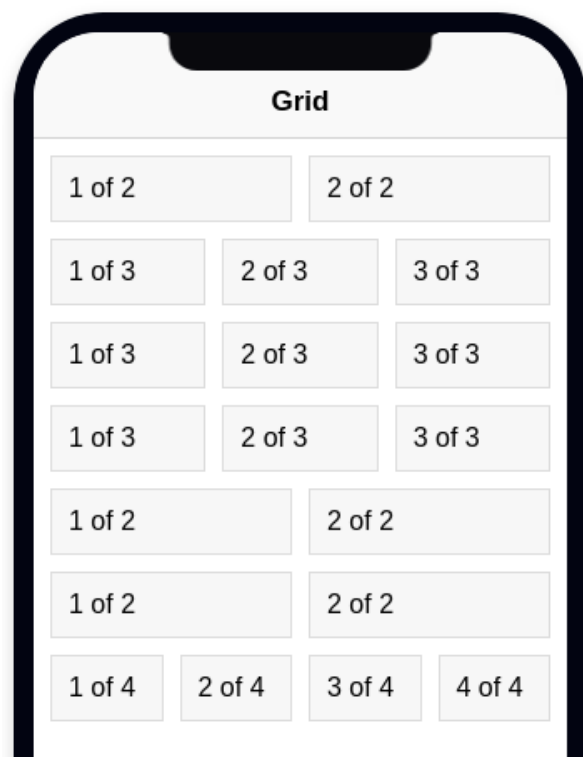
TEMA III

COMPONENTES IONIC Y SU USO EN APLICACIONES

3.4. Más Componentes

Ion-grid

Para utilizar y colocar bien los elementos de la app en la pantalla del móvil, utilizaremos este control junto con ion-row y ion-col. El funcionamiento es similar a bootstrap. Consideramos el ancho de la pantalla como 12 puntos. A partir de ahí cada fila define un número de columnas en función del ancho de las mismas.



Damos tamaño a nuestras columnas para organizar el espacio. Si la suma de los tamaños de nuestras columnas excede de 12, baja a la fila siguiente.

Adapta imagen y texto, si redimensionamos la pantalla no cambia el número de columnas sino el tamaño del contenido. No todo se ve igual según las dimensiones de la pantalla, es decir en un móvil quizás una columna por fila se ve bien, pero eso mismo en una tablet probablemente no. Este control nos proporciona una posibilidad para reajustar en función de la pantalla:

size, size-lg, size-md, size-sm

Pantallas grandes, medianas, pequeñas. Si no encaja en ninguna de esas tres le aplica el size.

Los tamaños asociados a esos atributos están en la documentación, en el componente

See the [Responsive Grid documentation](#) for more information.

Se pueden cambiar.

Al ion-grid le podemos añadir el atributo fixed. Añade márgenes a ambos lados para no cambiar el ancho del contenido de las columnas, es decir fija el contenido en función del ancho establecido

Ion-input

Este control nos servirá para cajas de texto de muchos tipos: text, password,email,...

Es importante elegir bien pq si queremos que aparezca la @ en el teclado necesitaremos que sea de tipo email

Es recomendable ponerlos dentro de un ion-list para desplazarlos juntos

```
<ion-content>
  <ion-list>
    <ion-list-header>
      <ion-label>Inputs</ion-label>
    </ion-list-header>
    <ion-item>
      <ion-label>Nombre:</ion-label>
      <ion-input type="text" placeholder="Tu nombre"></ion-input>
    </ion-item>
    <ion-item>
      <ion-label>Email:</ion-label>
      <ion-input type="email" placeholder="Tu email"></ion-input>
    </ion-item>
  </ion-list>
</ion-content>
```

Lo interesante, obviamente, es asociarlos a un formulario de manera que podamos usar las validaciones que incorpora Angular y asociarlos a campos de la clase para poder manejar la información introducida.

Para ello utilizaremos los formularios de angular a través del módulo `ReactiveFormsModule`.

Podemos manejar cada control de forma independiente con la clase **FormControl**, creando una variable de ese tipo en la clase y asociándola mediante `[formControl]` en la plantilla.

Aunque si tenemos varios controles es lógico agruparlos con la clase `FormGroup`, declaramos en la clase una variable (objeto) de tipo **FormGroup** para agrupar en ella todos los inputs (`FormControl`)

Veamos las tres posibilidades en el mismo ejemplo

Existe otra posibilidad, utilizar `Formbuilder` para declarar los objetos del `formgroup` en formato array en lugar de en notación de POO.

Podemos establecer reglas suyas de validación o añadir algún patrón nuestro a través del atributo `pattern` del `ion-input`.

Disponemos del evento `ngSubmit` asociado a la etiqueta `form`

ion-loading

Utilizaremos este componente para bloquear la interacción del usuario hasta que termine un proceso

En la documentación podemos copiar el evento para lanzarlo.

También podemos declararnos nuestra propia variable y nuestro evento para controlar cuándo lo queremos cerrar utilizando su método dismiss()

ion-menu

Este componente tiene dos identificadores importantes menuId, contentId. El primero es para poder abrir el menú cuándo queramos a través del método open de la clase MenuController, el segundo es para añadirlo al router-outlet, de esta manera sabe cuál es el menú que queremos inyectar.

Por lo demás las opciones son componentes ion-item que podemos configurar para redirigir a nuestras páginas.

Si queremos cerrar el menú después de elegir una opción <ion-menu-toggle>

Lo usual es generalizar los menús y cargar las opciones desde archivos json, cuándo veamos los servicios lo haremos así.