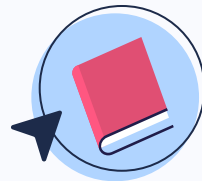


DuocUC

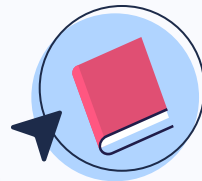


PGY4121: Ionic Crear App IOS, Android



Docente: @VivitaSol
v.pobletel@profesor.duoc.cl

Temas Puntuales



Revisaremos componentes de Ionic creando ejercicios simples para cada componente.



Componentes



Ion-action sheet

- Trabajaremos sobre el archivo action-sheet.page.html y action-sheet.page.ts.
- Navegue a www.ionicframework.com y seleccione: Ionic – action sheet
- Ingrese a action-sheet.page.html e inserte un botón

```
p > pages > action-sheet > action-sheet.page.html > ion-content
<ion-back-button
  text="Back"
  icon="buttonIcon">
</ion-back-button>
</ion-buttons>

</ion-toolbar>
</ion-header>

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

  <ion-button (click)="onClick()" expand="block">
    Mostrar Action - Sheet
  </ion-button>

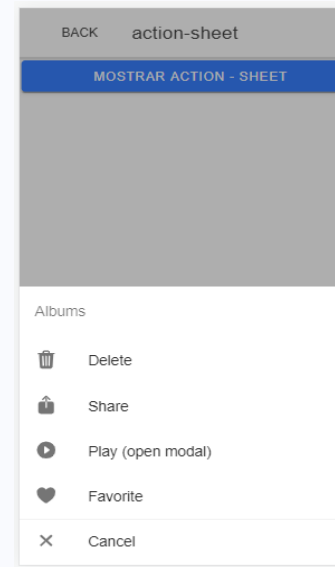
</ion-content>
```

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

```
<ion-button (click)="onClick()" expand="block">
  Mostrar Action - Sheet
</ion-button>
```

```
</ion-content>
```

- Recuerda utilizar los atajos, por ej: **lbutton y luego presiona la tecla tab**



Componentes

Ion-action sheet

- Ingrese al archivo page.ts de action-sheet e implementemos el método predeterminado que genera una ventana al hacer click.
- Acceda a la documentación oficial de este componente e inserte en este archivo el método `async presentActionSheet()` del apartado Usage



- Recuerda utilizar los atajos, por ej: **lbutton y luego presiona la tecla tab**

Usage

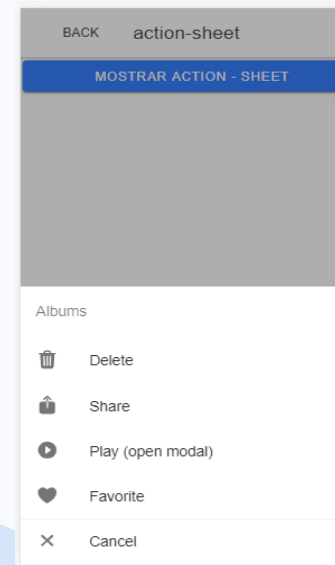
ANGULAR JAVASCRIPT REACT STENCIL VUE

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

@Component({
  selector: 'action-sheet-example',
  templateUrl: 'action-sheet-example.html',
  styleUrls: ['./action-sheet-example.css'],
})
export class ActionSheetExample {

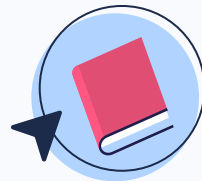
  constructor(public actionSheetController: ActionSheetController) {}

  async presentActionSheet() {
    const actionSheet = await this.actionSheetController.create({
      header: 'Albums',
      cssClass: 'my-custom-class',
      buttons: [{
        text: 'Delete',
        role: 'destructive',
        icon: 'trash',
        handler: () => {
          console.log('Delete clicked');
        }
      ]
    });
  }
}
```



Componentes

Ion-action sheet

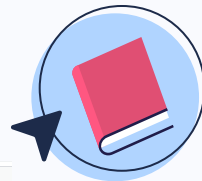


- ❑ En el archivo `..page.ts` de action sheet, debe importar:
`import { ActionSheetController } from '@ionic/angular';`
- ❑ En el constructor recibir como parámetro:
`constructor(private actionSheetController: ActionSheetController)`
- ❑ Implementamos el método `onClick() { }`
- ❑ Trasladamos el método `async presentActionSheet()` de la documentación
- ❑ Modificamos la forma de los icons implementando `outline`
- ❑ Creamos un class 'rojo' para el elemento 'Delete'
- ❑ Definimos un class 'rojo' en el archivo `'global.scss'`
- ❑ Propiedad `backdropDismiss:false` permite establecer que el menú contextual este activo al seleccionar una opción

▪ **Revise los archivos: `action-sheet.page.html`,
`action-sheet.page.ts` y `global.scss`.**

Componentes

Ion-action sheet



```
ion-sheet.page.html U • TS action-sheet.page.ts X global.scss M
action-sheet > TS action-sheet.page.ts > ActionSheetPage > presentActionSheet > [actionSheet]
import { Component, OnInit } from '@angular/core';
import { ActionSheetController } from '@ionic/angular';

@Component({
  selector: 'app-action-sheet',
  templateUrl: './action-sheet.page.html',
  styleUrls: ['./action-sheet.page.scss'],
})
export class ActionSheetPage implements OnInit {

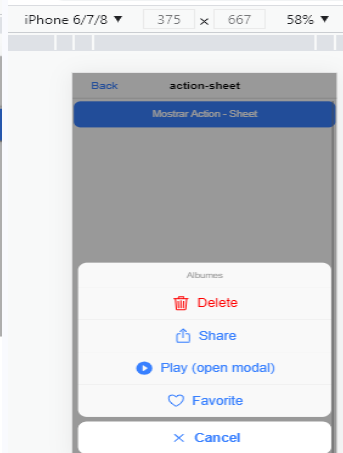
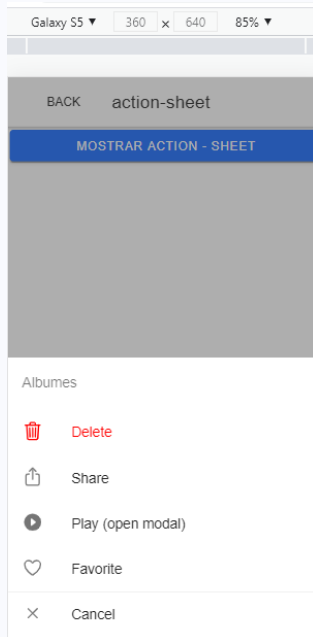
  constructor(private actionSheetController: ActionSheetController) {}

  ngOnInit() {}

  onClick(){
    this.presentActionSheet();
  }

  async presentActionSheet() {
    const actionSheet = await this.actionSheetController.create({
      header: 'Albumes',
      buttons: [{
        text: 'Delete',
        role: 'destructive',
        icon: 'trash-outline',
        cssClass: 'rojo',
        handler: () => {
          console.log('Delete clicked');
        }
      }, {
        text: 'Share',

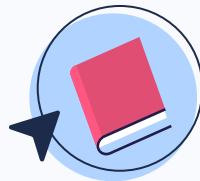
```



```
22 @import "~@ionic/angular/css/padding.css";
23 @import "~@ionic/angular/css/float-elements.css";
24 @import "~@ionic/angular/css/text-alignment.css";
25 @import "~@ionic/angular/css/text-transformation.css";
26 @import "~@ionic/angular/css/flex-utils.css";
27
28 .rojo, .rojo .sc-ion-action-sheet-md {
29   color: red !important;
30 }
31
```

Componentes

Ionic - alert



- ❑ Ingrese al archivo alert.page.html e inserte un button
- ❑ De la documentación oficial del sitio, inserte el siguiente método (apartado de Usage)

```
async presentAlert() {  
  const alert = await this.alertController.create({  
    header: 'Alert',  
    subHeader: 'Subtitle',  
    message: 'This is an alert message.',  
    buttons: ['OK']  
  });  
  
  await alert.present();  
}
```

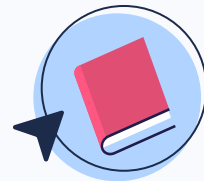
```
<ion-button (click)="presentAlert()" expand="block" >  
  Click me  
</ion-button>
```

```
<?xml version="1.0"?>  
<ion-alert>  
  <ion-header>  
    <ion-toolbar>  
      <ion-title>alert</ion-title>  
      <ion-buttons slot="start">  
        <ion-back-button  
          defaultHref="/"  
          text="Back"  
          icon="buttonIcon">  
        </ion-back-button>  
      </ion-buttons>  
    </ion-toolbar>  
  </ion-header>  
  <ion-content class="ion-padding">  
    <ion-button (click)="presentAlert()" expand="block" >  
      Click me  
    </ion-button>  
  </ion-content>  
</ion-alert>
```

Componentes

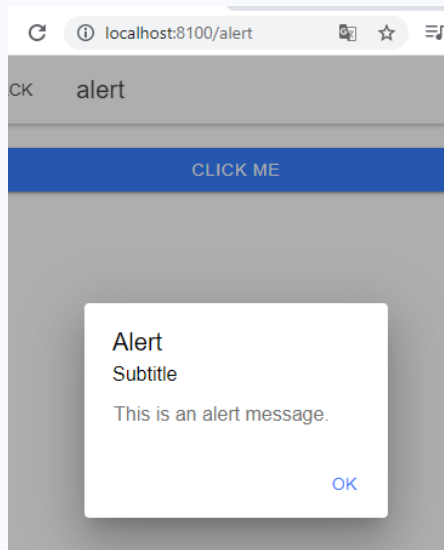
Ionic - alert

- ❑ Importe AlertController.
- ❑ Invoque el método en page.html.



```
<ion-button (click)="presentAlert()" expand="block" >  
    Click me  
</ion-button>
```

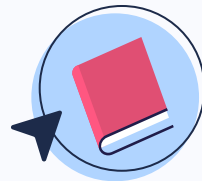
```
terminal  Help  • alert.page.ts - componentes - Visual Studio Code  
  
alert.page.html U  TS alert.page.ts U •  
c > app > pages > alert > TS alert.page.ts > AlertPage > presentAlert  
1 import { Component, OnInit } from '@angular/core';  
2 import { AlertController } from '@ionic/angular';  
3  
4 @Component({  
5   selector: 'app-alert',  
6   templateUrl: './alert.page.html',  
7   styleUrls: ['./alert.page.scss'],  
8 })  
9 export class AlertPage implements OnInit {  
10  
11   constructor(private alertController: AlertController) {}  
12  
13   ngOnInit() {  
14   }  
15  
16   async presentAlert() {  
17     const alert = await this.alertController.create({  
18       header: 'Alert',  
19       subHeader: 'Subtitle',  
20       message: 'This is an alert message.',  
21       buttons: ['OK']  
22     });  
23  
24     await alert.present();  
25   }  
26  
27 }
```



```
alert.page.html U x  TS alert.page.ts U •  
src > app > pages > alert > alert.page.html > ...  
1 <ion-header>  
2   <ion-toolbar>  
3     <ion-title>alert</ion-title>  
4  
5     <ion-buttons slot="start">  
6       <ion-back-button  
7         defaultHref="/"  
8         text="Back"  
9         icon="buttonIcon">  
10      </ion-back-button>  
11    </ion-buttons>  
12  
13  </ion-toolbar>  
14  
15 </ion-header>  
16  
17 <ion-content class="ion-padding">  
18  
19   <ion-button (click)="presentAlert()" expand="block" >  
20     Click me  
21   </ion-button>  
22 </ion-content>  
23
```


Componentes

Alert - multiplebuttons



- ❑ Ingrese al archivo `alert.page.html` e inserte un `button`
- ❑ De la documentación oficial del sitio, inserte el siguiente método (apartado de Usage)

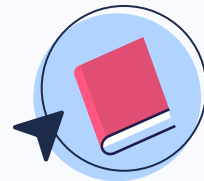
```
<ion-  
button (click)="presentAlertMultipleButtons()" expand="block"  
>  
    Click me MultipleButtons  
</ion-button>
```

```
async presentAlertMultipleButtons() {  
    const alert = await this.alertController.create  
(  
        {  
            header: 'Alert',  
            subHeader: 'Subtitle',  
            message: 'This is an alert message.',  
            buttons: ['Cancel', 'Open Modal', 'Delete']  
        });  
  
    await alert.present();  
}
```

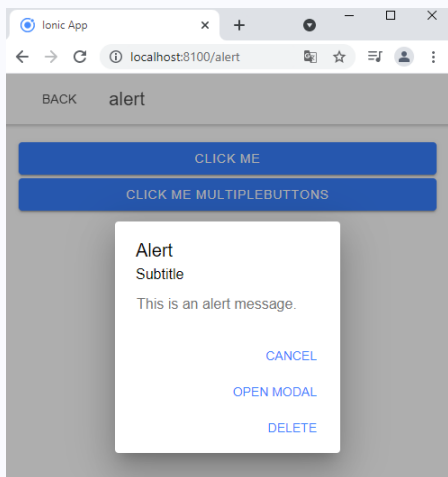
```
6 <ion-content class="ion-padding">  
7  
8     <ion-button (click)="presentAlert()" expand="block" >  
9         Click me  
10    </ion-button>  
11  
12    <ion-button (click)="presentAlertMultipleButtons()" expand="block">  
13        Click me MultipleButtons  
14    </ion-button>  
15  
16 </ion-content>
```

Componentes

Alert - multiplebuttons



- ❑ El resultado es un alert con distintas opciones.
- ❑ Para determinar cuál de los botones ha sido clickeado, realizamos la siguiente modificación:



```
async presentAlertMultipleButtons() {  
  const alert = await this.alertController.create({  
    header: 'Alert',  
    subHeader: 'Subtitle',  
    message: 'This is an alert message.',  
    buttons: [  
      {  
        text: 'Ok!',  
        handler:()=>{  
          console.log ('click en Ok!')  
        }  
      },  
      {  
        text: 'Cancel',  
        role: 'cancel',  
        cssClass: 'rojo'  
      }  
    ]  
  });  
  await alert.present();  
}
```

Componentes

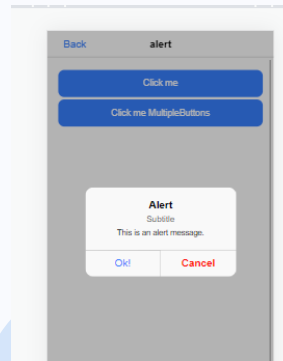
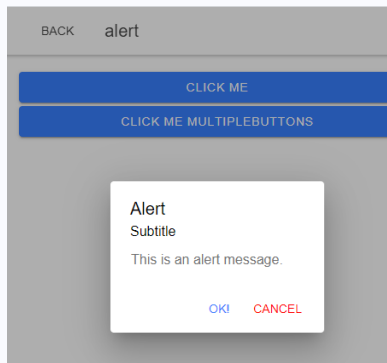
Alert - multiplebuttons



```
...page.html U • TS alert.page.ts x
pp > pages > alert > TS alert.page.ts > AlertPage > presentAlertMult
    await alert.present();
  }

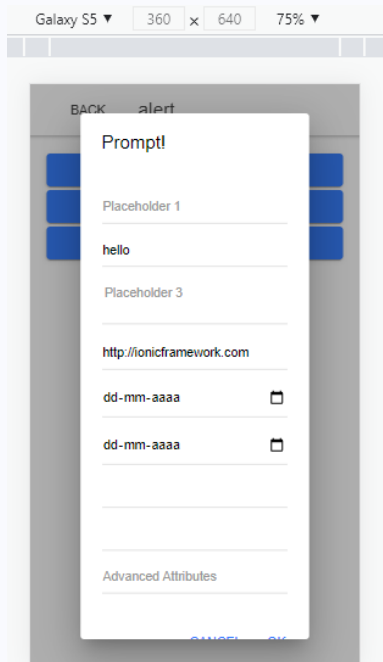
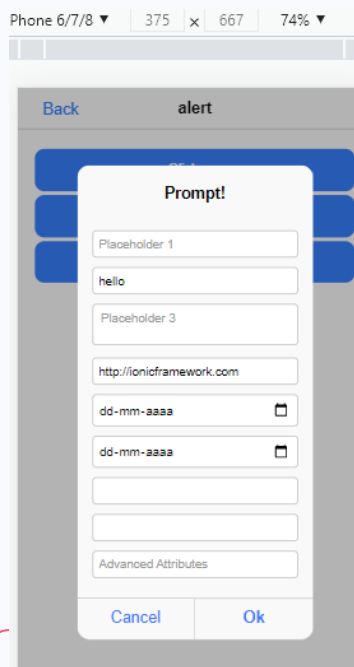
  async presentAlertMultipleButtons() {
    const alert = await this.alertController.create({
      header: 'Alert',
      subHeader: 'Subtitle',
      message: 'This is an alert message.',
      buttons: [
        {
          text: 'Ok!',
          handler: ()=>{
            console.log ('click en Ok!')
          }
        },
        {
          text: 'Cancel',
          role: 'cancel',
          cssClass: 'rojo'
        }
      ]
    });
    await alert.present();
  }
}
```

- ❑ En la propiedad buttons definimos dos objetos con sus propiedades.
- ❑ La propiedad handler() activa una acción en console.log.
- ❑ En el objeto Cancel, invocamos la class 'rojo' definido en el global.scss.



Componentes

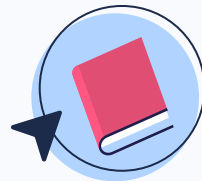
Alert - prompt



- ❑ Revise la documentación ionic.
- ❑ Inserte el código del método `presentAlertPrompt()`.
- ❑ Invoque el método en un nuevo botón dentro de `page.html` (en el componente `alert`)
- ❑ Ejecute el servidor, visualice la distribución de los elementos para Android y para ios.

Componentes

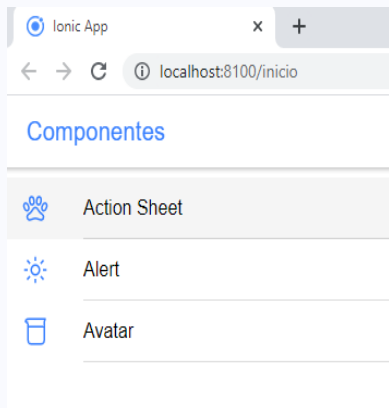
Ion-avatar



```
Microsoft Windows [Versión 10.0.19042.1110]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Vivitasol\Desktop\Ionic\componentes>ionic g page pages/avatar_
```

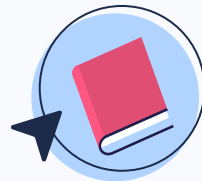
```
componentes : Componente[] = [
  {
    icon: 'paw-outline',
    name: 'Action Sheet',
    redirectTo: '/action-sheet'
  },
  {
    icon: 'sunny-outline',
    name: 'Alert',
    redirectTo: '/alert'
  },
  {
    icon: 'beaker-outline',
    name: 'Avatar',
    redirectTo: '/avatar'
  },
];
constructor() {}
ngOnInit() {
}
```



- ❑ Crearemos una nueva página dentro de pages.
- ❑ Al momento de crear /avatar se agrega a la app.routing.module.ts.
- ❑ Agregaremos al archivo inicio.page.ts el nuevo componente.
- ❑ Recuerde que redirectTo es la dirección registrada en app.routing.module.ts.

Componentes

Ion-avatar

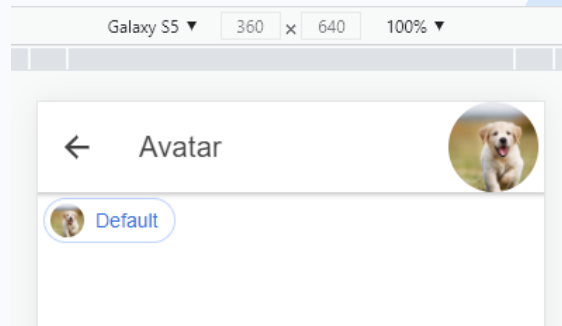
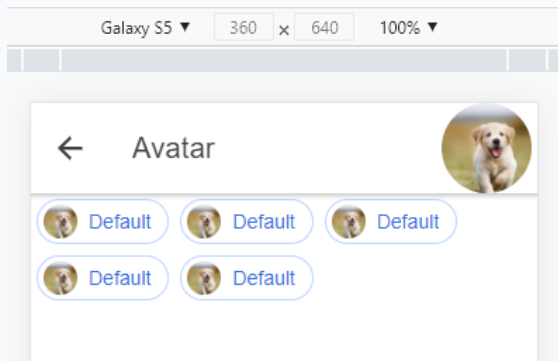


```
<ion-content>
  <ion-chip color="primary" outline="true">
    <ion-avatar>
      
    </ion-avatar>
    <ion-label>Default</ion-label>
  </ion-chip>
</ion-content>
```

- ❑ Accedemos a avatar.page.html.
- ❑ Insertamos un ion-chip en ion-content.
- ❑ Insertamos un ion-avatar en un ion-chip.

- ❑ Podemos generar un *ngFor

```
<ion-content>
  <ion-chip color="primary" outline="true"
    *ngFor="let item of [1,1,1,1,1]">
    <ion-avatar>
      
    </ion-avatar>
    <ion-label>Default</ion-label>
  </ion-chip>
</ion-content>
```



Componentes

Ion-avatar

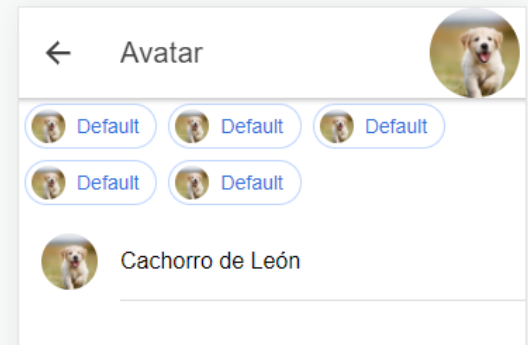
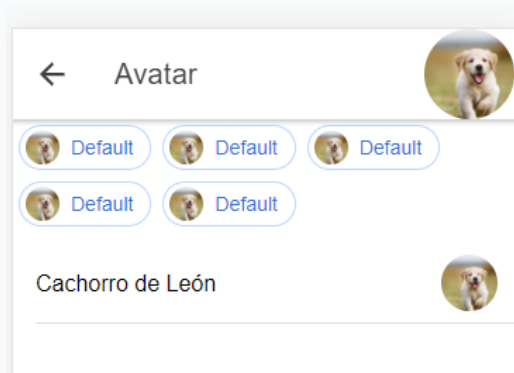


```
o.page.ts U  avatar.page.html U x cachorro.jpg (read-onl
op > pages > avatar > avatar.page.html > ...

<ion-content>
  <ion-chip color="primary" outline="true"
    *ngFor="let item of [1,1,1,1,1]">
    <ion-avatar>
      
    </ion-avatar>
    <ion-label>Default</ion-label>
  </ion-chip>

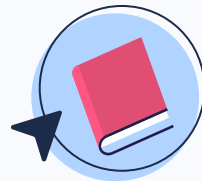
  <ion-list>
    <ion-item>
      <ion-avatar slot="end">
        
      </ion-avatar>
      <ion-label>Cachorro de León</ion-label>
    </ion-item>
  </ion-list>
</ion-content>
```

- ❑ Insertamos un ion-List
- ❑ Luego insertamos un ion-Item.
- ❑ En cada ítem insertamos un ion-Avatar.



Componentes

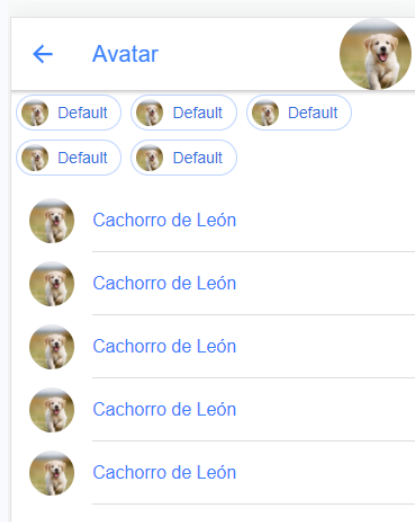
Ion-avatar



```
<ion-content>
  <ion-chip color="primary" outline="true"
    *ngFor="let item of [1,1,1,1,1]">
    <ion-avatar>
      
    </ion-avatar>
    <ion-label>Default</ion-label>
  </ion-chip>

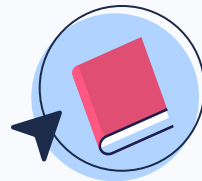
  <ion-list>
    <ion-item
      *ngFor="let item of [1,1,1,1,1]">
      <ion-avatar slot="start">
        
      </ion-avatar>
      <ion-label color="primary">Cachorro de León</ion-label>
    </ion-item>
  </ion-list>
</ion-content>
```

- ❑ Insertamos un ngFor
- ❑ Cambiamos a color primary algunos elementos.



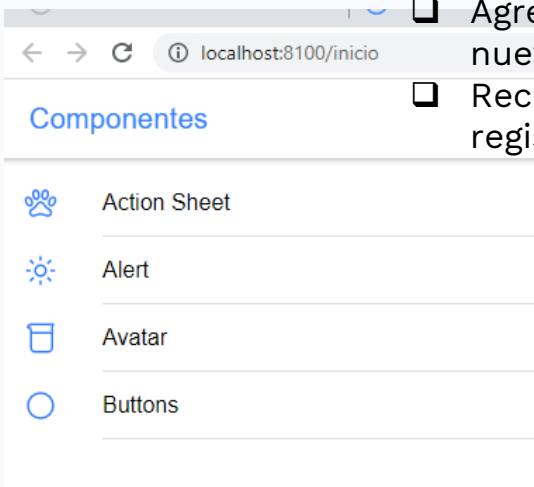
Componentes

Ion-button



```
C:\Users\Vivitasol\Desktop\Ionic\componentes>ionic g page pages/botones
> ng.cmd generate page pages/botones --project=app
CREATE src/app/pages/botones/botones-routing.module.ts (351 bytes)
CREATE src/app/pages/botones/botones.module.ts (479 bytes)
```

```
app > pages > inicio > TS inicio.page.ts > InicioPage
{
  icon: 'sunny-outline',
  name: 'Alert',
  redirectTo: '/alert'
},
{
  icon: 'beaker-outline',
  name: 'Avatar',
  redirectTo: '/avatar'
},
{
  icon: 'radio-button-off-outline',
  name: 'Buttons',
  redirectTo: '/botones'
},
];
```



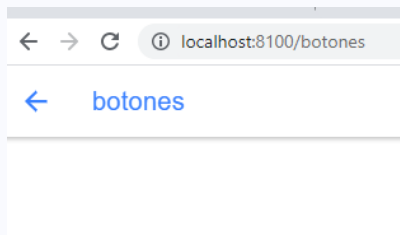
- ☐ Crearemos una nueva página dentro de pages.
- ☐ Al momento de crear /botones se agrega a la app.routing.module.ts.
- ☐ Agregaremos al archivo inicio.page.ts el nuevo componente.
- ☐ Recuerde que redirectTo es la dirección registrada en app.routing.module.ts.

Componentes

Ion-button



```
app > pages > botones > botones.page.html > ion-content
<ion-header>
  <ion-toolbar>
    <ion-buttons slot="start">
      <ion-back-button defaultHref="/" color="primary"></ion-back-button>
    </ion-buttons>
    <ion-title color="primary">botones</ion-title>
  </ion-toolbar>
</ion-header>
```



- ❑ Accedemos a botones.page.html.
- ❑ Insertamos ion-buttons.
- ❑ Luego un ion-back-button

Usage

ANGULAR JAVASCRIPT REACT STENCIL VUE

```
<!-- Default -->
<ion-button>Default</ion-button>

<!-- Anchor -->
<ion-button href="#">Anchor</ion-button>

<!-- Colors -->
<ion-button color="primary">Primary</ion-button>
<ion-button color="secondary">Secondary</ion-button>
<ion-button color="tertiary">Tertiary</ion-button>
<ion-button color="success">Success</ion-button>
<ion-button color="warning">Warning</ion-button>
<ion-button color="danger">Danger</ion-button>
<ion-button color="light">Light</ion-button>
<ion-button color="medium">Medium</ion-button>
<ion-button color="dark">Dark</ion-button>
```

Componentes

Ion-button



```
.page.ts U  botones.page.html U x  avatar.page.html U
p > pages > botones > botones.page.html > ion-content.ion-padding
</ion-header>

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

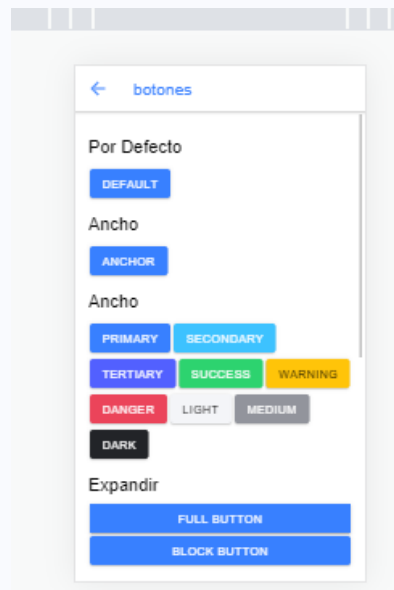
  <h3>Por Defecto</h3>
  <ion-button>Default</ion-button>

  <h3>Ancho</h3>
  <ion-button href="#">Anchor</ion-button>

  <h3>Ancho</h3>
  <ion-button color="primary">Primary</ion-button>
  <ion-button color="secondary">Secondary</ion-button>
  <ion-button color="tertiary">Tertiary</ion-button>
  <ion-button color="success">Success</ion-button>
  <ion-button color="warning">Warning</ion-button>
  <ion-button color="danger">Danger</ion-button>
  <ion-button color="light">Light</ion-button>
  <ion-button color="medium">Medium</ion-button>
  <ion-button color="dark">Dark</ion-button>

  <h3>Expandir</h3>
  <ion-button expand="full">Full Button</ion-button>
  <ion-button expand="block">Block Button</ion-button>
```

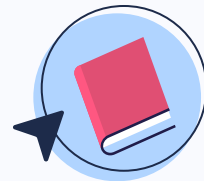
- ❑ Insertemos la codificación de la documentación oficial de ion-button.



Docente: @VivitaSol
v.pobletel@profesor.duoc.cl

Componentes

Ion-button



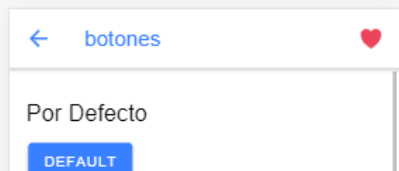
```
io.page.ts U  botones.page.html U  TS botones.page.ts U  avatar.page.html U
app > pages > botones > botones.page.html > ion-header
<ion-header>
  <ion-toolbar>
    <ion-buttons slot="start">
      <ion-back-button defaultHref="/" color="primary"></ion-back-button>
    </ion-buttons>
    <ion-title color="primary">botones</ion-title>

    <ion-buttons slot="end">
      <ion-button (click)="onClick()" color="danger">
        <ion-icon slot="icon-only"
          [name]=" favorito ? 'heart': 'heart-outline' "></ion-icon>
      </ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>
```

- ❑ Insertaremos un botón en el header de tipo corazón.
- ❑ `[name]=" favorito ? 'heart': 'heart-outline' ">`
- ❑ Expresión angular que evalúa el valor de 'favorito'
- ❑ En el archivo botones.page.ts, implementamos el método `onClick()`

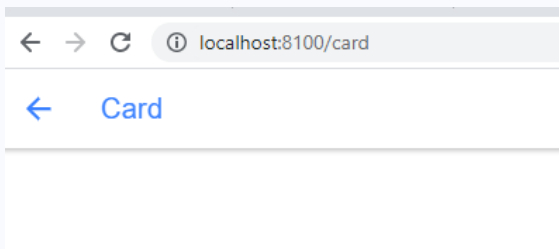
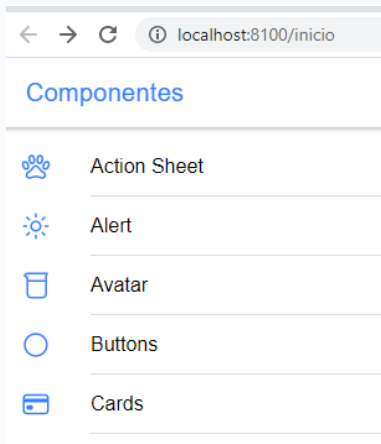
```
export class BotonesPage implements OnInit {
  favorito: boolean = false;
  constructor() { }
  ngOnInit() {
  }

  onClick(){
    this.favorito = !this.favorito;
  }
}
```



Componentes

Ion-card

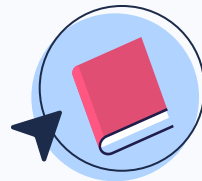


- ❑ Crearemos una nueva página dentro de pages:
ionic g page pages/card
- ❑ Al momento de crear /card se agrega a la app.routing.module.ts.
- ❑ Agregaremos al archivo inicio.page.ts el nuevo componente.
- ❑ Recuerde que redirectTo es la dirección registrada en app.routing.module.ts.
- ❑ Implementamos icon-buttons en card.page.html para volver al inicio.

```
<ion-toolbar>
  <ion-buttons slot="start" color="primary">
    <ion-back-button defaultHref="/" color="primary"></ion-back-button>
  </ion-buttons>
  <ion-title color="primary">Card</ion-title>
</ion-toolbar>
```

Componentes

Ion-card



```
<ion-card>
  <ion-card-header>
    <ion-card-subtitle>Card Subtitle</ion-card-subtitle>
    <ion-card-title>Card Title</ion-card-title>
  </ion-card-header>

  <ion-card-content>
    Keep close to Nature's heart... and break clear away, once in awhile,
    and climb a mountain or spend a week in the woods. Wash your spirit clean.
  </ion-card-content>
</ion-card>
```

```
<ion-card>
  <ion-item color="primary">
    <ion-icon slot="start" name="pin-outline"></ion-icon>
    <ion-label>Marcador</ion-label>
  </ion-item>

  <ion-card-content>
    Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quo qui fugit possimus perf
  </ion-card-content>
</ion-card>
```

- ❑ Insertaremos codificación de la documentación oficial.
- ❑ Insertaremos también un ion-icon y un ion-label en ion-card.

Card Subtitle

Card Title

Keep close to Nature's heart... and break clear away, once in awhile, and climb a mountain or spend a week in the woods. Wash your spirit clean.

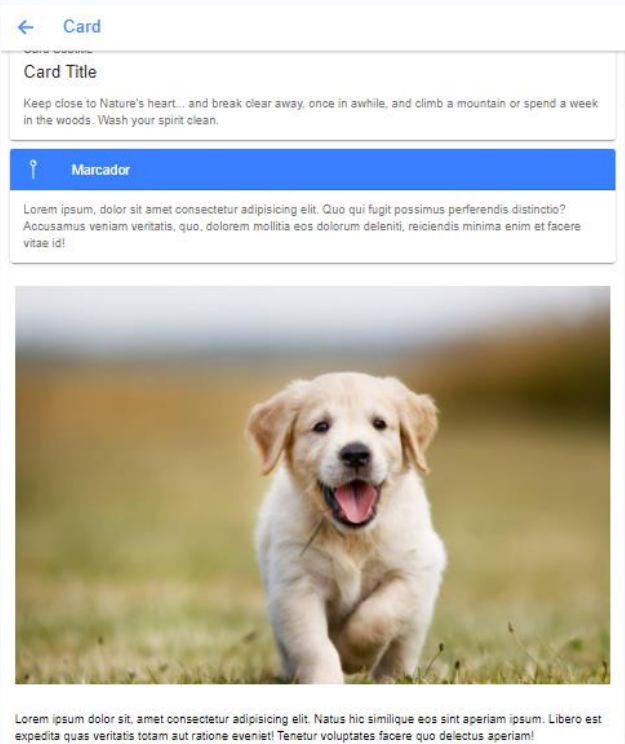


Marcador

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quo qui fugit possimus perferendis distinctio? Accusamus veniam veritatis, quo, dolorem mollitia eos dolorum deleniti, reiciendis minima enim et facere vitae id!

Componentes

Ion-card



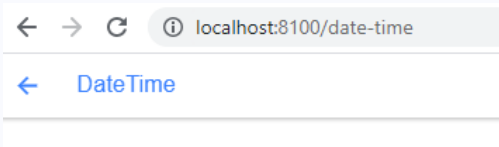
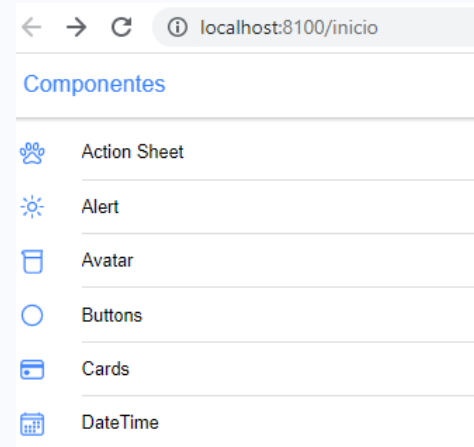
- ❑ Insertaremos ahora una imagen en un elemento ion-card.

```
<ion-card>
  
  <ion-card-content>
    Lorem ipsum dolor sit, amet consectetur adipisicing elit. Natus hic similique eos sin
  </ion-card-content>
</ion-card>
</ion-content>
```

- ❑ Actividad: Revise la documentación oficial e implemente un page para Checkbox como los elementos anteriores.

Componentes

Ion-date-time

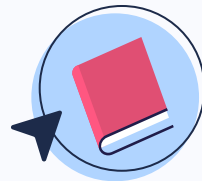


- ❑ Crearemos una nueva página dentro de pages:
ionic g page pages/dateTime
- ❑ Al momento de crear /dateTime se agrega a la app.routing.module.ts.
- ❑ Agregaremos al archivo inicio.page.ts el nuevo componente.
- ❑ Recuerde que redirectTo es la dirección registrada en app.routing.module.ts.
- ❑ Implementamos icon-buttons en card.page.html para volver al inicio.

```
<ion-toolbar>
  <ion-buttons slot="start" color="primary">
    <ion-back-button defaultHref="/" color="primary"></ion-back-
button>
  </ion-buttons>
  <ion-title color="primary">DateTime</ion-title>
</ion-toolbar>
```


Componentes

Ion-date-time



```
o.page.ts U  date-time.page.html U  TS date-time.page.ts U
pages > date-time > date-time.page.html > ion-content > ion-list > ion-item >
</ion-header>

<ion-content>
  <ion-list>

    <ion-list-header>
      <ion-label>Datos de Nacimiento</ion-label>
    </ion-list-header>

    <ion-item>
      <ion-label>Fecha de Nacimiento </ion-label>
      <ion-datetime value="fechaNaci.ToISOString()"
        display-format="DD MMMM YYYY"></ion-datetime>
    </ion-item>
```

```
export class DateTimePage implements OnInit {

  fechaNaci: Date = new Date();

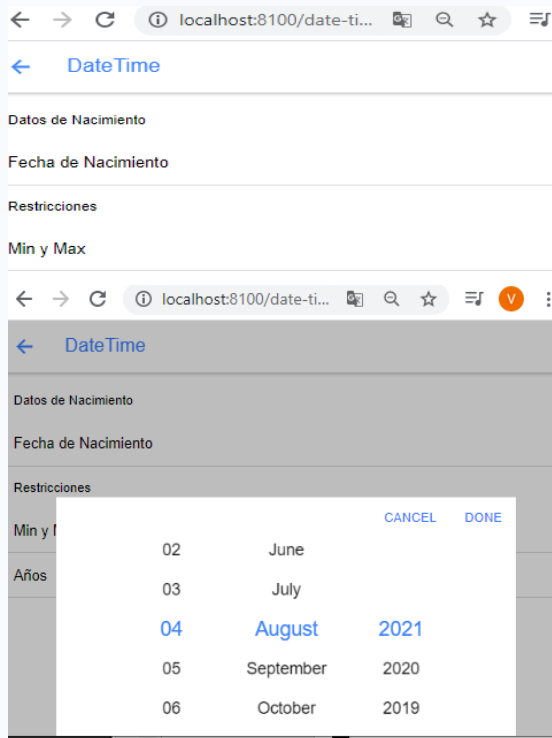
  constructor() { }
```

- ❑ Primero fijaremos la fecha actual para crear un calendario, en el archivo date-time.page.ts.
- ❑ Definimos un elemento de tipo Date().

- ❑ Luego accedemos a date-time.page.html e insertamos un elemento de tipo date-time.
- ❑ Implementamos el método ToISOString() y agregamos un formato por defecto.

Componentes

Ion-date-time



- ❑ Implementamos parámetros como max y min.

```
<ion-item>
  <ion-label>Min y Max</ion-label>
  <ion-datetime display-format="DD MMMM YYYY"
    min="2015-01-01"
    max="2025-12-31"></ion-datetime>
</ion-item>
```

- ❑ Modificamos parámetro para que se visualice solo el año.

```
<ion-item>
  <ion-label>Años</ion-label>
  <ion-datetime display-format="YYYY"
    min="2015"
    max="2025"></ion-datetime>
</ion-item>
```

¡Gracias!



¿Alguna pregunta?

- v.pobletel@profesor.duoc.cl



Docente: @VivitaSol
v.pobletel@profesor.duoc.cl