

Comandos

ionic start myApp tabs --no-git (blank, tabs, sidemenu)

ionic serve

ionic generate component

ionic generate page

ionic generate pipe

ionic cordova build android

ionic cordova run android --livereload

json-server db.json --delay 2000

* Añadir **HttpClientModule** en imports.

app.module.ts

- Las páginas se añaden en “declarations” y “entryComponents”.
- Los componentes se añaden en “declarations”.
- Los pipes se añaden en “declarations”.
- Los servicios se añaden en “providers”.

Pipe

```
@Pipe({
  name: 'lengthFormat'
})

transform(text: string, length: number, money: string): string {
  if (text.length > length){
    return text.substring(0, length-3) + '...' + money;
  }
}

-----

<p>{{fruit.description | lengthFormat: 40 : '€' | uppercase }}</p>

-----
```

En este ejemplo se crea un pipe que recibe dos parametros: un número y un string. Para utilizarlo se utiliza el nombre que se defina en “name” que en este caso es ‘ lengthFormat’ y se pasan los parámetros separados de dos puntos.

Además se pueden utilizar varios Pipes a la vez, separandolos por ‘|’.

Spinner

```
<div text-center *ngIf="loading">
  <ion-spinner></ion-spinner>
</div>
```

Navegación

	App.component.ts	Page
Ir directamente	This.nav.setRoot();	This.navCtrl.setRoot();
Ir hacía delante	This.nav.push();	This.navCtrl.push();
Ir hacía atrás	This.nav.pop();	This.nav.push();

```
import {NavController, NavParams} from 'ionic-angular';
```

```
constructor(public navCtrl: NavController,  
             public NavParams: NavParams)  
{}
```

```
this.navCtrl.push(FruitPage, {  
    fruit: fruit,  
    name: name  
});
```

En primer lugar se pasa la página a la que se va a ir, y seguidamente se pasan los parámetros que se envían entre llaves. Estos parámetros se reciben de la siguiente manera:

```
this.navParams.get('name');
```

Event Emit

```
@Output() hasVoted = new EventEmitter();
```

```
like(){  
    this.likeVote += 1;  
    this.hasVoted.emit('me gusta');  
}
```

```
<vote (hasVoted)="showAlert($event)"></vote>
```

```
showAlert(text: Event){  
    let alert = this.alertCtrl.create({  
        title: this.selectedFruit.name,  
        subTitle: 'Has votado ' + text + '.',  
        buttons: ['Dismiss']  
    });  
    alert.present();  
}
```

Pasar variable a componente

```
<common-navbar [name]="name"></common-navbar>
```

```
@Input() name: string;
```

ng-content

```
<expandable-content>
  <p>Contenido</p>
</expandable-content>

-----

<ng-content></ng-content>
```

Observable

```
constructor(private http: HttpClient){
}

getFruits(): Observable<FruitItems[]> {
  return <Observable<FruitItems[]>>
  this.http.get('http://localhost:3000/fruits');
}

saveFruit(fruit: FruitItems): Observable<FruitItems> {
  return <Observable<FruitItems>>
    this.http.put('http://localhost:3000/fruits/' + fruit.id, fruit);
}
```

Llamadas al servicio

```
this.fruitService.getFruits()
  .subscribe((items: FruitItems[])=>{
    this.fruits = items;
    this.loading = false;
  }, (error: HttpErrorResponse) => {
    console.log('Oh! Algo fue mal. ');
    this.errorLoading = true;
    this.loading = false;
  });

this.fruitService.saveFruit(this.fruit)
  .subscribe(() => {
    this.updating = false;
    this.toastService.showToast('Fruit update. ');
  }, () => {
    this.updating = false;
    this.toastService.showToast('Error updating fruit. ');
  })
```

Toast

```
showToast(message: string) {
  this.toastCtrl.create({
    message,
    duration: 3000
  }).present();
}
```

Reactive Forms

```
private form: FormGroup;
```

```

this.form = new FormGroup ({
  name: new FormControl(null, Validators.required),
  birthday: new FormControl(null, Validators.required),
  ...
});

this.form.patchValue(user);           Introduce valores en el formulario
this.form.patchValue({
  name: 'jose',
  birthday: new Date(1993-30-09),
  ...
});

```

Inputs

```

<form [formGroup]="form" (submit)="submitForm()">
  <ion-item>
    <ion-label>Name</ion-label>
    <ion-input type="text" formControlName="name"></ion-input>
  </ion-item>

  <ion-item>
    <ion-label>Birthdate</ion-label>
    <ion-datetime formControlName="birthDate"></ion-datetime>
  </ion-item>

  <ion-item>
    <ion-label>Sex</ion-label>
    <ion-select formControlName="sex">
      <ion-option value="male">Male</ion-option>
      <ion-option value="female">Female</ion-option>
    </ion-select>
  </ion-item>

  <ion-item>
    <ion-label>Phone</ion-label>
    <ion-input type="number" formControlName="phone"></ion-input>
  </ion-item>

  <ion-item>
    <ion-label>Email</ion-label>
    <ion-input type="email" formControlName="email"></ion-input>
  </ion-item>

  <div margin-top text-end>
    <button ion-button disabled="!form.valid">Save</button>
  </div>
</form>

```

Menú a la derecha

```

<ion-navbar>
  <button ion-button menuToggle end>
  ...

```

```

<ion-menu [content]="content" side="right" persistent >   [Esto se encuentra en app.html]

```

Card

```
<ion-card>
  <ion-card-header>
  </ion-card-header>
  <ion-card-content>
  </ion-card-content>
</ion-card>
```