

## Ejemplo práctico de Ionic

### Aplicación para visualizar recetas

Repositorio: <https://github.com/ManuCastrillonM/Ionic-example/>

\*Consejo: Cada paso del desarrollo está especificado en una rama del proyecto, puedes navegar al paso que desees utilizando: `git checkout [número-del-paso]`

#### Pre-requisitos:

- NodeJS
- Npm

#### Instalación:

> `npm install -g cordova ionic`

#### Desarrollo:

##### 1. Crear el proyecto

###### a. Creación del proyecto

> `ionic start ionic-example blank`

```
manuela@Manuela-PC:~/Documentos$ ionic start ionic-example blank
✓ Creating directory ./ionic-example - done!
Fetching app base (https://github.com/driftyco/ionic2-app-base/archive/master.tar.gz)
✓ Downloading - done!
Fetching starter template blank (https://github.com/driftyco/ionic2-starter-blank/archive/master.tar.gz)
✓ Downloading - done!
✓ Updating package.json with app details - done!
✓ Creating configuration file ionic.config.json - done!
Installing dependencies may take several minutes!
> npm install
✓ Running command - done!
> npm install --save-dev --save-exact @ionic/cli-plugin-ionic-angular@latest
✓ Running command - done!

🎵 🎵 🎵 Your Ionic app is ready to go! 🎵 🎵 🎵

Run your app in the browser (great for initial development):
  ionic serve

Run on a device or simulator:
  ionic cordova run ios

Test and share your app on a device with the Ionic View app:
  http://view.ionic.io

? Link this app to your Ionic Dashboard to use tools like Ionic View? No

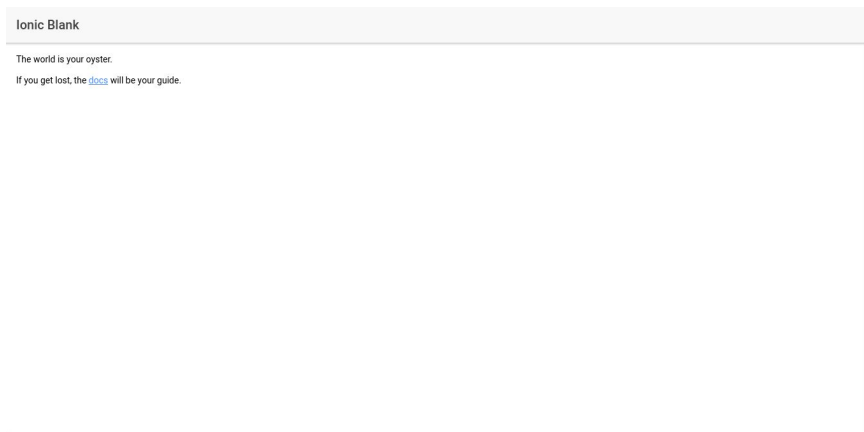
Go to your newly created project: cd ./ionic-example
```

###### b. Ingresar al directorio del proyecto

> `cd ionic-example`

###### c. Ejecutar por primer vez el proyecto

> ionic serve



## 2. Crear vista principal

### a. Crear tabs

> ionic generate tabs

```
manuela@Manuela-PC:~/Documentos/ionic-example$ ionic generate tabs tabs
? How many tabs? 3
? Name of this tab: Profile
? Name of this tab: Recipes
? Name of this tab: Logout
[OK] Generated a tabs page named tabs!
```

### b. Importar la página “TabPage” en el módulo de la aplicación, agregandola a “declarations” y “entryComponents”

```

app.module.ts
3 import { IonicApp, IonicErrorHandler, IonicModule } from '@ionic/angular';
4 import { SplashScreen } from '@ionic-native/splash-screen';
5 import { StatusBar } from '@ionic-native/status-bar';
6
7 import { MyApp } from './app.component';
8 import { HomePage } from '../pages/home/home';
9 import { TabsPage } from '../pages/tabs/tabs';
10
11 @NgModule({
12   declarations: [
13     MyApp,
14     HomePage,
15     TabsPage
16   ],
17   imports: [
18     BrowserModule,
19     IonicModule.forRoot(MyApp)
20   ],
21   bootstrap: [IonicApp],
22   entryComponents: [
23     MyApp,
24     HomePage,
25     TabsPage
26   ],
27   providers: [

```

- c. Cambiar el rootPage por “TabsPage” en el archivo “src/app/app.component.ts”

```

app.component.ts
1 import { Component } from '@angular/core';
2 import { Platform } from '@ionic/angular';
3 import { StatusBar } from '@ionic-native/status-bar';
4 import { SplashScreen } from '@ionic-native/splash-screen';
5 import { TabsPage } from '../pages/tabs/tabs';
6
7 @Component({
8   templateUrl: 'app.html'
9 })
10 export class MyApp {
11   rootPage:any = TabsPage;
12
13   constructor(platform: Platform, statusBar: StatusBar, splashScreen: SplashScreen) {
14     platform.ready().then(() => {
15       // Okay, so the platform is ready and our plugins are available.
16       // Here you can do any higher level native things you might need.
17       statusBar.styleDefault();
18       splashScreen.hide();
19     });
20   }
21 }

```

- d. Personalizar los iconos de las tabs [[Documentación sobre ionicons](#)]

```

1 <ion-tabs>
2   <ion-tab [root]="profileRoot" tabTitle="Profile" tabIcon="person"></ion-tab>
3   <ion-tab [root]="recipesRoot" tabTitle="Recipes" tabIcon="pizza"></ion-tab>
4   <ion-tab [root]="logoutRoot" tabTitle="Logout" tabIcon="log-out"></ion-tab>
5 </ion-tabs>

```

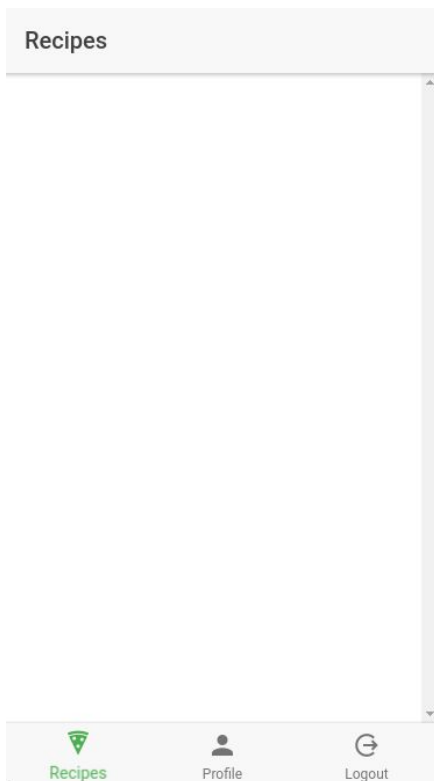
- e. Personalizar colores de la aplicación en el archivo "src/theme/variables.scss"

```

25
26 $colors: (
27   primary: #4caf50,
28   secondary: #ff9800,
29   danger: #ff5722,
30   light: #f4f4f4,
31   dark: #222
32 );

```

Aplicación actual:



3. Creación de provider para la página "Recipes"
  - a. Generar Provider para obtener las recetas
    - > ionic generate provider recipes

```
manuela@Manuela-PC:~/Documentos/ionic-example$ ionic generate provider recipes
[OK] Generated a provider named recipes!
```

b. Crear la clase “recipe” en el provider y un arreglo de las mismas

```
ts recipes.ts
1  import { Injectable } from '@angular/core';
2  import { Http } from '@angular/http';
3  import 'rxjs/add/operator/map';
4
5  export class Recipe {
6    constructor(
7      public id?: number,
8      public name?: string,
9      public ingredients?: string,
10     public instructions?: string
11   ) { }
12 }
13
14 const RECIPES: Recipe[] = [];
15
16 @Injectable()
17 export class RecipesProvider {
18
19   constructor(public http: Http) {
20     console.log('Hello RecipesProvider Provider');
21   }
22
23 }
```

c. Métodos para agregar, listar y eliminar recetas.

```
ts recipes.ts
39
40 addRecipe(recipe): void {
41   let recipeId = RECIPES.reduce((max, recipe) => Math.max(recipe.id, max), -1) + 1;
42   recipe.id = recipeId;
43   RECIPES.push(recipe);
44 }
45
46 getRecipes(): Promise<Recipe[]> {
47   return new Promise<Recipe[]>(resolve => {
48     setTimeout(() => {
49       resolve(RECIPES);
50     }, FETCH_LATENCY);
51   })
52 }
53
54 getRecipe(id: number | string): Promise<Recipe> {
55   return new Promise<Recipe>(resolve => {
56     setTimeout(() => {
57       resolve(RECIPES.filter(recipe => recipe.id === +id)[0]);
58     }, FETCH_LATENCY);
59   })
60 }
61
62 removeRecipe(id: number | string): void {
63   let recipeIdx = +id - 1;
64   RECIPES.splice(recipeIdx, 1);
65 }
66 }
```

Documentación:

- [Método Reduce\(\)](#)
- [Promesas en Javascript](#)
- [Método Splice\(\)](#)

d. Importar servicio de recetas en el controlador de la página “recipes”

```
1 import { Component } from '@angular/core';
2 import { IonicPage, NavController, NavParams } from 'ionic-angular';
3 import { Recipe, RecipesProvider } from '../../providers/recipes/recipes'
4
5 @IonicPage()
6 @Component({
7   selector: 'page-recipes',
8   templateUrl: 'recipes.html',
9 })
10 export class RecipesPage {
11
12   constructor(public navCtrl: NavController,
13     public recipeService: RecipesProvider) {
14   }
15
16   ionViewDidLoad() {
17     console.log('ionViewDidLoad RecipesPage');
18   }
19
20 }
```

e. Cargar la lista de recetas desde el provider en el método “ngOnInit()” y guardarlas en un arreglo.

```
7   selector: 'page-recipes',
8   templateUrl: 'recipes.html',
9 })
10 export class RecipesPage {
11
12   recipes: Recipe[];
13
14   constructor(public navCtrl: NavController,
15     public recipeService: RecipesProvider) {
16   }
17
18   ionViewDidLoad() {
19     console.log('ionViewDidLoad RecipesPage');
20   }
21
22   ngOnInit() {
23     this.recipeService.getRecipes().then(
24       (recipes) => {
25         console.log(recipes);
26         this.recipes = recipes;
27       },
28       (err) => {
29         console.log("Oops something is wrong!!");
30       }
31     );
32   }
```

Documentación:

- [Ciclo de vida de una vista](#)



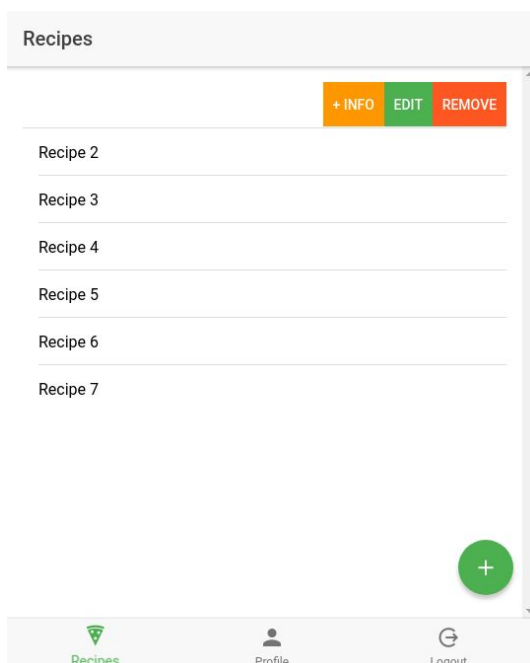
## 4. Creación de vista “Recipes”

- a. Se crea la lista de elementos que va a mostrar las recetas.

```

1  </ion-header>
2
3  <ion-content padding>
4    <ion-list *ngIf="recipes && recipes.length > 0">
5      <ion-item-sliding *ngFor="let recipe of recipes">
6        <ion-item>
7          {{recipe.name}}
8        </ion-item>
9
10       <ion-item-options side="right">
11         <button ion-button color="secondary" (click)="presentRecipeModal(recipe)">+ Info</button>
12         <button ion-button (click)="edit(recipe)">Edit</button>
13         <button ion-button color="danger" (click)="remove(recipe)">Remove</button>
14       </ion-item-options>
15     </ion-item-sliding>
16   </ion-list>
17
18   <ion-fab right bottom>
19     <button ion-fab (click)="add()">
20       <ion-icon name="add"></ion-icon>
21     </button>
22   </ion-fab>
23
24   <p *ngIf="recipes && recipes.length == 0">
25     No recipes available
26   </p>
27 </ion-content>

```



## 5. Crear formulario para agregar y editar recetas.

Ya que el formulario para agregar y editar recetas es similar, este se puede crear como componente para ser reutilizado posteriormente.

### a. Crear componente “recipe-form”

```
manuela@Manuela-PC:~/Documentos/ionic-example$ ionic generate component recipe-form
[OK] Generated a component named recipe-form!
```

### b. Editar vista del formulario

```
recipe-form.html
1 <form #recipeForm="ngForm" name="recipeForm" novalidate>
2   <ion-item>
3     <ion-label floating color="primary">Name</ion-label>
4     <ion-input [(ngModel)]="recipe.name" name="name" type="text" #name="ngModel" required></ion-input>
5   </ion-item>
6
7   <p [hidden]="name.valid" color="danger" padding-left>
8     The field is required
9   </p>
10
11  <ion-item>
12    <ion-label floating color="primary">Ingredients</ion-label>
13    <ion-textarea [(ngModel)]="recipe.ingredients" name="ingredients" type="text" #ingredients="ngModel" required></ion-textarea>
14  </ion-item>
15
16  <p [hidden]="ingredients.valid" color="danger" padding-left>
17    The field is required
18  </p>
19
20  <ion-item>
21    <ion-label floating color="primary">Instructions</ion-label>
22    <ion-textarea [(ngModel)]="recipe.instructions" name="instructions" type="text" #instructions="ngModel" required></ion-textarea>
23  </ion-item>
24
25  <p [hidden]="instructions.valid" color="danger" padding-left>
26    The field is required
27  </p>
28
29  <ion-row responsive-sm>
30    <ion-col>
31      <button ion-button [disabled]="!recipeForm.valid" (click)="processRecipe()" type="submit" color="primary">
32        {{submitBtn}}
33      </button>
34    </ion-col>
35  </ion-row>
36 </form>
```

Documentación:

- [NgModel](#)

### c. Crear instrucciones para el texto del botón “submit”



```
recipe-form.html  recipe-form.ts
1 import { Component, Input, Output, EventEmitter } from '@angular/core';
2 import { Recipe, RecipesProvider } from '../providers/recipes/recipes'
3
4 @Component({
5   selector: 'recipe-form',
6   templateUrl: 'recipe-form.html'
7 })
8 export class RecipeFormComponent {
9
10   @Input() recipe: Recipe;
11   @Input() isUpdate: boolean;
12   @Output() fireAction: EventEmitter<Recipe> = new EventEmitter<Recipe>();
13   submitBtn: string;
14
15   constructor(private recipeService: RecipesProvider) {
16     console.log('Hello RecipeFormComponent Component');
17   }
18
19   ngOnInit() {
20     this.submitBtn = this.isUpdate ? "Update" : "Save";
21   }
22
23   processRecipe(): void {
24     this.fireAction.emit(this.recipe);
25   }
26
27 }
```

Documentación:

- [@Input y @Output](#)
- [EventEmitter](#)

## 6. Agregar recetas!

- Generar página para agregar receta  
> ionic generate page recipe-add

```
manuela@Manuela-PC:~/Documentos/ionic-example$ ionic generate page recipe-add
[OK] Generated a page named recipe-add!
```

- Importar página en el módulo de la aplicación

```
recipe-add.html recipe-add.ts recipes.ts app.module.ts
import { NgModule } from '@angular/core';
import { MyApp } from '../app.component';
import { HomePage } from '../pages/home/home';
import { TabsPage } from '../pages/tabs/tabs';
import { RecipeAddPage } from '../pages/recipe-add/recipe-add';
import { RecipesProvider } from '../providers/recipes/recipes';
import { RecipeFormComponent } from '../components/recipe-form/recipe-form';

@NgModule({
  declarations: [
    MyApp,
    HomePage,
    TabsPage,
    RecipeFormComponent,
    RecipeAddPage
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp)
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp,
    HomePage,
    TabsPage,
    RecipeAddPage
  ]
})
```

c. Agregar componente “recipe-form” a la vista “recipe-add”

```
recipe-add.html
<ion-header>
  <ion-navbar>
    <ion-title>Add recipe</ion-title>
  </ion-navbar>
</ion-header>

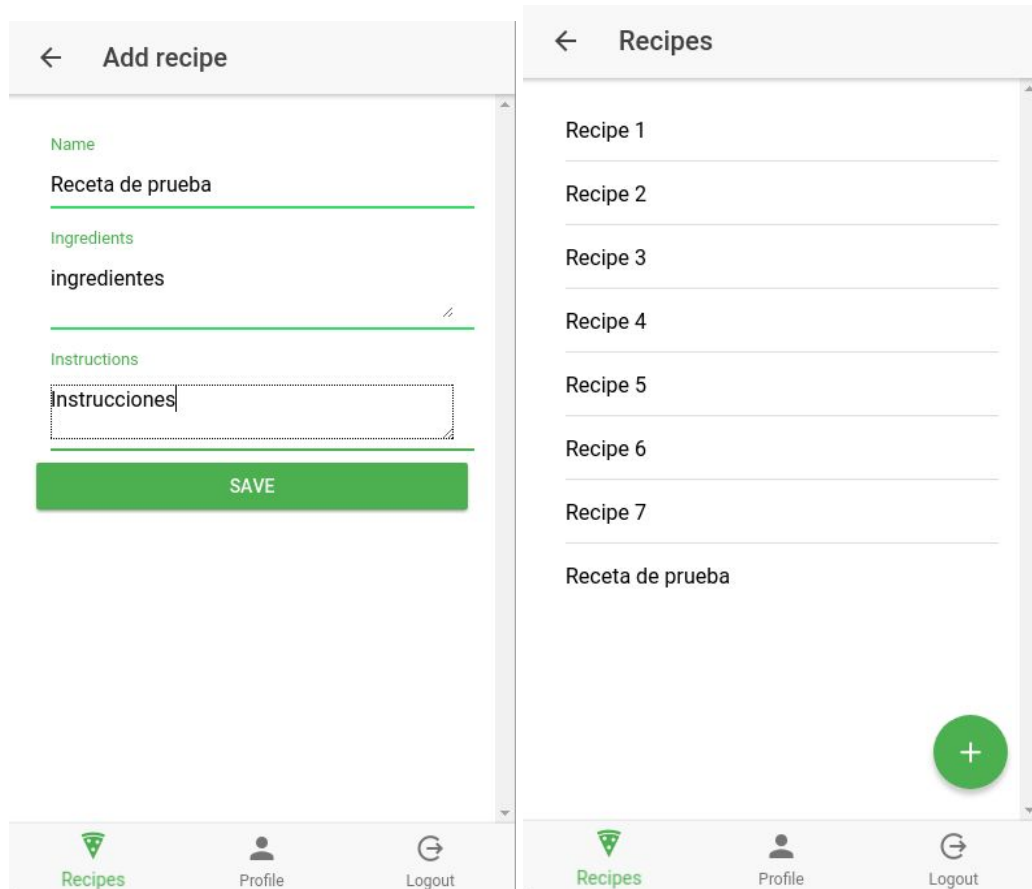
<ion-content padding>
  <recipe-form (fireAction)="save($event)" [isUpdate]="false" [recipe]="recipe"></recipe-form>
</ion-content>
```

d. Crear método para agregar receta a través del “RecipesProvider”

```
recipe-add.html recipe-add.ts
2 import { IonicPage, NavController } from 'ionic-angular';
3 import { Recipe, RecipesProvider } from '.../providers/recipes/recipes';
4 import { RecipesPage } from '.../recipes/recipes';
5
6 @IonicPage()
7 @Component({
8   selector: 'page-recipe-add',
9   templateUrl: 'recipe-add.html',
10 })
11 export class RecipeAddPage {
12
13   recipe: Recipe = new Recipe();
14   constructor(
15     public navCtrl: NavController,
16     private recipeService: RecipesProvider
17   ) { }
18
19   save(recipe) {
20     this.recipeService.addRecipe(recipe);
21     this.navCtrl.push(RecipesPage);
22   }
23
24   ionViewDidLoad() {
25     console.log('ionViewDidLoad RecipeAddPage');
26   }
27
28 }
```

- e. Implementar método “add()” en el controlador de la página “recipes”, que se ejecutará al pulsar el botón con el símbolo “+”

```
recipe-add.html recipe-add.ts recipes.ts
1 import { Component } from '@angular/core';
2 import { IonicPage, NavController } from 'ionic-angular';
3 import { Recipe, RecipesProvider } from '.../providers/recipes/recipes';
4 import { RecipeAddPage } from '.../recipe-add/recipe-add';
5
35 add() {
36   this.navCtrl.push(RecipeAddPage);
37 }
```



## 7. Mostrar detalles de una receta

### a. Crear página “recipe-info”

> ionic generate page recipe-info

```
manuela@Manuela-PC:~/Documentos/ionic-example$ ionic generate page recipe-info
[OK] Generated a page named recipe-info!
```

### b. Importar página al módulo de la aplicación

```

app.module.ts
12 import { RecipeFormComponent } from '../components/recipe-form/recipe-form';
13 import { RecipeInfoPage } from '../pages/recipe-info/recipe-info';
14
15 @NgModule({
16   declarations: [
17     MyApp,
18     HomePage,
19     TabsPage,
20     RecipeFormComponent,
21     RecipeAddPage,
22     RecipeInfoPage
23   ],
24   imports: [
25     BrowserModule,
26     IonicModule.forRoot(MyApp)
27   ],
28   bootstrap: [IonicApp],
29   entryComponents: [
30     MyApp,
31     HomePage,
32     TabsPage,
33     RecipeAddPage,
34     RecipeInfoPage
35   ],

```

c. Importar “ModalController” en el controlador de la página “recipes”

```

app.module.ts  recipe-info.html  recipe-info.ts  recipes.ts  recipes.html
1 import { Component } from '@angular/core';
2 import { ModalController, IonicPage, NavController } from 'ionic-angular';
3 import { Recipe, RecipesProvider } from '../providers/recipes/recipes';
4 import { RecipeAddPage } from '../recipe-add/recipe-add';
5 import { RecipeInfoPage } from '../recipe-info/recipe-info';
6
7 @IonicPage()
8 @Component({
9   selector: 'page-recipes',
10  templateUrl: 'recipes.html',
11 })
12 export class RecipesPage {
13
14   recipes: Recipe[];
15
16   constructor(public navCtrl: NavController,
17     public modalCtrl: ModalController,
18     public recipeService: RecipesProvider) {
19

```

d. Implementar método “PresentRecipeModal()” en el controlador de la página “recipes” el cual se ejecuta al seleccionar la opción “+ info” en cada una de las recetas.

```

41 presentRecipeModal(recipe) {
42   let recipeModal = this.modalCtrl.create(RecipeInfoPage, {
43     recipe: recipe
44   });
45   recipeModal.present();
46 }

```

e. Importar “ViewController” en el constructor del controlador de la página “recipe-info”

```

1 import { Component } from '@angular/core';
2 import { IonicPage, NavController, ViewController, NavParams } from 'ionic-angular';
3
4 @IonicPage()
5 @Component({
6   selector: 'page-recipe-info',
7   templateUrl: 'recipe-info.html',
8 })
9 export class RecipeInfoPage {
10
11   constructor(public navCtrl: NavController,
12     public navParams: NavParams,
13     public viewCtrl: ViewController) { }

```

- f. En el método “ngOnInit” recuperar la receta enviada como parámetro y asignarla a una variable tipo “Recipe” en el controlador asegurándonos de importar la clase “Recipe”.

```

22 ngOnInit() {
23   this.recipe = this.navParams.get("recipe");
24 }

```

- g. Crear vista HTML con una card en la cual se va a mostrar la información detallada de la receta.

```

1 <ion-header>
2
3   <ion-toolbar>
4     <ion-title>Recipe detail</ion-title>
5     <ion-buttons start>
6       <button ion-button (click)="closeModal()">
7         <ion-icon name="close"></ion-icon>
8       </button>
9     </ion-buttons>
10  </ion-toolbar>
11
12 </ion-header>
13
14 <ion-content padding>
15   <ion-card>
16     <img [src]="assets/img/recipe.jpg" />
17     <ion-card-content>
18       <ion-card-title color="secondary">{{recipe.name}}</ion-card-title>
19       <h2>Ingredients:</h2>
20       <p> {{recipe.ingredients}} </p>
21       <h2>Instructions:</h2>
22       <p> {{recipe.instructions}} </p>
23     </ion-card-content>
24   </ion-card>
25 </ion-content>

```

- h. Implementar método “closeModal()” para cerrar el modal

```

26 closeModal(){
27   this.viewCtrl.dismiss();
28 }

```





## Generar APK:

- > cordova platform add android
- > cordova build --release android

## Ejercicio propuesto:

- Implementar la funcionalidad para editar y eliminar recetas.