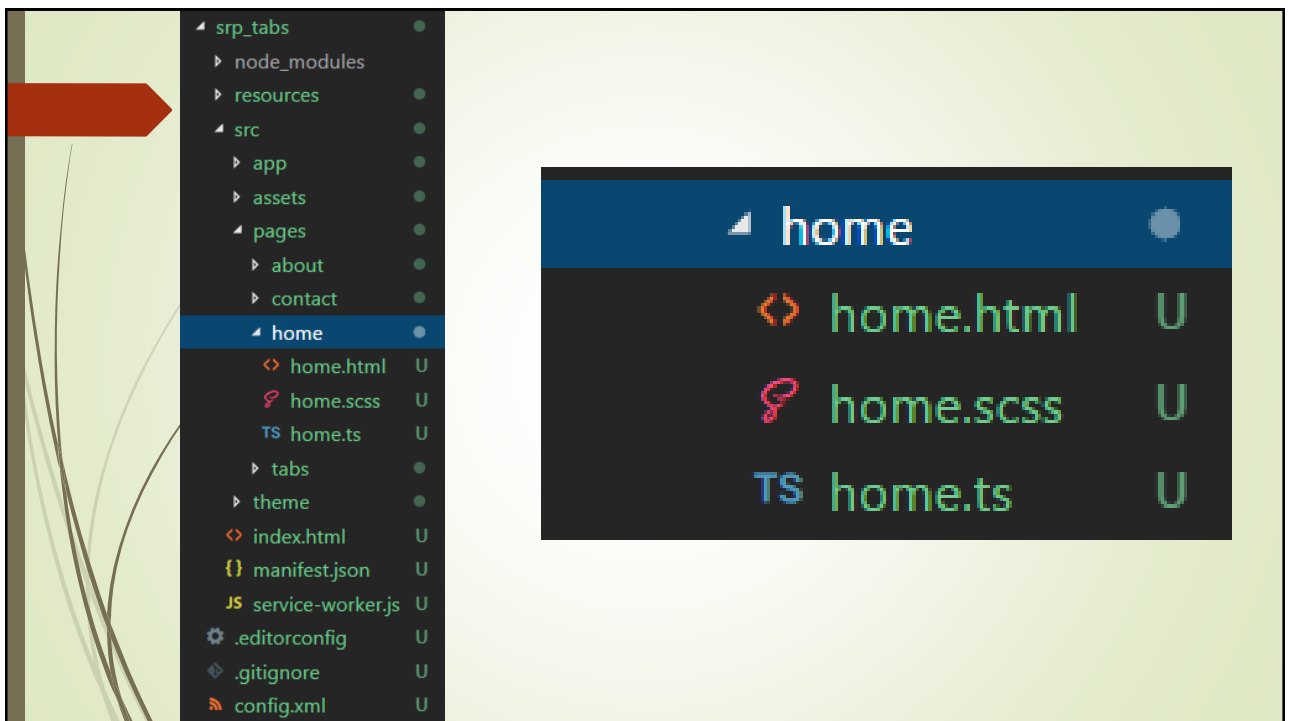


Personalización de aplicativos móviles

Ing. Silvia Rodríguez Paredes





home.scss

- Contiene el archivo sass donde podremos modificar el estilo de los componentes de la página.



home.ts

- Archivo typescript que contiene el controlador de la página, donde se define el comportamiento de la misma.

Ejemplo. la función con la lógica a ejecutarse cuando se pulse sobre un botón de la página etc.



home.html

- Contiene la plantilla html de la página.

```
<ion-header>
  <ion-navbar>
    <ion-title>Home</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <h2>Welcome to Ionic!</h2>
  <p>
    This starter project comes with simple tabs-based layout for apps
    that are going to primarily use a Tabbed UI.
  </p>
  <p>
    Take a look at the <code>src/pages/</code> directory to add or change tabs,
    update any existing page or create new pages.
  </p>
</ion-content>
```

<https://ionicframework.com/docs/v2/components/>

Es seguro | <https://ionicframework.com/docs/components/>

docs Framework Products Developers Businesses Support Ecosystem Sign up

HOME

INTRO

COMPONENTS

Overview

Action Sheets

Alerts

Badges

Buttons

Cards

Checkbox

Components

Ionic esta basado en angular.
Si los componentes no son los suficiente se pueden crear componentes propios, personalizados de la misma manera que en angular.

you to
cluding
looks like
API docs for

```
<ion-header>
  <ion-navbar>
    <ion-title>Home</ion-title>
  </ion-navbar>
</ion-header>
```

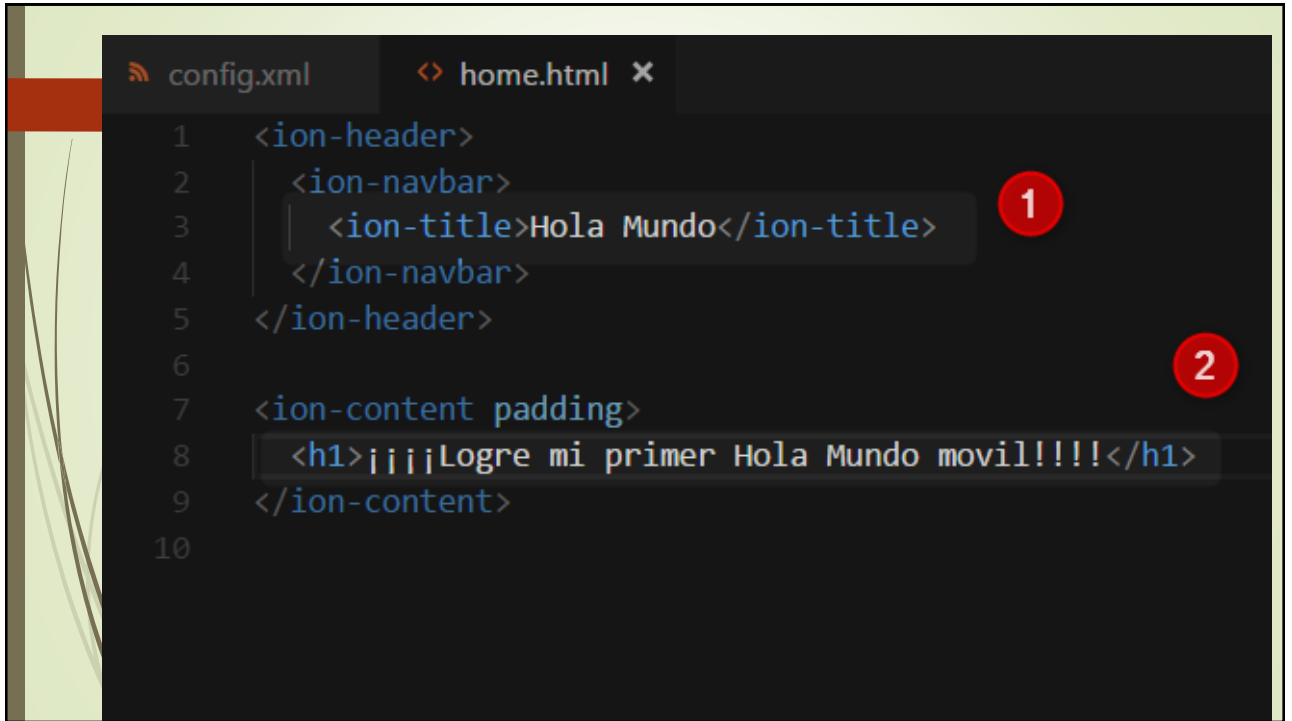
Título

Barra de navegacion

Cabecera

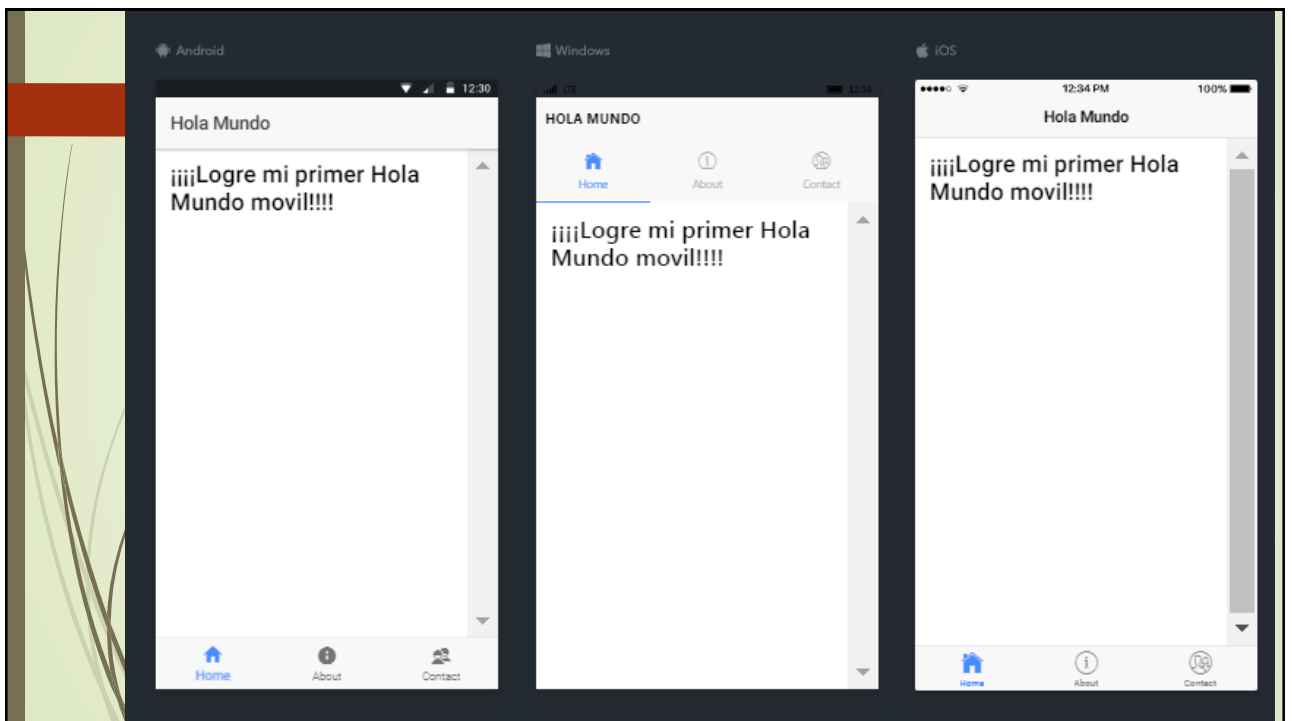
```
<ion-content padding>
  <h2>Welcome to Ionic!</h2>
  <p>
    This starter project comes with simple tabs-based layout for apps
    that are going to primarily use a Tabbed UI.
  </p>
  <p>
    Take a look at the <code>src/pages/</code> directory to add or change tabs,
    update any existing page or create new pages.
  </p>
</ion-content>
```

Contenido



```
config.xml  home.html x
1  <ion-header>
2    <ion-navbar>
3      <ion-title>Hola Mundo</ion-title>
4    </ion-navbar>
5  </ion-header>
6
7  <ion-content padding>
8    <h1>¡¡¡¡Logre mi primer Hola Mundo movil!!!!</h1>
9  </ion-content>
10
```

The image shows a code editor with two tabs: 'config.xml' and 'home.html'. The 'home.html' tab is active, showing HTML code for an Ionic application. Two red circles with white numbers are overlaid on the code: circle '1' is next to the '<ion-title>Hola Mundo</ion-title>' line, and circle '2' is next to the '<h1>¡¡¡¡Logre mi primer Hola Mundo movil!!!!</h1>' line.



INICIANDO LA PROGRAMACION !!!



Que es Data Binding ?

- **Data binding** es un mecanismo mediante el cual podemos enlazar los elementos de la interfaz de usuario con los objetos que contienen la información a mostrar. El caso más típico de **data binding** es el enlazar un control de la interfaz de usuario con un valor o registro de una base de datos.

Crear una aplicación en blanco

`ionic start srodriguez_acierta blank`

Entramos al proyecto

`Cd srodriguez_acierta`

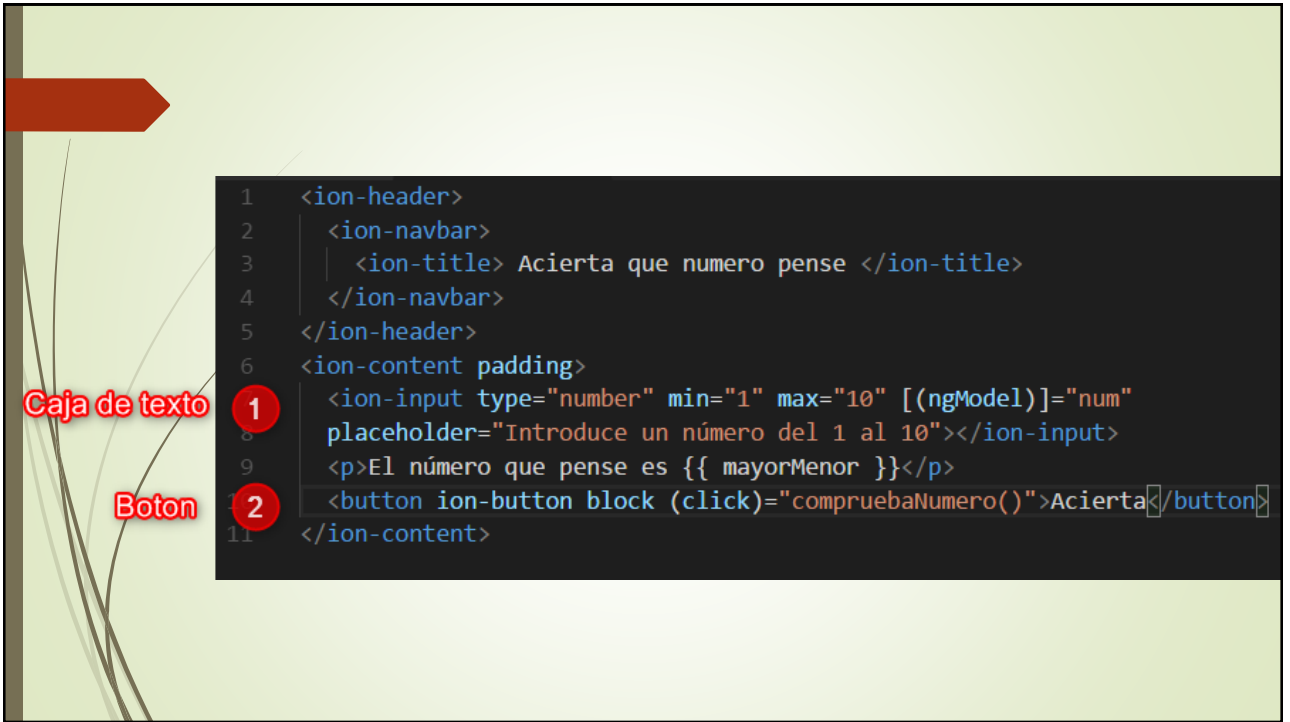
```
<ion-header>
```

```
<ion-navbar>
```

```

1  <ion-header>
2    <ion-navbar>
3      <ion-title> Acierta que numero pense </ion-title>
4    </ion-navbar>
5  </ion-header>
6  <ion-content padding>
7    <ion-input type="number" min="1" max="10" [(ngModel)]= "num"
8    placeholder="Introduce un número del 1 al 10"></ion-input>
9    <p>El número que pense es {{ mayorMenor }}</p>
10   <button ion-button block (click)="compruebaNumero()">Acierta</button>
11 </ion-content>
```

Alt + shift + F



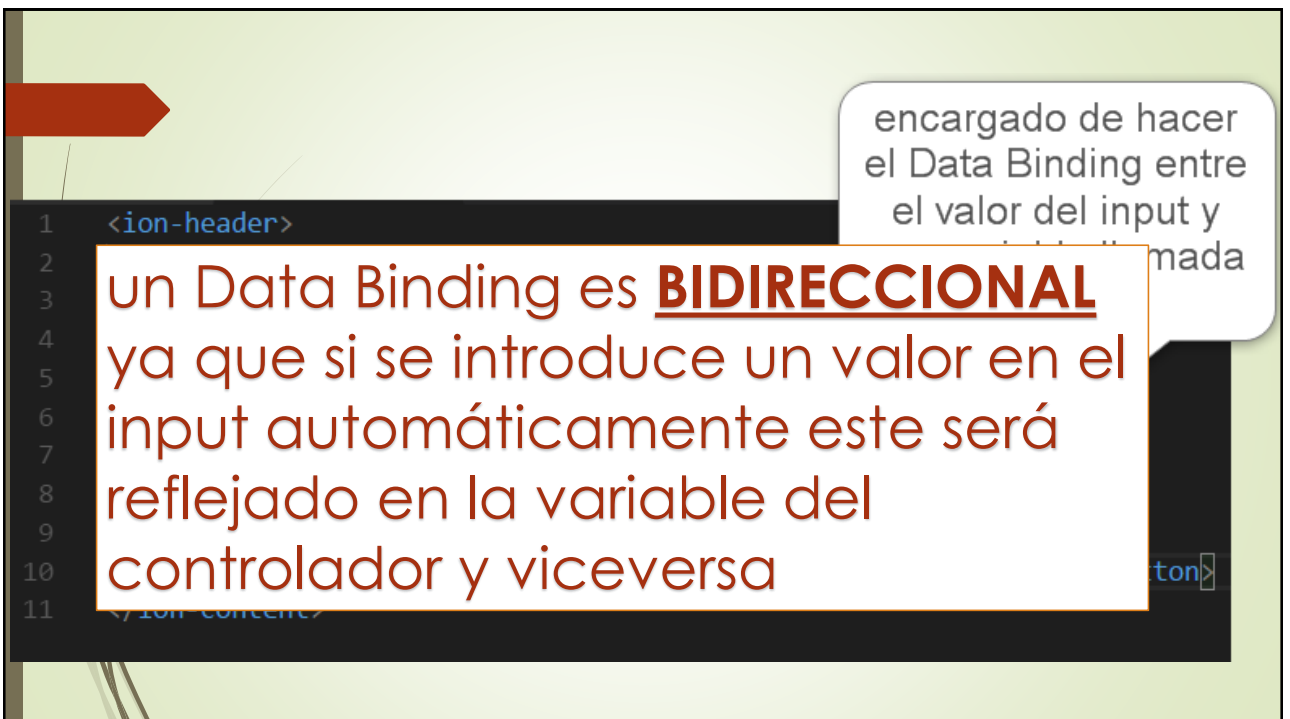
Caja de texto

Boton

```

1 <ion-header>
2   <ion-navbar>
3     <ion-title> Acierta que numero pense </ion-title>
4   </ion-navbar>
5 </ion-header>
6 <ion-content padding>
7   <ion-input type="number" min="1" max="10" [(ngModel)]="num"
8     placeholder="Introduce un número del 1 al 10"></ion-input>
9   <p>El número que pense es {{ mayorMenor }}</p>
10  <button ion-button block (click)="compruebaNumero()">Acierta</button>
11 </ion-content>

```



encargado de hacer el Data Binding entre el valor del input y la variable de controladora

un Data Binding es **BIDIRECCIONAL** ya que si se introduce un valor en el input automáticamente este será reflejado en la variable del controlador y viceversa

```

1 <ion-header>
2
3
4
5
6
7
8
9
10
11

```



```

1  <ion-header>
2    <ion-navbar>
3      <ion-title> Acierta que
4    </ion-navbar>
5  </ion-header>
6  <ion-content padding>
7    <ion-input type="number" min= 1 max= 10 [(ngModel)]= num"
8    placeholder="Introduce un número del 1 al 10"></ion-input>
9    <p>El número que pense es {{ mayorMenor }}</p>
10   <button ion-button block (click)= compruebaNumero()">Acierta</button>
11 </ion-content>

```

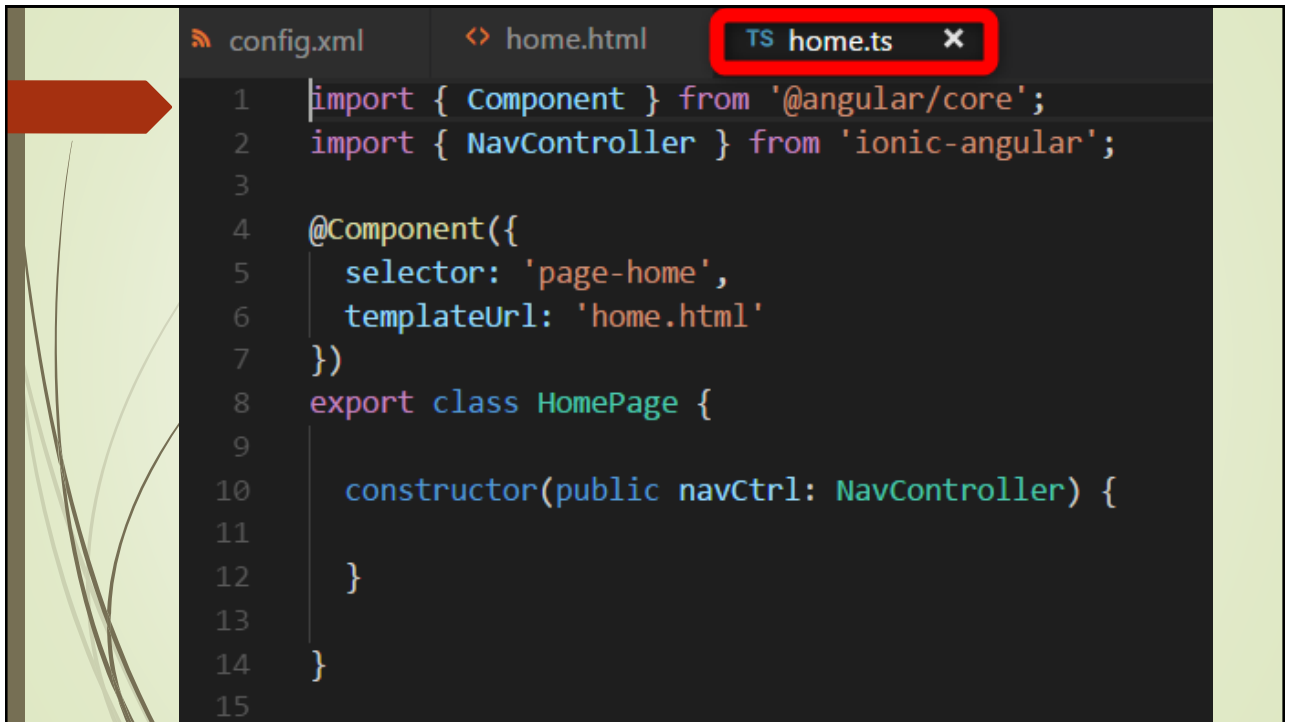
Cuando se utiliza {{ }}
Se crea un databinding
pero este unidireccional

```

1  <ion-header>
2    <ion-navbar>
3      <ion-title> Acierta que numero pense </ion-title>
4    </ion-navbar>
5  </ion-header>
6  <ion-content padding>
7    <ion-input type="number" min="1" max="10" [(ngModel)]="num"
8    placeholder="Introduce un número del 1 al 10"></ion-input>
9    <p>El número que pense es {{ mavorMenor }}</p>
10   <button ion-button block (click)="compruebaNumero()">Acierta</button>
11 </ion-content>

```

<https://ionicframework.com/docs/v2/components/#buttons>



```

1  import { Component } from '@angular/core';
2  import { NavController } from 'ionic-angular';
3
4  @Component({
5    selector: 'page-home',
6    templateUrl: 'home.html'
7  })
8  export class HomePage {
9
10     constructor(public navCtrl: NavController) {
11
12     }
13
14 }
15

```



importar módulos que contienen librerías y clases

NavController. permite la navegación entre distintas páginas.(*)
Component. Las páginas son componentes de Angular, por se necesita importar component.



```
1 import { Component } from '@angular/core';
2 import { NavController } from 'ionic-angular';
3
4 @Component({
5   selector: 'page-home',
6   templateUrl: 'home.html'
7 })
8 export class HomePage {
9
10   constructor(public navCtrl: NavController) {
11
12   }
13
14 }
15
```

@Component
decorador de Angular.

registrar un componente.

Selector: es el selector css
templateUrl: es la plantilla html que va a renderizar la página.

Tipos de decoradores

- ➡ @Component
- ➡ @Injectable
- ➡ @Pipe
- ➡ @Directive

se exporta una clase llamada HomePage

La creación de clases es muy similar a como sería en otros lenguajes de programación orientado a objetos como Java.

Definición de variables

```

1  import { Component } from '@angular/core';
2  import { NavController } from 'ionic-angular';
3
4  @Component({
5    selector: 'page-home',
6    templateUrl: 'home.html'
7  })
8  export class HomePage {
9    1 num: number;
10   2 mayorMenor: string = '...';
11    constructor(public navCtrl: NavController) {
12
13    }
14
15  }

```

Tipos primitivos de variable que podemos definir son:

- number (Numérico).
- string (cadenas de texto).
- boolean (Booleano: true o false).
- any (cualquier tipo).
- Array.

```
export class HomePage {  
  num: number;  
  mayorMenor: string = '...';  
  
  1 numEncontrado: number = this.numAleatorio(0,10);  
  constructor(public navCtrl: NavController) {  
  
  }  
}
```

Declaramos la función

```
//  
export class HomePage {  
  num: number;  
  mayorMenor: string = '...';  
  
  numEncontrado: number = this  
  constructor(public navCtrl: NavController) {  
  
    (alias) class NavController  
    import NavController  
  
    numAleatorio(a,b) {  
      return Math.round(Math.random()*(b-a)+parseInt(a));  
    }  
  }  
}
```

Guarda y compila.

