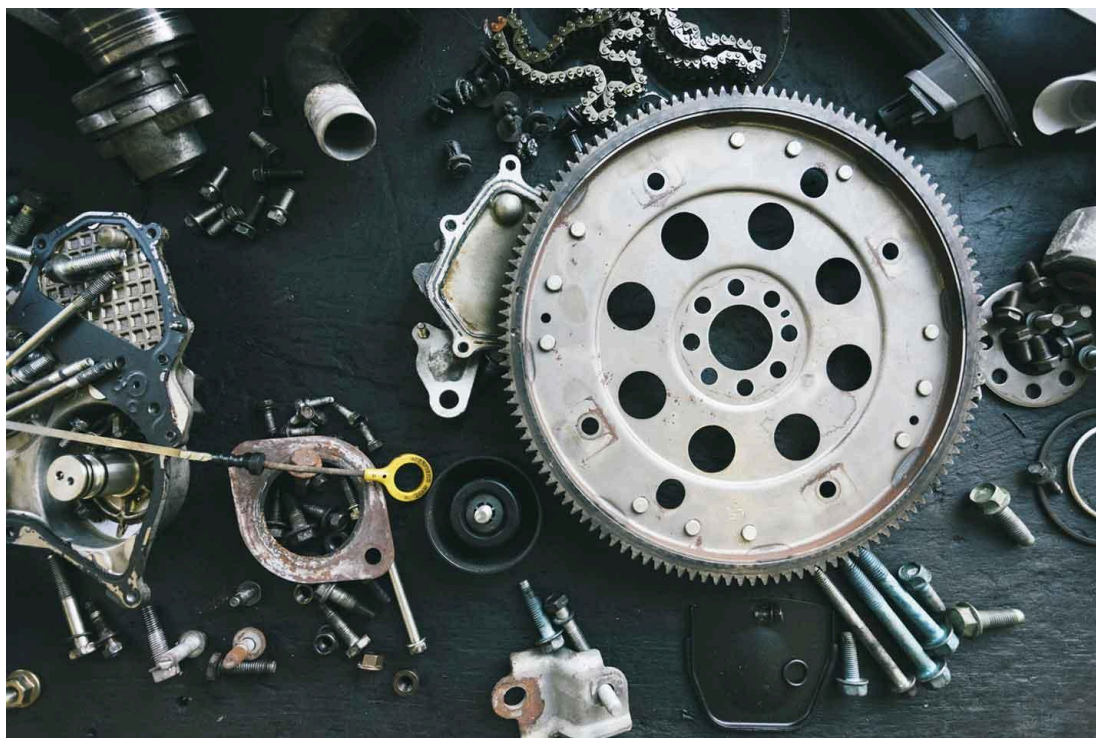


IONIC v8

Fab



Progresa
Una formación de futuro

ÍNDICE

Introducción	3
Preparación	3
Componente	3
Icono	3
Json	4
Lógica	4
Visualización	4
Explicación	6
Iconos	6
Colores	7



INTRODUCCIÓN

Son contenedores que tienen uno o más botones fab. Tienen que estar colocados en una posición fija que no hace scroll con el contenido. El fab debe tener un botón fab principal. El fab puede contener también una lista fab que aparece cuando se haga click en el botón fab principal. El botón principal puede contener varias listas fab y especificará el lado hacia el que se desplegarán.

La documentación del fab está en la página:

<https://ionicframework.com/docs/api/fab>

PREPARACIÓN

COMPONENTE

Para empezar a trabajar con el fab vamos a necesitar una página donde usarlo. Creamos la página en la **terminal**:

```
ionic generate page pages/fab --spec=false
```

ICONO

Añadimos en nuestro **app.component.ts** el icono que queremos usar para el menú en la lista de iconos(addIcons).

```
addIcons({grid, logoAppleAppstore, americanFootball, beaker,  
star, trash, shareSocial, caretForwardCircle, close, card,  
pin, car});
```

JSON

Añadimos en nuestro [menu.json](#) el nuevo elemento que queremos que aparezca tanto en el menú como en la página de inicio.

```
, {  
  "nombre": "Fab",  
  "ruta": "/fab",  
  "icono": "car"  
}
```

LÓGICA

Para poder generar un scroll en la página, vamos a crear un array de 100 elementos y así crear una lista de items con él en el html. En [fab.page.ts](#) creamos el array:

```
export class FabPage implements OnInit {  
  datos = Array(100);  
  constructor() { }  
  ngOnInit() {  
  }  
}
```

VISUALIZACIÓN

Y ahora ponemos el listado de items con el array. En [fab.page.html](#):

```
<app-header titulo="Fab"></app-header>
```

```
<ion-content class="ion-padding">
```

```
<ion-list>
```

```
  <ion-item *ngFor="let i of datos; let index=index">
```



```

        Item {{ index + 1 }}
    </ion-item>
</ion-list>

    <ion-fab          vertical="bottom"          horizontal="end"
slot="fixed">
    <ion-fab-button color="success">
        <ion-icon name="Add"></ion-icon>
    </ion-fab-button>
    <ion-fab-list side="top">
        <ion-fab-button color="facebook">
            <ion-icon name="logo-facebook"></ion-icon>
        </ion-fab-button>
        <ion-fab-button color="twitter">
            <ion-icon name="logo-twitter"></ion-icon>
        </ion-fab-button>
        <ion-fab-button color="youtube">
            <ion-icon name="logo-youtube"></ion-icon>
        </ion-fab-button>
    </ion-fab-list>
    <ion-fab-list side="start">
        <ion-fab-button color="twitter">
            <ion-icon name="logo-vimeo"></ion-icon>
        </ion-fab-button>
        <ion-fab-button color="youtube">
            <ion-icon name="logo-google"></ion-icon>
        </ion-fab-button>
        <ion-fab-button color="github">
            <ion-icon name="logo-github"></ion-icon>
        </ion-fab-button>
    </ion-fab-list>
</ion-fab>

</ion-content>

```



```

<ion-footer>
  <ion-toolbar color="danger">
    <ion-title class="ion-text-center">
      Este es el footer
    </ion-title>
  </ion-toolbar>
</ion-footer>

```

Explicación

```

    <ion-fab          vertical="bottom"          horizontal="end"
slot="fixed">
vertical="bottom"

```

Posición en el eje vertical. Puede ser "bottom", "center" o "top".

```
horizontal="end"
```

Posición en el eje horizontal. Puede ser "center", "end" o "start".

```
slot="fixed"
```

El botón estará fijo y no se moverá a pesar del scroll.

```
<ion-fab-list side="top">
```

Hacia dónde se va a abrir el menú al hacer click en el botón. Puede ser "start", "end", "top" o "bottom".

ICONOS

Antes de terminar tenemos que añadir todos los iconos que hemos usado para el fab, incluyendo el central. En [app.component.ts](#):

```

addIcons({grid, logoAppleAppstore, americanFootball, beaker,
star, trash, shareSocial, caretForwardCircle, close, card,
pin, car, logoFacebook, logoTwitter, logoTiktok,
logoInstagram, logoYoutube, logoGithub, add});

```

COLORES

Ahora para poder darle colores a los botones vamos a crear nuevos colores para nuestra aplicación. Primero tenemos que seleccionar un color, con sus tintas, sombras y brillos. Ionic nos proporciona un generador de colores.

<https://ionicframework.com/docs/theming/color-generator>

Ahí podremos trabajar con los colores y crear configuraciones. En este caso vamos a elegir un color parecido al rojo de Youtube y lo copiamos del código que genera. Una vez tenemos el color vamos a añadirlo a nuestra aplicación en el archivo **variables.scss** de la carpeta Theme. Tenemos que añadir por un lado el color y por otro las clases que utilizará Ionic.

```
:root{  
  /** Youtube **/  
  --ion-color-youtube: #ff0404;  
  --ion-color-youtube-rgb: 255,4,4;  
  --ion-color-youtube-contrast: #ffffff;  
  --ion-color-youtube-contrast-rgb: 255,255,255;  
  --ion-color-youtube-shade: #e00404;  
  --ion-color-youtube-tint: #ff1d1d;  
}  
  
.ion-color-youtube {  
  --ion-color-base: var(--ion-color-youtube) !important;  
    --ion-color-base-rgb:      var(--ion-color-youtube-rgb)  
!important;  
    --ion-color-contrast:      var(--ion-color-youtube-contrast)  
!important;  
                                --ion-color-contrast-rgb:  
var(--ion-color-youtube-contrast-rgb) !important;  
  --ion-color-shade: var(--ion-color-youtube-shade) !important;  
  --ion-color-tint: var(--ion-color-youtube-tint) !important;  
}
```