

# **Desarrollo de aplicaciones para móviles**

## **Modulo 2: Desarrollo de aplicaciones**

### **CSS Theming**

---



## Presentación:

En la presente unidad aprenderemos como customizar la visual de nuestra aplicación.



## Objetivos:

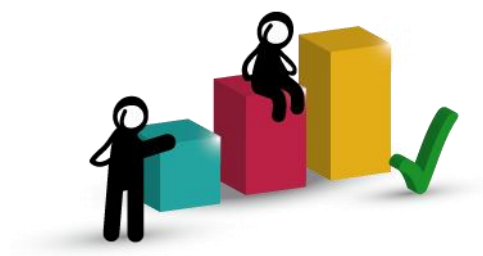
### Que los participantes\*:

- Aprender los mecanismos básicos para customizar las aplicaciones.
- Pueden utilizar el responsive provisto por IONIC
- Aprendan a utilizar de manera básica Saas



## Bloques temáticos\*:

- Modificaciones de texto
- Transformaciones de texto
- Contenido
- Márgenes
- Responsive grid
- Tamaño de las grillas
- Atributos
- Breakpoints
- Colors theming
- Saas variables
- Especificaciones propias de las plataformas



## Consignas para el aprendizaje colaborativo

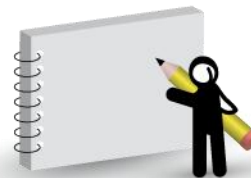
En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC\*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.



## Tomen nota\*

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.

## IONIC Theme

---

Ionic nos prove ciertas funcionalidades resueltas a la hora de querer darle estilos a nuestras aplicaciones.

Podemos ver las mismas en: <http://ionicframework.com/docs/theming/css-utilities/>

### Modificaciones de texto

Atributo	Estilo CSS
<b>text-left</b>	<b>text-align: left</b>
<b>text-right</b>	text-align: right
<b>text-center</b>	text-align: center
<b>text-justify</b>	text-align: justify
<b>text-wrap</b>	white-space: nowrap
<b>text-nowrap</b>	white-space: normal

### Transformaciones de texto

Atributo	Estilo CSS equivalente
<b>text-uppercase</b>	text-transform: uppercase
<b>text-lowercase</b>	text-transform: lowercase
<b>text-capitalize</b>	text-transform: capitalize





## Contenido

Atributo	Estilo CSS equivalente
<b>padding</b>	padding: 16px
<b>padding-top</b>	padding-top: 16px
<b>padding-left</b>	padding-left: 16px
<b>padding-right</b>	padding-right: 16px
<b>padding-bottom</b>	padding-bottom: 16px
<b>padding-vertical</b>	padding: 16px 0
<b>padding-horizontal</b>	padding: 0 16px
<b>no-padding</b>	padding: 0

## Márgenes

Atributo	Estilo CSS equivalente
<b>margin</b>	margin: 16px
<b>margin-top</b>	margin-top: 16px
<b>margin-left</b>	margin-left: 16px
<b>margin-right</b>	margin-right: 16px
<b>margin-bottom</b>	margin-bottom: 16px
<b>margin-vertical</b>	margin: 16px 0
<b>margin-horizontal</b>	margin: 0 16px
<b>no-margin</b>	margin: 0

## Responsive grid

La grilla está compuesta por filas y columnas. Para esto debemos utilizar **ion-grid** para crear la grilla, **ion-row** para la fila y **ion-col** para las columnas.

```

1 of 3      2 of 3      3 of 3

<ion-grid>
  <ion-row>
    <ion-col>
      1 of 3
    </ion-col>
    <ion-col>
      2 of 3
    </ion-col>
    <ion-col>
      3 of 3
    </ion-col>
  </ion-row>
</ion-grid>

```

En este caso vemos 3 columnas del mismo tamaño.

## Tamaño de las grillas

Por defecto la grilla toma el 100% del ancho. El tamaño máximo de la grilla esta definido en la variable saas **\$grid-max-widths**.

Name	Value	Description
<b>xs</b>	auto	No setea la grid a ningun tamaño, utilizado para que se acomode a los Dispositivos mobiles
<b>sm</b>	540px	Setea el grid a 540px (min-width: 576px)
<b>md</b>	720px	Setea el grid a 720px (min-width: 768px)
<b>lg</b>	960px	Setea el grid a 960px (min-width: 992px)
<b>xl</b>	1140px	Setea el grid a 1140px (min-width: 1200px)

Las columnas se acomodan automáticamente al tamaño máximo que será identificado con el número 12.

Por ejemplo:

1 of 2	2 of 2	
1 of 3	2 of 3	3 of 3

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

También podremos definir si queremos darle un tamaño particular a una columna (una columna más grande que otra) por ejemplo:



1 of 3	2 of 3 (wider)	3 of 3
1 of 3	2 of 3 (wider)	3 of 3

```

<ion-grid>
  <ion-row>
    <ion-col>
      1 of 3
    </ion-col>
    <ion-col col=8>
      2 of 3 (wider)
    </ion-col>
    <ion-col>
      3 of 3
    </ion-col>
  </ion-row>
  <ion-row>
    <ion-col>
      1 of 3
    </ion-col>
    <ion-col col=6>
      2 of 3 (wider)
    </ion-col>
    <ion-col>
      3 of 3
    </ion-col>
  </ion-row>
</ion-grid>

```

Aca vemos que la segunda columna tiene la clase **col-8**, recordemos que como máximo tenemos 12. Es decir podemos poner a una columna **col-2** y a las otras **col-8** y **col-4** pero en ningún caso puede ser mayor a **col-12**.

## Atributos

Propiedad	Descripción
<b>no-padding</b>	Removes padding from the grid and immediate children columns.
<b>fixed</b>	Set a max width based on the screen size.

## Breakpoints

Nombre	Valor	Ancho	Descripción
<b>xs</b>	0	col-	Set columns when (min-width: 0)
<b>sm</b>	576px	col-sm-	Set columns when (min-width: 576px)
<b>md</b>	768px	col-md-	Set columns when (min-width: 768px)
<b>lg</b>	992px	col-lg-	Set columns when (min-width: 992px)
<b>xl</b>	1200px	col-xl-	Set columns when (min-width: 1200px)

Podemos ver mas sobre responsive grid en:

<http://ionicframework.com/docs/theming/responsive-grid/>

## Colors theming

En **src/them/variables.scss** podremos encontrar las reglas para modificar el color de nuestra app.

```
$colors: (  
  primary:    #488aff,  
  secondary:  #32db64,  
  danger:     #f53d3d,  
  light:      #f4f4f4,  
  dark:       #222  
);
```

Podemos agregar a la base de colores nuevas variables, como por ejemplo:

```
$colors: (  
  primary:    #488aff,  
  secondary:  #32db64,  
  danger:     #f53d3d,  
  light:      #f4f4f4,  
  dark:       #222,  
  twitter: (  
    base: #55acee,  
    contrast: #ffffff  
  )  
);
```

Aquí vemos que agregamos el índice twitter, con sus subíndices base y contrast.

Luego si queremos utilizar este nuevo color en otro componente css debemos hacer lo siguiente:

```
my-component {  
  background: color($colors, twitter, base);  
}
```

## Saas variables

Las variables saas nos facilitan el trabajo, ya que podemos definir las mismas una vez y utilizarlas multiples veces.

Es decir en caso de querer modificar el valor de la variable lo haremos en un solo lugar, y el cambio se modificara en el resto de nuestra aplicación.

Las variables saas comienzan con \$

```
$control-height: 40px;
```

Y para utilizarlas solo debemos hacer referencia a ella:

```
.header {  
  height: $control-height;  
}  
  
.sub-header {  
  height: $control-height;  
}
```

---

## Especificaciones propias de las plataformas

Ionic utiliza modos para customizar los componentes. Cada plataforma posee un modo por default que puede ser sobrescrito por el programador.

Por ejemplo una aplicación en android utilizara el modo **md** (material design)

El tag **ion-app** tendrá la clase **class="md"** agregado como default.

Plataforma	Modo	Detalle
<b>ios</b>	ios	Modo utilizado en dispositivos que tengan IOS
<b>android</b>	md	Utilizado en dispositivos con android. Basado en Material Design.
<b>windows</b>	wp	Utilizado en dispositivos con windows phone
<b>core</b>	md	En cualquier plataforma, basado en material design.



## Ejercicio

---

Continuamos trabajando en la app del colegio Manuel Belgrano.

Nos han pedido que personalizemos la misma para que utilice los colores propios del colegios (azul, celeste y blanco)

Por otro lado nos han pedido que mostremos una demo del listado de mis cursos, para lo cual haremos una responsive grid con los siguientes datos:

- Nombre del curso
- Año
- Cantidad de alumnos

Al hacer clic en un curso se debe redirigir a una nueva pantalla en la cual veremos el detalle del curso (utilizando responsive grid):

- Nombre y apellido del alumno
- Notas
  - Evaluación
  - TP
  - Promedio



## Bibliografía utilizada y sugerida

<http://ionicframework.com/docs/theming/>

<http://ionicframework.com/docs/theming/css-utilities/>

<http://ionicframework.com/docs/theming/responsive-grid/>

## Lo que vimos:

En esta unidad vimos como personalizar nuestra app y como utilizar responsive grid para mostrar datos.



## Lo que viene:

En la próxima unidad veremos persistencia de datos a través del componente Storage

