



UNIVERSIDAD DE COLIMA

FACULTAD DE TELEMATICA

INGENIERIA DE SOFTWARE

IONIC - Proyecto completo para agregar productos a una
base de datos MYSQL utilizando un API REST

Alumno(s):

Noel Gabriel Camacho Cárdenas

19/11/2018

1. Crear un proyecto IONIC en blanco para nuestra app.
2. Crear las paginas para nuestra aplicación:
 1. Utilizaremos la pagina principal para mostrar en el caso de este proyecto el listado de producto por lo que en el archivo home.html contendrá lo siguiente:
 2. Crearemos una pagina para agregar productos llamada agregarproducto en la terminal de visual studio on el comando:

i. Ionic g page agregarproducto

1. Crearemos una página para ver el detalle de los productos llamada detalleproducto en la terminal de visual studio con el comando:

i. Ionic g page detalleproducto

1. Crearemos un provider para manejar el acceso a los datos llamado negocio en la terminal de visual studio con el comando:
 - i. Ionic g provider negocio**

Agregarproducto.html

```

<ion-header>
  <ion-navbar color="primary">
    <ion-title>Nuevo producto</ion-title>
    <ion-buttons end navPop>
      <button ion-button icon-only (click)="AgregarProducto()">
        <ion-icon name="md-checkmark"></ion-icon>
      </button>
    </ion-buttons>
  </ion-navbar>
</ion-header>
<ion-content padding>
  <form>
    <ion-list>
      <ion-item>
        <ion-label floating>Nombre</ion-label>
        <ion-input name="nombre" type="text" [(ngModel)]="producto.nombre" clearInput></ion-input>
      </ion-item>
      <ion-item>
        <ion-label floating>Precio de venta</ion-label>
        <ion-input name="preciodeventa" type="number" [(ngModel)]="producto.preciodeventa" clearInput></ion-input>
      </ion-item>
      <ion-item>
        <ion-label floating>Cantidad</ion-label>
        <ion-input name="cantidad" type="number" [(ngModel)]="producto.cantidad" clearInput></ion-input>
      </ion-item>
      <ion-item>
        <ion-label floating>Precio de costo</ion-label>
        <ion-input name="preciodecosto" type="number" [(ngModel)]="producto.preciodecosto" clearInput></ion-input>
      </ion-item>
      <ion-item>
        <ion-label floating>Descripción</ion-label>
        <ion-input name="descripcion" type="text" [(ngModel)]="producto.descripcion" clearInput></ion-input>
      </ion-item>
      <ion-item>
        <ion-label floating>URL de fotografía</ion-label>
        <ion-input name="fotografia" type="text" [(ngModel)]="producto.fotografia" clearInput></ion-input>
      </ion-item>
    </ion-list>
  </form>
</ion-content>

```

```

TS agregarproducto.ts *
1  import { Component } from '@angular/core';
2  import { IonicPage, NavController, NavParams } from 'ionic-angular';
3  import { NegocioProvider } from '../../providers/negocio/negocio';
4  //import { HomePage } from '../home/home';
5
6  /**
7
8   * Generated class for the AgregarproductoPage page.
9
10  *
11
12  * See https://ionicframework.com/docs/components/#navigation for more info on
13
14  * Ionic pages and navigation.
15
16  */
17
18
19  @IonicPage()
20  @Component({
21    selector: 'page-agregarproducto',
22    templateUrl: 'agregarproducto.html',
23  })
24  export class AgregarproductoPage {
25    mensaje:any;
26    public producto=[];
27    constructor(public navCtrl: NavController, public navParams: NavParams,public negocio:NegocioProvider) {
28    }
29
30    ionViewDidLoad() {
31      console.log('ionViewDidLoad AgregarproductoPage');
32    }
33
34    AgregarProducto()
35    {
36      this.negocio.putProducto(this.producto)
37      .subscribe(
38        (data) => { // Success
39          this.mensaje = data;
40          if(this.mensaje.status=="ok")
41          {

```

```

AgregarProducto()
{
  this.negocio.putProducto(this.producto)
  .subscribe(
    (data) => { // Success
      this.mensaje = data;
      if(this.mensaje.status=="ok")
      {
        alert("Producto agregado!!");
      }
      else
        alert("No se pudo agregar el producto");
    },

    (error) =>{
      alert("Error al agregar el producto!!");
      console.error(error);
    }
  )
}
}

```



```

14 <ion-navbar color="secondary">
15   <ion-title>Edición del Producto</ion-title>
16   <ion-buttons end navPop>
17     <button ion-button icon-only (click)="ActualizarProducto()">
18       <ion-icon name="md-checkmark"></ion-icon>
19     </button>
20   </ion-buttons>
21 </ion-navbar>
22 </ion-header>
23
24 <ion-content padding>
25   <form>
26     <ion-list>
27       <ion-item>
28         <ion-label floating>Nombre</ion-label>
29         <ion-input name="nombre" type="text" [(ngModel)]="producto.nombre" clearInput></ion-input>
30       </ion-item>
31       <ion-item>
32         <ion-label floating>Precio de venta</ion-label>
33         <ion-input name="preciodeventa" type="number" [(ngModel)]="producto.preciodeventa" clearInput></ion-input>
34       </ion-item>
35       <ion-item>
36         <ion-label floating>Cantidad</ion-label>
37         <ion-input name="cantidad" type="number" [(ngModel)]="producto.cantidad" clearInput></ion-input>
38       </ion-item>
39       <ion-item>
40         <ion-label floating>Precio de costo</ion-label>
41         <ion-input name="preciodecosto" type="number" [(ngModel)]="producto.preciodecosto" clearInput></ion-input>
42       </ion-item>
43       <ion-item>
44         <ion-label floating>Descripción</ion-label>
45         <ion-input name="descripcion" type="text" [(ngModel)]="producto.descripcion" clearInput></ion-input>
46       </ion-item>
47       <ion-item>
48         <ion-label floating>URL de fotografía</ion-label>
49         <ion-input name="fotografia" type="text" [(ngModel)]="producto.fotografia" clearInput></ion-input>
50       </ion-item>
51       <ion-item>
52         
53       </ion-item>
54     </ion-list>
55   </form>

```

TS detalleproducto.ts ×

```
1  import { Component } from '@angular/core';
2  import { IonicPage, NavController, NavParams } from 'ionic-angular';
3  import { NegocioProvider } from '../providers/negocio/negocio'
4
5  /**
6
7   * Generated class for the DetalleproductoPage page.
8
9   *
10
11   * See https://ionicframework.com/docs/components/#navigation for more info on
12
13   * Ionic pages and navigation.
14
15   */
16
17
18  @IonicPage()
19  @Component({
20    selector: 'page-detalleproducto',
21    templateUrl: 'detalleproducto.html',
22  })
23
24  export class DetalleproductoPage {
25    productox:any;
26    mensaje:any;
27    public producto=[];
28    constructor(public navCtrl: NavController,
29      public navParams: NavParams,
30      public negocio:NegocioProvider) {
31      let data3 = this.navParams.data;
32      console.log(data3); //this shows as {}
33      this.producto=data3;
34    }
35
36    ionViewDidLoad() {
37      console.log('ionViewDidLoad DetalleproductoPage');
38    }
39
40  }
```

```

41
42 ActualizarProducto()
43 {
44     this.negocio.updateProducto(this.producto)
45     .subscribe(
46         (data) => { // Success
47             this.mensaje = data;
48             if(this.mensaje.status=="ok")
49             {
50                 alert("Producto actualizado!!");
51                 //this.navParams.get("HomePage").someFnToUpdateParent();
52                 //this.navCtrl.pop();
53             }
54             else
55                 alert("No se pudo actualizar el producto");
56         },
57         (error) =>{
58             alert("Error al actualizar el producto!!");
59             console.error(error);
60         }
61     )
62 }
63
64
65 }
66

```

```

TS negocio.ts x
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3
4 /*
5
6     Generated class for the NegocioProvider provider.
7     See https://angular.io/guide/dependency-injection for more info on providers
8     and Angular DI.
9 */
10
11 @Injectable()
12 export class NegocioProvider {
13     constructor(public http: HttpClient) {
14         console.log('Hello NegocioProvider Provider');
15     }
16
17     updateProducto(registro)
18     {
19         var uri="https://noelcamacho.com/proyectoCompleto/actualizarproducto.php?nombre="+registro.nombre
20             +"&cantidad="+registro.cantidad
21             +"&preciodeventa="+registro.preciodeventa
22             +"&preciodecosto="+registro.preciodecosto
23             +"&descripcion="+registro.descripcion
24             +"&fotografia="+registro.fotografia
25             +"&idproducto="+registro.idproducto;
26         var encoded = encodeURI(uri);
27         // alert("encoded="+encoded) ;
28         return this.http.get(encoded);
29     }
30 }
31

```



```

putProducto(registro)
{
    var uri="https://noelcamacho.com/proyectoCompleto/agregarproducto.php?nombre="+registro.nombre
        + "&cantidad="+registro.cantidad
        + "&preciodeventa="+registro.preciodeventa
        + "&preciodecosto="+registro.preciodecosto
        + "&descripcion="+registro.descripcion
        + "&fotografia="+registro.fotografia;
    var encoded = encodeURIComponent(uri);
    // alert("encoded="+encoded) ;
    return this.http.get(encoded);
}

deleteProducto(idproducto)
{
    return this.http.get('https://noelcamacho.com/proyectoCompleto/eliminarproducto.php?idproducto='+idproducto);
}

getProductos()
{
    return this.http.get('https://noelcamacho.com/proyectoCompleto/listarproductos.php');
    //return this.http.get('https://randomuser.me/api/?results=25');
}
}

```

1. Crearemos la base de datos en mysql y el API REST en php para acceder a los datos que tenga las siguientes operaciones:

```

actualizarproducto.php
6  $password = 'colima';
7  $dbname = "id6797666_misdatos";
8  $nombre=$_GET["nombre"];
9  $preciodeventa=$_GET["preciodeventa"];
10 $cantidad=$_GET["cantidad"];
11 $preciodecosto=$_GET["preciodecosto"];
12 $descripcion=$_GET["descripcion"];
13 $fotografia=$_GET["fotografia"];
14 $idproducto=$_GET["idproducto"];
15
16 // Create connection
17 $conn = new mysqli($servername, $username, $password, $dbname);
18 // Check connection
19 if ($conn->connect_error) {
20     die("Connection failed: " . $conn->connect_error);
21 }
22
23 $sql = "UPDATE producto SET nombre='$nombre', preciodecosto=$preciodecosto, cantidad=$cantidad, preciodeventa=$preciodeventa, descripcion='$descripcion'";
24
25 if ($conn->query($sql) === TRUE) {
26     echo '{"status":"ok"}';
27 } else {
28     echo '{"status":"error"}';
29 }
30
31 $conn->close();
32
33 ?>

```

```

agregarproducto.php ×
6  $password = "colima";
7  $dbname = "id6797666_misdatos";
8  $nombre=$_GET["nombre"];
9  $preciodeventa=$_GET["preciodeventa"];
10 $cantidad=$_GET["cantidad"];
11 $preciodecosto=$_GET["preciodecosto"];
12 $descripcion=$_GET["descripcion"];
13 $fotografia=$_GET["fotografia"];
14
15 // Create connection
16 $conn = new mysqli($servername, $username, $password, $dbname);
17 // Check connection
18 if ($conn->connect_error) {
19     die("Connection failed: " . $conn->connect_error);
20 }
21
22 $sql = "INSERT INTO producto (nombre, preciodeventa, cantidad,preciodecosto,descripcion,fotografia)
23     VALUES ('$nombre', $preciodeventa,$cantidad,$preciodecosto,'$descripcion', '$fotografia')";
24
25 if ($conn->query($sql) === TRUE) {
26     echo '{"status":"ok"}';
27 } else {
28     echo '{"status":"error"}';
29 }
30
31 $conn->close();
32
33 ?>

```

```

eliminarproducto.php ×
1  <?php
2
3  Header('Access-Control-Allow-Origin: *');
4
5
6  $servername = "localhost";
7  $username = "id6797666_alumnos";
8  $password = "colima";
9  $dbname = "id6797666_misdatos";
10
11 $idproducto=$_GET["idproducto"];
12
13 // Create connection
14 $conn = new mysqli($servername, $username, $password, $dbname);
15 // Check connection
16 if ($conn->connect_error) {
17     die("Connection failed: " . $conn->connect_error);
18 }
19
20 $sql = "DELETE FROM producto WHERE idproducto=$idproducto";
21
22 if ($conn->query($sql) === TRUE) {
23     echo '{"status":"ok"}';
24 } else {
25     echo '{"status":"error"}';
26 }
27
28 $conn->close();
29
30 ?>

```

```

listarproductos.php x
22 // output data of each row
23 $registros=array();
24 $i=0;
25 while($row = $result->fetch_assoc()) {
26     $registros[$i]=$row;
27     $i++;
28 }
29 echo '{"records":'. json_encode($registros).'}';
30 } else {
31     echo '{"records":[]}';
32 }
33
34 $conn->close();
35
36 ?>

```

Y el resultado es el siguiente, pantallas de la aplicación funcionando.

