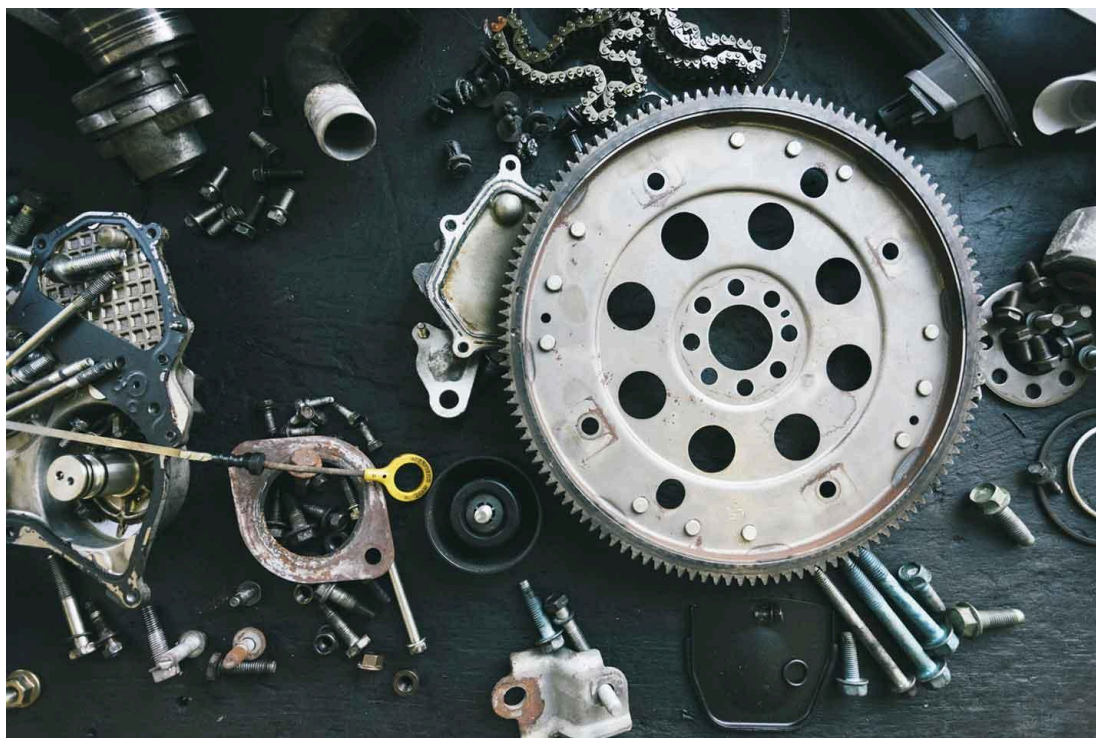


# IONIC v8

*Alert*



**Progresa**  
Una formación de futuro

# ÍNDICE

<b>Introducción</b>	<b>3</b>
<b>Ejemplo</b>	<b>3</b>
Visualización	3
Explicación	4
Lógica del componente	5
Explicación	8

# INTRODUCCIÓN

---

Un Alert es un diálogo que se muestra al usuario con información o recolecta información del usuario usando inputs. Un alert aparece en la parte superior de la aplicación y tiene que ser manualmente cerrado por el usuario antes de poder seguir utilizando la aplicación. Puede tener opcionalmente cabecera, subcabecera y un mensaje en forma de header, subHeader y message.

La documentación del alert está en la página:

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

## EJEMPLO

### VISUALIZACIÓN

---

En primer lugar vamos a crear dos botones que nos ejecutarán el alert al pulsarlos. En la página [alert.page.html](#):

```
<!-- Si no tenemos valor en título ponemos Alert page, y si si tenemos valor lo ponemos -->  
<app-header [titulo]="titulo" || 'Alert Page' "></app-header>
```

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

```
<h2>Alert</h2>
```

```
  <ion-button color="primary"  
    expand="full"  
    id="mostrar-alert">Mostrar Alert  
</ion-button>
```

```
<ion-alert  
  trigger="mostrar-alert"  
  header="Título del alert"
```



```

subHeader="Subtítulo"
message="Esto es una alerta"
[buttons]="botonesAlert"></ion-alert>

```

```

<h2>Alert con Input</h2>
<ion-button color="secondary"
    expand="full"
    id="mostrar-input">Mostrar Input
</ion-button>

```

```

<ion-alert
    trigger="mostrar-input"
    header="Alert con Input"
    subHeader="Inserte el nombre"
    [inputs]="inputs"
    [buttons]="botonesInput"></ion-alert>

```

```

<h2>Alert con Radiobutton</h2>
<ion-button expand="block"
    id="mostrar-radio">Radio</ion-button>
<ion-alert
    trigger="mostrar-radio"
    header="Selecciona color"
    [buttons]="radioButtons"
    [inputs]="radioInputs"
></ion-alert>
</ion-content>

```

### Explicación

```
trigger="mostrar-alert"
```

Define el id del elemento que lanza el alert.

```
[buttons]="botonesAlert"
```

Array dónde están definidos los botones del alert.

```
[inputs]="inputs"
```

Array dónde están definidos los inputs del alert.

## LÓGICA DEL COMPONENTE

Y ahora vamos a crear la variable título y ver la lógica de estas dos funciones en [alert.page.ts](#):

```
import {Component, OnInit} from '@angular/core';
import {CommonModule} from '@angular/common';
import {FormsModule} from '@angular/forms';
import {IonAlert, IonButton, IonContent} from
'@ionic/angular/standalone';
import {HeaderComponent} from
"../../components/header/header.component";

@Component({
  selector: 'app-alert',
  templateUrl: './alert.page.html',
  styleUrls: ['./alert.page.scss'],
  standalone: true,
  imports: [IonContent, CommonModule, FormsModule,
HeaderComponent, IonButton, IonAlert]
})
export class AlertPage implements OnInit {
  titulo!: string;
  botonesAlert = [
    {
      text: 'Cancel',
      role: 'cancel',

```

```

    cssClass: 'secondary',
    handler: () => {
        // Qué tenemos que hacer cuando pulsen el botón
        console.log('Cancelar');
    }
}, {
    text: 'Ok',
    handler: () => {
        console.log('Botón OK');
    }
}
];
inputs = [
    {
        name: 'nombre',
        type: 'text', // Tipo de input, como en HTML
        placeholder: 'Placeholder'
    }
];
botonesInput = [
    {
        text: 'Cancelar',
        role: 'cancel',
        handler: () => {
            console.log('Confirmado Cancelar');
        }
    },
    {
        text: 'Ok',
        handler: (data: any) => {
            console.log('Confirmado Ok y datos', data);
            // Guardamos en nuestra variable de la clase
            // título el nombre guardado en el input
            this.titulo = data.nombre;
        }
    }
];

```



```

    }
  }
];
public radioButtons = [ {
  text: 'Cancelar',
  role: 'cancel',
  handler: () => {
    console.log('Confirmado Cancelar');
  }
},
{
  text: 'Ok',
  handler: (data: any) => {
    console.log('Confirmado Ok y color: ', data);
  }
}
];
public radioInputs = [
  {
    label: 'Red',
    type: 'radio',
    value: 'red',
  },
  {
    label: 'Blue',
    type: 'radio',
    value: 'blue',
  },
  {
    label: 'Green',
    type: 'radio',
    value: 'green',
  },
];

```

```
constructor() {  
  }  
  ngOnInit() {  
  }  
}
```

## Explicación

**subHeader:** 'Subtítulo'

Es la subcabecera de nuestro alert.

**message:** 'Esto es una alerta'

El mensaje que saldrá en el alert.

**buttons:** [{

Es el array de botones del alert. En nuestro caso tendrá 2: OK y Cancelar

**inputs:** [ {

Es un array de elementos de entrada o inputs y en el representaremos los campos de texto de entrada que tendrá nuestro alert.

**name:** 'nombre',

El nombre del input.

**type:** 'text',

El tipo del input.

**placeholder:** 'Placeholder'

El texto que aparece en el cuadro de texto por defecto.

**handler:** ( data ) => {

Es el manejador para ver qué hacer al pulsar ese botón. Podemos pasarle la información recogida en el alert, en este caso la recogida por los inputs.