



# Componentes nativos de Ionic

UT4 - Introducción al desarrollo de aplicaciones hibridas con Ionic





# Objetivos de aprendizaje

- Conocer las características principales de Ionic
- Crear una primera aplicación usando Ionic
- Distinguir los principales módulos y componentes que componen una aplicación Ionic
- Introducir el concepto de decoración de clases (TS)
- Distinguir los conceptos de módulo y componente (Angular)





# Componentes de Ionic

- Ionic proporciona una amplia variedad de componentes prediseñados que podemos usar para desarrollar nuestra App.
- Son muy parecidos a los ofrecidos por las interfaces nativas
- Los componentes son etiquetas HTML que podemos usar en cualquier parte de nuestra aplicación.





# Componentes de Ionic

- Los componentes de Ionic incluyen
  - Atributos propios
  - Estilos CSS encapsulados
  - Eventos propios
    - Distintos de los usados en Web porque la interacción móvil es diferente.





# Extensión de VSCode

- Ionic Snippets
  - Autocompleta etiquetas o propiedades de Ionic a partir de la letra i -

The screenshot shows the Visual Studio Code Marketplace interface. On the left, there is a search bar with the text "ionic sni". Below it, a list of extensions is displayed, with "Ionic Snippets" at the top. To the right, the details page for "Ionic Snippets" is shown. The page includes the extension's logo (a blue "IX" icon), its name "Ionic Snippets", its version "v2.2.2", the developer "fivethree", the download count "357,442", and a 5-star rating. It also features tabs for "DETAILS", "FEATURES", and "CHANGELOG". Below this, there is a large "Ionic X VS Code Snippets" logo. To the right of the logo, there are sections for "Categories" (Snippets), "Resources" (Marketplace Issues, Repository, License), and "More Info" (Published date, Last released date, Identifier). At the bottom, there is a note stating "All code snippets are based on and follow the Ionic 5 API and Ionic 4 API".



# Sass y CSS3



- Como sabemos, los estilos habitualmente se definen en CSS3
- Tanto HTML como CSS carecen de las construcciones habituales de un lenguaje de programación (variables, funciones, bucles)
- Sass es una extensión de CSS que añade estas características a CSS
  - Otra de las más conocidas es Less

{less}

# Sass y CSS3



- La manera de trabajar habitual de Sass es:
  - Escribimos los estilos en Sass (archivo .scss)
  - Compilamos el código a CSS
    - Necesita compilación porque el navegador no entiende Sass
  - Usamos el CSS en nuestra página Web

Ionic se encarga de realizar este proceso de manera automática para nosotros

# Variables en CSS3

- En sus últimas especificaciones, CSS3 ha incluido el concepto de variable
  - La especificación oficial las llama propiedades personalizadas

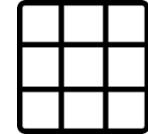
Definición	Anteponemos -- al nombre que le queremos dar a la variable	--color-principal:#FABADA;
Uso	var(variable);	p{ color: <u>var(--color-principal)</u> ; }



# El Grid de Ionic

Componentes nativos de Ionic



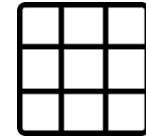


# El Grid de Ionic

- Está organizado en filas y columnas
- El contenido que agregamos va dentro de las columnas, no de las filas
- Cada fila está dividida en 12 columnas.
  - Podemos determinar cuánto espacio queremos que ocupe cada columna.

Estructura similar a la usada en frameworks de diseño como Bootstrap



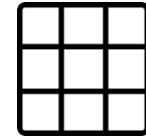


# El Grid de Ionic

ion-grid	Definición de un Grid
ion-row	Definición de una fila
ion-col	Definición de columnas dentro de mi fila. Por defecto se dividen equitativamente (el número de columnas que ocupa un elemento depende de cuantas columnas hay en la fila)

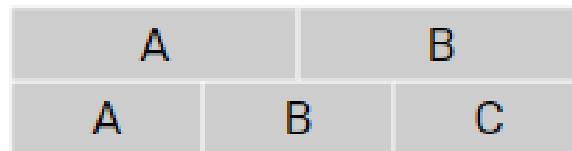
No deberíamos usar demasiadas columnas  
Al trabajar para móvil, el contenido podría ser ilegible





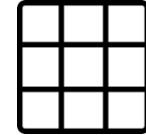
# El Grid de Ionic

```
<ion-grid class="ion-text-center">  
  <ion-row>  
    <ion-col>A</ion-col>  
    <ion-col>B</ion-col>  
  </ion-row>  
  <ion-row>  
    <ion-col>A</ion-col>  
    <ion-col>B</ion-col>  
    <ion-col>C</ion-col>  
  </ion-row>  
</ion-grid>
```



Los Grid ocupan el 100% del contenedor donde estamos situados

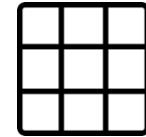




# <ion-grid>

Atributo	Explicación	Valores
fixed	Permite especificar el ancho del grid	<p>false → Ocupa lo que ocupe el contenido true→ El ancho del grid depende del breakpoint en el que nos encontramos:</p> <ul style="list-style-type: none"><li>- En pantallas pequeñas (&lt;100%) el ion-grid ocupa el 100% de la pantalla.</li><li>- En pantallas mayores el ancho se ajusta a un valor fijo (540, 720..) que depende del tamaño de la pantalla</li></ul>



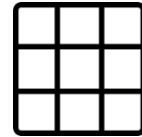


# <ion-col>

Atributo	Explicación	Valores
size	Permite especificar cuantas columnas va a ocupar la columna	auto → Ocupa lo que ocupe el contenido numero: entero → Número de columnas determinado
offset	Permite desplazar una columna	numero: entero → Número de columnas que desplazo hacia la derecha

```
<ion-grid class="ion-text-center">
  <ion-row>
    <ion-col size="auto">A</ion-col>
    <ion-col>B</ion-col>
  </ion-row>
  <ion-row>
    <ion-col size="8">A</ion-col>
    <ion-col>B</ion-col>
    <ion-col>C</ion-col>
  </ion-row>
</ion-grid>
```

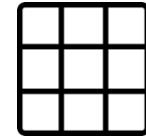




# Tamaños

- Podemos añadir un sufijo a la propiedad `size` que determina cuanto ocupa en función de la resolución a la que se va a aplicar la propiedad

Sufijo	Resolución a la que se aplica
<code>xs</code>	<576px
<code>sm</code>	Entre 576px y 768px
<code>md</code>	Entre 768px y 992px
<code>lg</code>	Entre 992px y 120px
<code>xl</code>	>1200px



# Tamaños

```
<ion-grid class="ion-text-center">  
  <ion-row>  
    <ion-col size="12">A</ion-col>  
    <ion-col size="12">B</ion-col>  
    <ion-col size="12">C</ion-col>  
  </ion-row>  
</ion-grid>
```



Todas las resoluciones

A  
B  
C

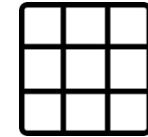
```
<ion-grid class="ion-text-center">  
  <ion-row>  
    <ion-col size="12" size-sm="9">A</ion-col>  
    <ion-col size="12" size-sm="9">B</ion-col>  
    <ion-col size="12" size-sm="9">C</ion-col>  
  </ion-row>  
</ion-grid>
```

Resoluciones <sm

A  
B  
C

Resoluciones sm y mayores

A  
B  
C



# Tamaños



```
<ion-grid class="ion-text-center">  
  <ion-row>  
    <ion-col size="12" size-md="9" size-lg="6">A</ion-col>  
    <ion-col size="12" size-md="9" size-lg="6">B</ion-col>  
    <ion-col size="12" size-md="9" size-lg="6">C</ion-col>  
  </ion-row>  
</ion-grid>
```

Resoluciones < md

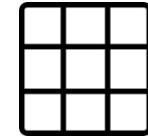


Resoluciones entre md y lg



Resoluciones lg y mayores





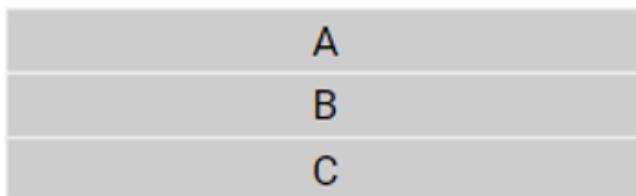
# Offsets



```
<ion-grid class="ion-text-center">
  <ion-row>
    <ion-col size="12" size-md="9" size-la="6">A</ion-col>
    <ion-col size="12" size-md="9" size-la="6">B</ion-col>
    <ion-col size="12" size-md="9" size-la="6">C</ion-col>
  </ion-row>
</ion-grid>
```

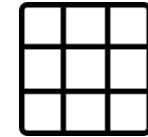
Resoluciones < md

Resoluciones entre md y lg



Resoluciones lg y mayores





# Offsets

```
<ion-grid class="ion-text-center">  
  <ion-row>  
    <ion-col size="6" offset="2">A</ion-col>  
  </ion-row>  
</ion-grid>
```



A

```
<ion-grid class="ion-text-center">  
  <ion-row>  
    <ion-col size="6" offset="2" offset-xl="4">A</ion-col>  
  </ion-row>  
</ion-grid>
```



Podemos combinarlo con breakpoints



# Utilizando componentes

- Al no usar módulos, tenemos que importar los componentes de Ionic en nuestro propio componente

TS

```
import { Component } from '@angular/core';
import { IonGrid, IonRow, IonCol } from '@ionic/angular/standalone';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
  imports:[IonGrid, IonRow, IonCol]
})
export class HomePage {
  constructor() {}
}
```



# Utilizando componentes

- Para evitar tener que importar componentes uno a uno, podemos importar IonicModule importar que incluye todos los componentes:

```
import { Component } from '@angular/core';
import { IonicModule } from '@ionic/angular';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
  imports:[IonicModule]
})

export class HomePage {
  constructor() {}
}
```

TS





# Componentes básicos

Componentes nativos de Ionic





# Cabecera, contenido, pie

Etiqueta	Uso
ion-header	<p>Cabecera de nuestras vistas. Se alinea al comienzo de la página</p> <p>También puede representar la cabecera de un elemento concreto.</p>
ion-content	<p>Contenido de nuestras vistas.</p> <p>Ocupa el 100% del ancho y 100% de alto.</p> <p>Proporciona scroll integrado si el contenido supera el 100%</p>
ion-footer	<p>Pie de página de nuestras vistas. Se alinea al final de la página</p>





# <ion-header>

Etiqueta	Uso
mode	Modo de funcionamiento [ ios   md ]
translucent	[El contenido debe tener fullscreen] [Solo funciona en modo iOS] Permite al contenido hacer scroll tras la cabecera
collapse	Efecto de scroll que se aplicará [condense   fade ] [Solo funciona en modo iOS]

Se recomienda usar el header para incluir toolbars  
Cuando incluimos una toolbar en el header, su contenido se ajusta



# Algunos componentes básicos

<a href="#"><u>ion-text</u></a>	<a href="#"><u>ion-button</u></a>
<a href="#"><u>ion-list</u></a>	<a href="#"><u>ion-item</u></a>
<a href="#"><u>ion-icon</u></a>	<a href="#"><u>ion-card</u></a>
<a href="#"><u>ion-toolbar</u></a>	





# <ion-text>

Atributo	Explicación	Valores
Color	Color que queremos aplicar tomado de la paleta de colores	primary   secondary   tertiary   success   danger   warning   dark   medium   light   <u>undefined</u>

```
<ion-text color="primary">  
  <h1>Mi viaje por Guarromán</h1>  
</ion-text>>
```

Mi viaje por Guarromán





# <ion-button>

- Elemento pulsable

Atributo	Explicación	Valores
[disabled]	Deshabilita elemento	true   false
expand	Expande el elemento	block   full
shape	Permite elementos redondos	round
fill	Tipo de relleno	clear   outline   solid
size	Tamaño	small   default   large

```
<ion-button color="warning" size="large" fill="outline">  
    Favoritos  
    <ion-icon slot="end" name="star"></ion-icon>  
</ion-button>
```



FAVORITOS ★



# <ion-list>

- Permite crear listas con elementos que serán iguales
- Cada elemento estará formado por un <ion-item>
  - Dentro de los ion-item metemos otros elementos como ion-label

Atributo	Explicación	Valores
inset	Si es verdadero, la lista tiene márgenes alrededor	Booleano (true   <u>false</u> )
lines	Especifica como debería mostrar los bordes inferiores en los elementos de la lista  También se puede aplicar a cada ion-item, en ese caso prevalece sobre el estilo de la lista	full   inset   none   <u>undefined</u>



# <ion-list>

```
<ion-list>
<ion-item><ion-label>León</ion-label></ion-item>
<ion-item><ion-label>Hormiga</ion-label></ion-item>
<ion-item><ion-label>Pez</ion-label></ion-item>
<ion-item><ion-label>Tigre</ion-label>></ion-item>
<ion-item><ion-label>Araña</ion-label></ion-item>
</ion-list>
```



León

Hormiga

Pez

Tigre

Araña





# <ion-list-header>

- Se pone al principio de la lista para darle un título a la misma y la acompaña



```
<ion-list>
  <ion-list-header>
    <h1>Animales salvajes</h1>
  </ion-list-header>
  <ion-item>León</ion-item>
  <ion-item>Hormiga</ion-item>
  <ion-item>Pez</ion-item>
  <ion-item>Tigre</ion-item>
  <ion-item>Araña</ion-item>
</ion-list>
```

Animales salvajes

León

Hormiga

Pez

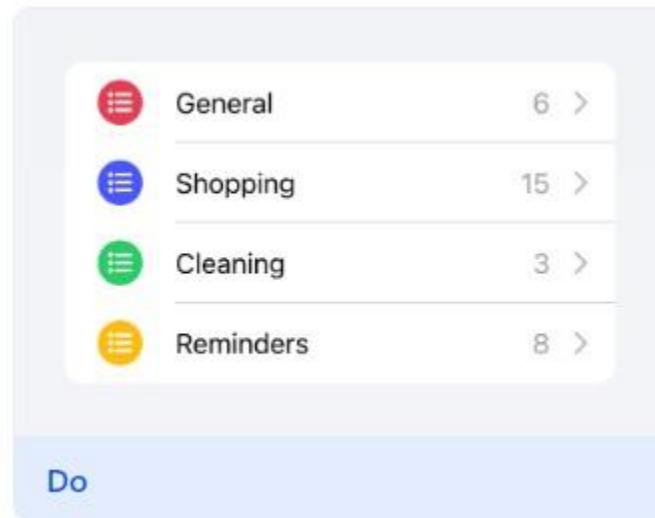
Tigre

Araña



# <ion-item>

- Elementos de una lista
- Permiten una amplia variedad de elementos internos: Iconos, botones, checkboxes...





# Organizando elementos en listas

Etiqueta	Explicación
<a href="#"><code>&lt;ion-item-divider&gt;</code></a>	Permite separar elementos en una lista
<a href="#"><code>&lt;ion-item-group&gt;</code></a>	Agrupan elementos similares
<a href="#"><code>&lt;ion-item-sliding&gt;</code></a>	Añade elementos expandibles al arrastrar
<a href="#"><code>&lt;ion-item-options&gt;</code></a>	Muestra opciones en un elemento expandible
<a href="#"><code>&lt;ion-label&gt;</code></a>	Información general de la etiqueta
<a href="#"><code>&lt;ion-note&gt;</code></a>	Añaden subtítulos que dan más información a la etiqueta





# <ion-icon>

- Permite añadir un ícono de la librería [ionicons](#)

Atributo	Explicación	Valores
logo-X	Ícono que aplicamos	Ver códigos
size	Tamaño del icon	small   large
color	Color del ícono	primary   secondary..

```
<ion-icon name="tv-outline"
          color="danger"
          size="large">
</ion-icon>
```





# Usando iconos

- Si usamos iconos necesitamos importarlos y agregarlos en el componente.
  - Podemos hacerlo en AppComponent si queremos usar los iconos en todos los componentes

```
import { addIcons } from 'ionicons';
import { logoIonic, starHalfOutline } from 'ionicons/icons';
export class AppComponent {
  constructor() {
    addIcons({ logoIonic, starHalfOutline });
  }
}
```

TS

app/app.component.ts

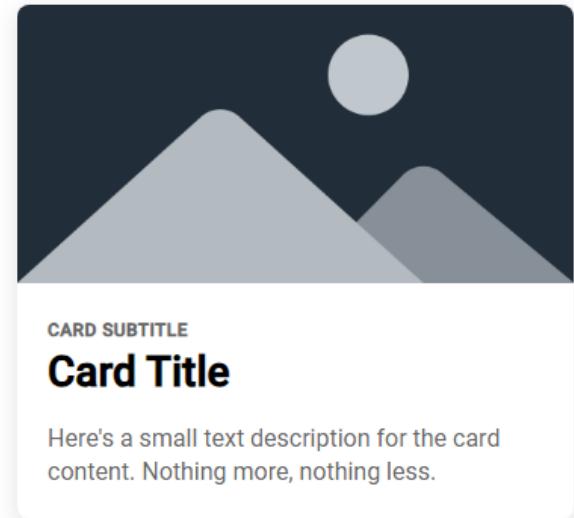




# <ion-card>

- Componentes que permiten mostrar imágenes, textos, botones, listas...
- Etiquetas internas

Etiqueta	Explicación
<ion-card-content>	Contenido
<ion-card-header>	Cabecera
<ion-card-title>	Título
<ion-card-subtitle>	Subtítulo





# <ion-toolbar>

- La barra de herramientas se suele posicionar al comienzo o al fin de la página y proporciona contenido y acciones.
  - Si se posiciona en header o footer, no hace scroll con la página (se posiciona de manera fija)
  - Si se posiciona en el contenido, hace scroll con la página



# Toolbar y botones

- Los **botones** en un tool bar pueden llevar el atributo slot que indica su rol dentro del toolbar

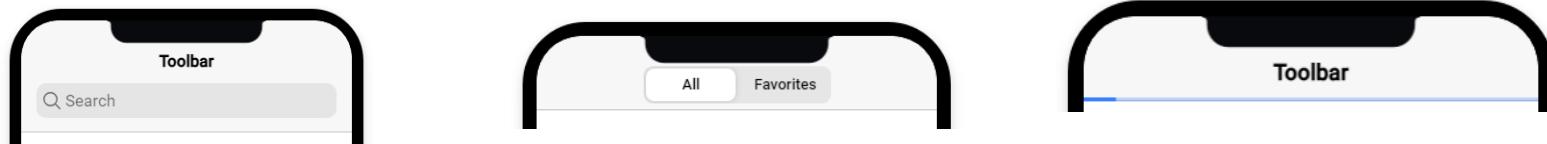
Atributo	Explicación	Valores
slot	Rol del botón	start   end   primary   secondary icon-only [aplicado a iconos]



# <ion-toolbar>

- Etiquetas que podemos usar internamente

Etiqueta	Explicación
<a href="#"><u>&lt;ion-title&gt;</u></a>	Título de la toolbar, especifica en qué sección estamos
<a href="#"><u>&lt;searchbar&gt;</u></a>	Debería ser el único hijo del toolbar, y coge toda la anchura y altura del mismo
<a href="#"><u>&lt;ion-segment&gt;</u></a>	Permite cambiar entre dos vistas
<a href="#"><u>&lt;ion-progress-bar&gt;</u></a>	Muestra un proceso de carga





# Utilidades CSS

Componentes nativos de Ionic





# Utilidades CSS

- Ionic aporta una gran variedad de utilidades CSS para:
  - Modificar el texto
  - Posicionamiento del elemento
  - Espaciado
  - Bordes
  - Propiedades Flex

A la mayor parte de las clases le podemos añadir el breakpoint (`xs`, `sm`, `md`, `lg`, `xl`) para indicar en qué resoluciones





# Utilidades CSS: Alineación de texto

## Posibles valores

ion-text-start	ion-text-justify
ion-text-end	ion-text-wrap
ion-text-center	ion-text-nowrap

```
<div class="ion-text-center">  
  <h3>text-center</h3>  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
</div>
```



# Utilidades CSS: Transformación de texto

## Posibles valores

ion-text-uppercase

ion-text-lowercase

ion-text-capitalize

```
<div class="ion-text-uppercase">  
    <h3>text-center</h3>  
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
</div>
```





# Utilidades CSS: Varios

Valor	Descripción
ion-hide	Oculta un elemento
ion-no-border	Oculta el borde de un elemento

```
<div class="ion-hide">  
  <h3>text-center</h3>  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
</div>
```





# Utilidades CSS: Padding

Posibles valores	
ion-padding	ion-padding-top
ion-padding-start	ion-padding-end
ion-padding-bottom	ion-padding-horizontal
ion-padding-vertical	ion-no-padding

```
<div class="ion-padding-start">  
  <h3>text-center</h3>  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
</div>
```



El valor base de padding (16px) se puede redefinir en la variable --ion-padding





# Utilidades CSS: Margen

Posibles valores	
ion-margin	ion-margin-top
ion-margin-start	ion-margin-end
ion-margin-bottom	ion-margin-horizontal
ion-margin-vertical	ion-no-margin

```
<div class="ion-margin-start">  
  <h3>text-center</h3>  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
</div>
```

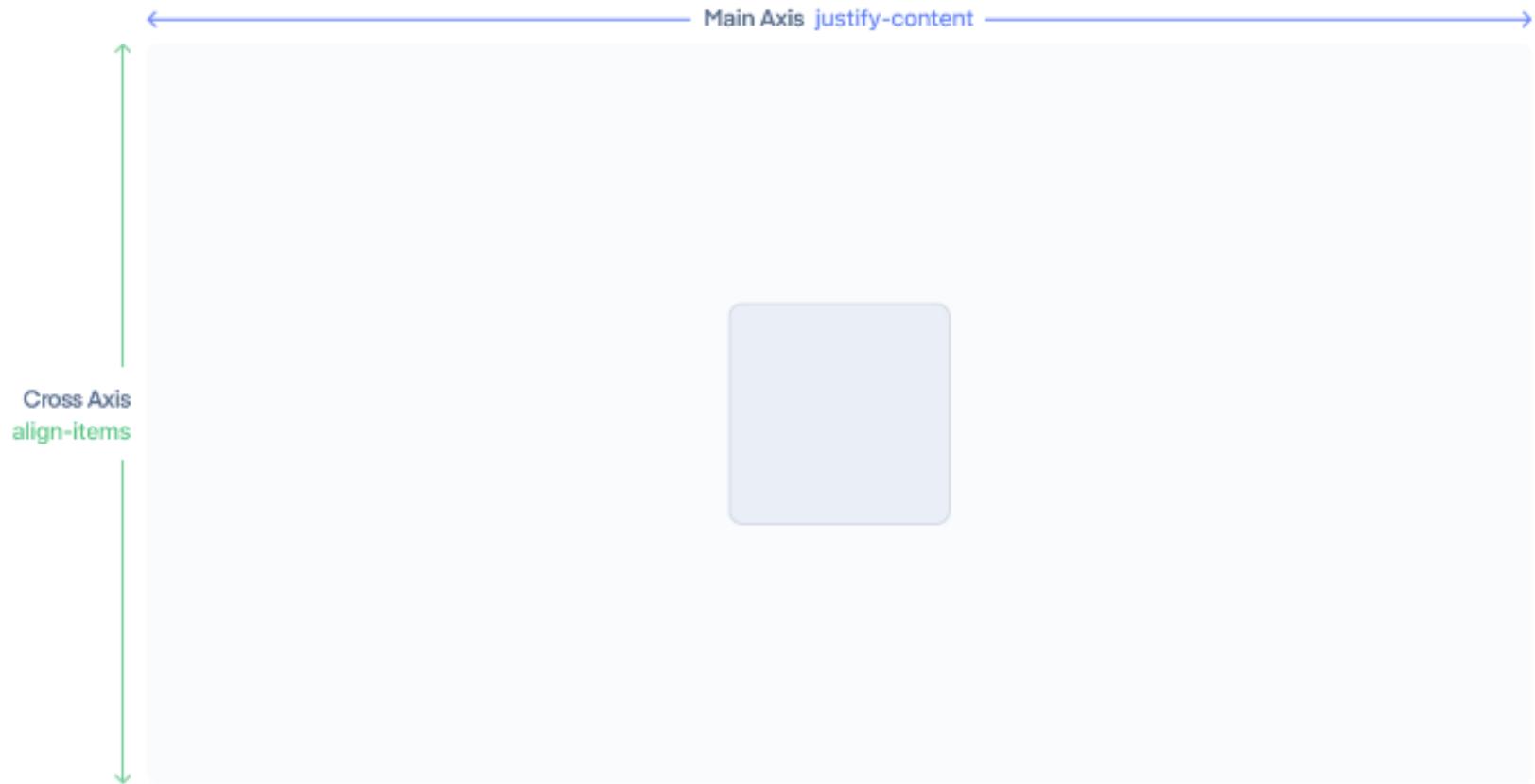


El valor base de margin (16px) se puede redefinir en la variable --ion-margin





# Utilidades CSS: Propiedades Flex





# Utilidades CSS: Contenedores Flex

Posibles valores	
ion-justify-content-start	ion-align-items-start
ion-justify-content-center	ion-align-items-center
ion-justify-content-end	ion-align-items-end
ion-justify-content-baseline	ion-align-items-baseline
ion-justify-content-between	ion-align-items-stretch
ion-justify-content-evenly	
ion-justify-content-center	

Lo aplicamos a filas de un row



# Utilidades CSS: Contenedores Flex



```
<ion-grid>
  <ion-row class="ion-justify-content-end">
    <ion-col size="auto">
      <div>Hola</div>
    </ion-col>
    <ion-col size="auto">
      <div>Hola</div>
    </ion-col>
  </ion-row>
</ion-grid>
```





# Utilidades CSS: Elementos Flex

## Posibles valores

ion-align-self-start

ion-align-self-end

ion-align-self-center

Lo aplicamos a elementos dentro de un row (col)





# Utilidades CSS: Elementos Flex



```
<ion-grid>
  <ion-row>
    <ion-col class="ion-align-self-start">
      <div>1 of 4</div>
    </ion-col>
    <ion-col class="ion-align-self-center">
      <div>2 of 4</div>
    </ion-col>
    <ion-col class="ion-align-self-end">
      <div>3 of 4</div>
    </ion-col>
    <ion-col>
      <div>4 of 4 # # #</div>
    </ion-col>
  </ion-row>
</ion-grid>
```





# Personalización de estilo

Componentes nativos de Ionic





# Redefiniendo el estilo

- Podemos redefinir el modo en que se comporta un elemento determinado
  - Cada elemento tiene sus propias propiedades CSS que puedo redefinir ([ejemplo: Botón](#))



ion-text{

--background:#FABADA;  
--color:white;

}

Selector CSS para acceder al elemento cuyo estilo quiero definir

Variables que redefinen el color y el color de fondo de dicho componente



# Paleta de colores

- Ionic define una paleta de colores básica que podemos usar para dar color a cualquier componente
  - En lugar de usar códigos de color, uso las palabras clave primary, secondary...

Primary	#3880ff	▼
Secondary	#3dc2ff	▼
Tertiary	#5260ff	▼
Success	#2dd36f	▼
Warning	#ffc409	▼
Danger	#eb445a	▼
Dark	#222428	▼
Medium	#92949c	▼
Light	#f4f5f8	▼



# Estilos de plataforma

- Ionic tiene dos modos que se pueden usar para adaptar el aspecto de los componentes de cada plataforma:
  - ios → Estilos para IOS
  - md → Estilos para Android [Material Design]
- La clase se inyecta en el HTML al generar para una plataforma específica.
- Cada plataforma tiene su modo por defecto pero se puede adaptar.



# Adaptando estilos de plataforma

- Podemos adaptar los estilos generando reglas que anteponen una clase con el nombre de la plataforma



```
.ios ion-badge {  
    text-transform: uppercase;  
}
```



# Estilos CSS globales

- Los estilos CSS deberían ir en cada componente de manera individual
- Si queremos aplicar algún estilo de manera global lo podemos incluir en el archivo global.scss
- Por ejemplo, podemos redefinir la paleta correspondiente a un modificador de color:

```
:root {  
    --ion-color-secondary: #006600;  
    --ion-color-secondary-rgb: 0, 102, 0;  
    --ion-color-secondary-contrast: #ffffff;  
    --ion-color-secondary-contrast-rgb: 255, 255, 255;  
    --ion-color-secondary-shade: #005a00;  
    --ion-color-secondary-tint: #1a751a;  
}
```





# Generador de paletas

**ionic** DOCS

Getting Started

- Overview
- Environment Setup
- CLI Installation
- Packages & CDN
- Ionic VS Code Extension
- Next Steps

Upgrade Guides

- Updating to v7
- Updating to v6
- Updating to v5
- Updating to v4

Developing

- Starting
- Previewing

generated code directly into your Ionic project.

Primary

#3880ff

Secondary

#3dc2ff

Tertiary

#5260ff

Success

#2dd36f

Warning

#ffc409

Danger

#eb445a

iOS      Android

