



# UA.MASTER MOVILES

MÁSTER UNIVERSITARIO EN DESARROLLO DE SOFTWARE  
PARA DISPOSITIVOS MÓVILES

## PROGRAMACIÓN HIPERMEDIA PARA DISPOSITIVOS MÓVILES

Ionic v4 – Componentes 2

1. Grid / Sistema de rejilla
2. Utilidades CSS
  1. Alineación del texto
  2. Ocultar elementos
3. Formularios
  1. Inputs
  2. Checkbox
  3. Radio buttons

- Ionic incluye un sistema de rejilla muy parecido al utilizado por Bootstrap.
- Para definir la rejilla haremos uso de tres etiquetas:
  - `<ion-grid>` → Raíz de la rejilla.
  - `<ion-row>` → Define las filas de la rejilla.
  - `<ion-col>` → Define las columnas de la rejilla.
- Por ejemplo, para definir una fila con 3 columnas del mismo tamaño:

```
<ion-grid>
  <ion-row>
    <ion-col> <div>1 de 3</div> </ion-col>
    <ion-col> <div>2 de 3</div> </ion-col>
    <ion-col> <div>3 de 3</div> </ion-col>
  </ion-row>
</ion-grid>
```

- Igual que en Bootstrap cada fila se divide en 12 huecos.
- Para definir un tamaño específico de columna usaremos el atributo “size”.

```
<ion-grid>
  <ion-row>
    <ion-col>          <div>1 de 3</div>      </ion-col>
    <ion-col size="8"> <div>2 de 3</div>      </ion-col>
    <ion-col>          <div>3 de 3</div>      </ion-col>
  </ion-row>
  <ion-row>
    <ion-col size="2"> <div>1 de 3</div>      </ion-col>
    <ion-col size="6"> <div>2 de 3</div>      </ion-col>
    <ion-col size="4"> <div>3 de 3</div>      </ion-col>
  </ion-row>
</ion-grid>
```

1 de 3	2 de 3		3 de 3
1 de 3	2 de 3	3 de 3	

- Al atributo “size” podemos añadirle los siguientes modificadores de tamaño de pantalla: **xs**, sm, md, lg y xl

```
<ion-grid>
  <ion-row>
    <ion-col size="4" size-md="3">    <div>1 de 4</div>    </ion-col>
    <ion-col size="4" size-md="3">    <div>2 de 4</div>    </ion-col>
    <ion-col size="4" size-md="3">    <div>3 de 4</div>    </ion-col>
    <ion-col size="4" size-md="3">    <div>4 de 4</div>    </ion-col>
  </ion-row>
  <ion-row>
    <ion-col size="12" size-md>    <div>1 de 3</div>    </ion-col>
    <ion-col size="12" size-md>    <div>2 de 3</div>    </ion-col>
    <ion-col size="12" size-md>    <div>3 de 3</div>    </ion-col>
  </ion-row>
</ion-grid>
```

1 de 4	2 de 4	3 de 4
3 de 4		
1 de 3		
2 de 3		
3 de 3		

1 de 4	2 de 4	3 de 4	3 de 4
1 de 3		2 de 3	3 de 3

- También podemos especificar que las columnas se ajusten al ancho de sus contenidos con:
  - `size="auto"`
  - `size-{TAM}="auto"`
    - Donde {TAM} puede ser `sm`, `md`, `lg`, `xl`

```
<ion-grid>
  <ion-row>
    <ion-col size="auto">    <div>1 de 2</div>    </ion-col>
    <ion-col>                <div>2 de 2</div>    </ion-col>
  </ion-row>
  <ion-row>
    <ion-col size-md="auto"> <div>1 de 2</div>    </ion-col>
    <ion-col>                <div>2 de 2</div>    </ion-col>
  </ion-row>
</ion-grid>
```

- Ionic incluye una serie de clases CSS que nos facilitan la alineación de contenidos:
  - `.ion-text-left` → Alineación a la izquierda.
  - `.ion-text-right` → Alineación a la derecha.
  - `.ion-text-start` → Depende de la dirección del texto (*left-to-right* o *right-to-left*).
  - `.ion-text-end` → Depende de la dirección del texto.
  - `.ion-text-center` → Centrar el texto.
  - `.ion-text-justify` → Justificar el texto.

```
<ion-col>
  <div class="ion-text-center">
    <h3>Texto centrado</h3>
  </div>
</ion-col>
```

- Ionic permite especificar la alineación del texto según el tamaño de pantalla utilizando las clases CSS:

`.ion-text-{SIZE}-{MODIFIER}`

- Donde:
  - {SIZE} **puede ser:** sm, md, lg, xl
  - {MODIFIER} **puede ser:** left, right, start, end, center, justify

```
<ion-col>
  <div class="ion-text-md-center">
    <h3>Texto centrado</h3>
  </div>
</ion-col>
```



- La clase CSS “ion-hide” nos permite ocultar contenidos.
- A esta clase podemos añadirle modificadores para que los contenidos solo se muestre u oculten para determinados tamaños de pantalla:

“ion-hide-{SIZE}-{DIR}”, donde:

- {SIZE} puede ser sm, md, lg o xl.
- {DIR} puede ser “up” o “down”.

```
<ion-col>
  <div class="ion-hide-sm-down"> ... </div>
</ion-col>
<ion-col>
  <div class="ion-hide-md-up"> ... </div>
</ion-col>
```

- Para añadir inputs (campos de edición) a un formulario utilizaremos la etiqueta “**<ion-input>**”:

```
<ion-input></ion-input>
```

- A esta etiqueta podemos asignarle las siguientes propiedades:
  - `name` → Nombre del campo al que se asignan los datos al enviar el formulario.
  - `value` → Valor del campo.
  - `type` → **Define el tipo del input:** `type="date|email|number|password|search|tel|text|time|url"`
  - `inputmode` → **Puede valer** `"none"`, `"text"`, `"tel"`, `"url"`, `"email"`, `"numeric"`, `"decimal"`, `"search"`

- El campo input dispone de más propiedades de configuración:
  - `placeholder` → Texto a mostrar dentro del input (se borrará cuando el usuario pulse sobre el campo).
  - `clearInput` → Añade un botón que permitirá borrar el contenido del input.
  - `autofocus` → Asigna el foco al campo indicado al abrir la página.
  - `color` → Selecciona el color del texto.
  - `max`, `min` → Limita el rango de valores.
  - `maxlength`, `minlength` → Limitar la longitud del campo.
  - `required` → Establecer como requerido.

```
<ion-input name="campo1" value="valor" type="number"
color="primary" placeholder="Escribe aquí..." autofocus
clearInput></ion-input>
```

- Para añadir etiquetas a un formulario usaremos “<ion-label>” y las agruparemos con el input dentro de un “<ion-item>”:

```
<ion-item>
  <ion-label>Etiqueta 1:</ion-label>
  <ion-input></ion-input>
</ion-item>
```

- Podemos configurar la posición de la etiqueta mediante:  
position= “fixed | stacked | floating”

---

Fixed:

---

Etiqueta Stacked:

---

Etiqueta Floating:

---

---

Fixed:

---

Texto ejemplo

---

Etiqueta Stacked:

---

Texto ejemplo

---

Etiqueta Floating:

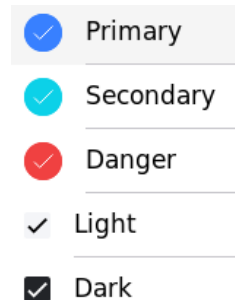
---

Texto ejemplo

---

- Para añadir checkbox disponemos de la etiqueta “`ion-checkbox`”.
- Podemos configurar su color, modo (`mode="md|ios"`) y slot (`slot="start|end"`):

```
<ion-checkbox color="primary"></ion-checkbox>
<ion-checkbox color="secondary"></ion-checkbox>
<ion-checkbox color="danger"></ion-checkbox>
<ion-checkbox color="light" mode="md"></ion-checkbox>
<ion-checkbox color="dark" mode="md"></ion-checkbox>
```



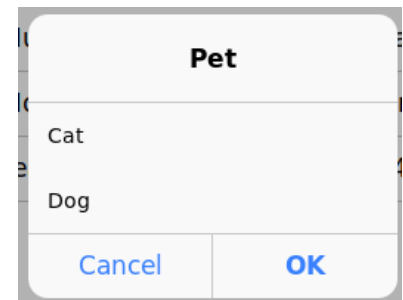
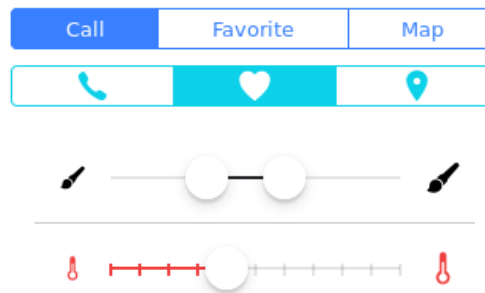
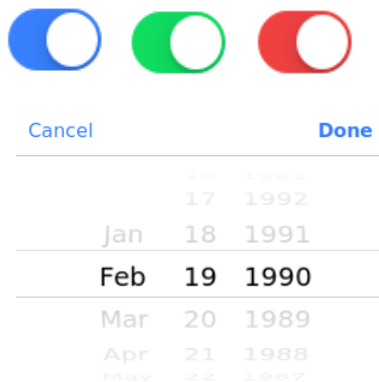
- Para añadir la etiqueta:

```
<ion-item>
  <ion-label>Primary</ion-label>
  <ion-checkbox slot="start" color="primary"></ion-checkbox>
</ion-item>
```

- Para añadir radio buttons usamos la etiqueta “ion-radio”.
- Para que sean excluyentes tenemos que agruparlos con “ion-radio-group”.
- También podemos configurar su color, modo, slot y marcarlos por defecto con “checked”.

```
<ion-list>
  <ion-radio-group>
    <ion-list-header>
      <ion-label>Fruta</ion-label>
    </ion-list-header>
    <ion-item>
      <ion-label>Manzana</ion-label>
      <ion-radio slot="start" value="manzana" checked></ion-radio>
    </ion-item>
    <ion-item>
      <ion-label>Plátano</ion-label>
      <ion-radio slot="start" value="platano"></ion-radio>
    </ion-item>
  </ion-radio-group>
</ion-list>
```

- Ionic incluye muchos más componentes que podemos utilizar para crear formularios y solicitar datos:
  - ion-select <https://ionicframework.com/docs/api/select>
  - ion-toggle <https://ionicframework.com/docs/api/toggle>
  - ion-datetime <https://ionicframework.com/docs/api/datetime>
  - ion-picker <https://ionicframework.com/docs/api/picker>
  - ion-range <https://ionicframework.com/docs/api/range>
  - ion-segment <https://ionicframework.com/docs/api/segment>



Para enviar un formulario utilizaremos:

- Binding de eventos añadiendo a la etiqueta de apertura del formulario:

```
<form (ngSubmit)="logForm()">
```

- Binding bidireccional añadiendo a cada input un binding con una variable del controlador:

```
<ion-input name="campo1" type="text"  
  [(ngModel)]="formulario.campo1"></ion-input>
```



## Ejemplo de formulario:

```
<form (ngSubmit)="logForm()">
  <ion-item>
    <ion-label>Campo 1:</ion-label>
    <ion-input [(ngModel)]="formulario.campo1" name="campo1">
      </ion-input>
  </ion-item>
  <ion-item>
    <ion-label>Campo 2:</ion-label>
    <ion-input [(ngModel)]="formulario.campo2" name="campo2">
      </ion-input>
  </ion-item>
  <ion-button type="submit">Enviar</ion-button>
</form>
```

Y en el controlador asociado tendríamos:

```
export class FormsPage {  
  formulario = {};  
  
  logForm() {  
    console.log(this.formulario);  
  }  
}
```

**¿PREGUNTAS?**