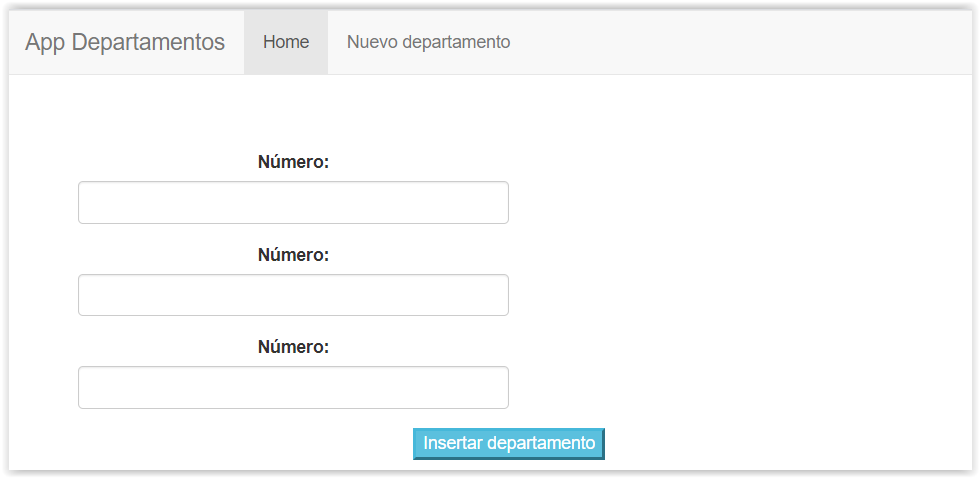
**APLICACION CRUD DEPARTAMENTOS COMPLETA**

* Vamos a realizar una aplicación CRUD accediendo a un servicio Api con todas las posibles acciones.
* Utilizaremos, formularios, Routing y parámetros.







Incluimos los módulos para trabajar con formularios y Routing dentro de Angular

**APP.MODULE.TS**

import { BrowserModule } from "@angular/platform-browser";

import { NgModule } from "@angular/core";

import { FormsModule } from "@angular/forms";

import { AppComponent } from "./app.component";

import { DepartamentosComponent } from "./components/departamentos/departamentos.component";

import { HttpClientModule } from "@angular/common/http";

import { routing, appRoutingProviders } from "./app.routing";

import { MenuPrincipalComponent } from './components/menu-principal/menu-principal.component';

import { InsertarDepartamentoComponent } from './components/insertar-departamento/insertar-departamento.component';

import { DetallesDepartamentoComponent } from './components/detalles-departamento/detalles-departamento.component';

@NgModule({

  declarations: [AppComponent, DepartamentosComponent, MenuPrincipalComponent, InsertarDepartamentoComponent, DetallesDepartamentoComponent],

  imports: [BrowserModule, FormsModule, HttpClientModule, routing],

  providers: [appRoutingProviders],

  bootstrap: [AppComponent]

})

export class AppModule {}

Comenzamos creando un nuevo componente llamado **MenuPrincipal**

**MENUPRINCIPAL.COMPONENT.HTML**

<nav class="navbar navbar-default">

  <div class="container-fluid">

    <!-- Brand and toggle get grouped for better mobile display -->

    <div class="navbar-header">

      <button

        type="button"

        class="navbar-toggle collapsed"

        data-toggle="collapse"

        data-target="#bs-example-navbar-collapse-1"

        aria-expanded="false"

      >

        <span class="sr-only">Toggle navigation</span>

        <span class="icon-bar"></span>

        <span class="icon-bar"></span>

        <span class="icon-bar"></span>

      </button>

      <a class="navbar-brand" href="/">App Departamentos</a>

    </div>

    <!-- Collect the nav links, forms, and other content for toggling -->

    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

      <ul class="nav navbar-nav">

        <li class="active">

          <a href="/">Home <span class="sr-only">(current)</span></a>

        </li>

        <li><a href="/insertar">Nuevo departamento</a></li>

      </ul>

    </div>

    <!-- /.navbar-collapse -->

  </div>

  <!-- /.container-fluid -->

</nav>

Creamos una nueva carpeta llamada **services** y dentro un nuevo fichero llamado **global.ts**

**GLOBAL.TS**

export var Global = {

  urlpersonas: "https://servicioapipersonasmvcpgs.azurewebsites.net/api",

  urldepartamentos:

    "https://apicruddepartamentoscore.azurewebsites.net/"

};

Creamos una carpeta llamada **models** y un nuevo modelo llamado **departamento.ts**

**DEPARTAMENTO.TS**

export class Departamento {

  constructor(

    public numero: number,

    public nombre: string,

    public localidad: string

  ) {}

}

Creamos una nueva carpeta llamada **services** y un nuevo fichero llamado **departamento.service.ts**

**DEPARTAMENTO.SERVICE.TS**

import { Injectable } from "@angular/core";

import { HttpClient, HttpHeaders } from "@angular/common/http";

import { Departamento } from "./../models/departamento";

import { Observable } from "rxjs";

import { Global } from "./global";

@Injectable()

export class DepartamentoService {

  public url: string;

  constructor(private \_http: HttpClient) {

    this.url = Global.urldepartamentos;

  }

  getDepartamentos(): Observable<any> {

    var peticion = "/departamentos";

    return this.\_http.get(this.url + peticion);

  }

  buscarDepartamento(iddepartamento): Observable<any> {

    var peticion = "/departamentos/" + iddepartamento;

    return this.\_http.get(this.url + peticion);

  }

  insertarDepartamento(departamento): Observable<any> {

    //DEBEMOS CONVERTIR EL OBJETO DEPARTAMENTO

    //A JSON STRING

    let deptjson = JSON.stringify(departamento);

    //INDICAMOS EL TIPO DE PETICION QUE VAMOS A REALIZAR

    let headers = new HttpHeaders().set("Content-Type", "application/json");

    var peticion = "/departamentos/post";

    return this.\_http.post(this.url + peticion, deptjson, { headers: headers });

  }

  modificarDepartamento(departamento): Observable<any> {

    //DEBEMOS CONVERTIR EL OBJETO DEPARTAMENTO

    //A JSON STRING

    let deptjson = JSON.stringify(departamento);

    //INDICAMOS EL TIPO DE PETICION QUE VAMOS A REALIZAR

    let headers = new HttpHeaders().set("Content-Type", "application/json");

    var peticion = "/departamentos/put";

    return this.\_http.put(this.url + peticion, deptjson, { headers: headers });

  }

  eliminarDepartamento(iddepartamento): Observable<any> {

    var peticion = "/departamentos/delete/" + iddepartamento;

    return this.\_http.delete(this.url + peticion);

  }

}

Creamos un fichero para Routin llamado **app.routing.ts**

**APP.ROUTING.TS**

//IMPORTAMOS LOS MODULOS DEL ROUTER DE ANGULAR

import { ModuleWithProviders } from "@angular/core";

import { Routes, RouterModule } from "@angular/router";

//AHORA DEBEMOS IMPORTAR LOS COMPONENTES QUE VAYAN A FORMAR PARTE

//DE LA NAVEGACION, ES DECIR, LOS QUE VISUALIZAREMOS

//DENTRO DE LAS ZONAS DE NUESTRO SITE

//IMPORTAMOS EL COMPONENTE DE NUMERO DOBLE

import { DepartamentosComponent } from "./components/departamentos/departamentos.component";

import { InsertarDepartamentoComponent } from "./components/insertar-departamento/insertar-departamento.component";

import { DetallesDepartamentoComponent } from "./components/detalles-departamento/detalles-departamento.component";

//DEFINIMOS EL ARRAY DE RUTAS, ES DECIR, LA CONFIGURACION DE TODAS

//LAS RUTAS PARA NUESTRA APP, QUE SERA NUMERO DOBLE SIN PARAMETROS Y CON //PARAMETRO OPCIONAL.

const appRoutes: Routes = [

  { path: "", component: DepartamentosComponent },

  { path: "insertar", component: InsertarDepartamentoComponent },

  { path: "detalles/:iddepartamento", component: DetallesDepartamentoComponent }

];

//AHORA EXPORTAMOS EL MODULO DEL ROUTER PARA PODER

//UTILIZARLO DENTRO DE ANGULAR

export const appRoutingProviders: any[] = [];

export const routing: ModuleWithProviders = RouterModule.forRoot(appRoutes);

Dibujamos el Routing dentro de la página principal **app.component.html**

**APP.COMPONENT.CSS**

body {

  padding-top: 50px;

}

.starter-template {

  padding: 40px 15px;

  text-align: center;

}

**APP.COMPONENT.HTML**

<app-menu-principal></app-menu-principal>

<div class="container">

  <div class="starter-template">

    <router-outlet></router-outlet>

  </div>

</div>

Agregamos las referencias a Bootstrap dentro de **index.html**

**INDEX.HTML**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <title>Ejemplos</title>

    <base href="/" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <link rel="icon" type="image/x-icon" href="favicon.ico" />

    <link

      rel="stylesheet"

      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"

      integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"

      crossorigin="anonymous"

    />

  </head>

  <body>

    <app-root></app-root>

    <script

      src="https://code.jquery.com/jquery-3.1.1.min.js"

      integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8="

      crossorigin="anonymous"

    ></script>

    <script

      src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"

      integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Txa"

      crossorigin="anonymous"

    ></script>

  </body>

</html>

Creamos el componente **Departamentos**

**DEPARTAMENTOS.COMPONENT.HTML**

<table class="table table-bordered">

  <thead>

    <tr>

      <th>NUMERO</th>

      <th>NOMBRE</th>

      <th>LOCALIDAD</th>

      <th></th>

    </tr>

  </thead>

  <tbody>

    <tr \*ngFor="let dept of departamentos">

      <td>{{ dept.numero }}</td>

      <td>{{ dept.nombre }}</td>

      <td>{{ dept.localidad }}</td>

      <td>

        <a href="detalles/{{ dept.numero }}">

          Detalles

        </a>

      </td>

    </tr>

  </tbody>

</table>

**DEPARTAMENTOS.COMPONENT.TS**

import { Component, OnInit, ViewChild, ElementRef } from "@angular/core";

import { DepartamentoService } from "./../../services/departamento.service";

import { Departamento } from "./../../models/departamento";

@Component({

  selector: "app-departamentos",

  templateUrl: "./departamentos.component.html",

  styleUrls: ["./departamentos.component.css"],

  providers: [DepartamentoService]

})

export class DepartamentosComponent implements OnInit {

  public departamentos: Array<Departamento>;

  constructor(private \_departamentoService: DepartamentoService) {}

  cargarDepartamentos() {

    this.\_departamentoService.getDepartamentos().subscribe(

      response => {

        this.departamentos = response;

        console.log("Todo OK");

      },

      error => {

        console.log(error);

      }

    );

  }

  ngOnInit(): void {

    this.cargarDepartamentos();

  }

}

Agregamos un componente llamado InsertarDepartamento

**INSERTARDEPARTAMENTO.COMPONENT.HTML**

<form #insertForm="ngForm">

  <div class="form-group" style="width:50%">

    <label>Número: </label>

    <input type="number" class="form-control" #cajanumero />

  </div>

  <div class="form-group" style="width:50%">

    <label>Número: </label>

    <input type="text" class="form-control" #cajanombre />

  </div>

  <div class="form-group" style="width:50%">

    <label>Número: </label>

    <input type="text" class="form-control" #cajalocalidad />

  </div>

  <button (click)="insertarDepartamento()" class="btn-info">

    Insertar departamento

  </button>

</form>

**INSERTARDEPARTAMENTO.COMPONENT.TS**

import { Component, OnInit, ViewChild, ElementRef } from "@angular/core";

import { DepartamentoService } from "./../../services/departamento.service";

import { Router, ActivatedRoute, Params } from "@angular/router";

@Component({

  selector: "app-insertar-departamento",

  templateUrl: "./insertar-departamento.component.html",

  styleUrls: ["./insertar-departamento.component.css"],

  providers: [DepartamentoService]

})

export class InsertarDepartamentoComponent implements OnInit {

  @ViewChild("cajanumero") cajanumero: ElementRef;

  @ViewChild("cajanombre") cajanombre: ElementRef;

  @ViewChild("cajalocalidad") cajalocalidad: ElementRef;

  insertarDepartamento() {

    var numero = this.cajanumero.nativeElement.value;

    var nombre = this.cajanombre.nativeElement.value;

    var localidad = this.cajalocalidad.nativeElement.value;

    var departamento = {

      numero: numero,

      nombre: nombre,

      localidad: localidad

    };

    this.\_departamentoService.insertarDepartamento(departamento).subscribe(

      response => {

        this.\_router.navigate(["/"]);

      },

      error => {

        console.log(error);

      }

    );

  }

  constructor(

    private \_departamentoService: DepartamentoService,

    private \_router: Router

  ) {}

  ngOnInit(): void {}

}

Creamos un componente llamado **DetallesDepartamento**

**DETALLESDEPARTAMENTO.COMPONENT.HTML**

<h1>{{ departamento.numero }}</h1>

<h1>{{ departamento.nombre }}</h1>

<h1>{{ departamento.localidad }}</h1>

<form #deleteForm="ngForm">

  <div class="form-group" style="width:50%">

    <input type="hidden" class="form-control" #cajanumero />

  </div>

  <button

    (click)="eliminarDepartamento(departamento.numero)"

    class="btn btn-danger"

  >

    Eliminar departamento

  </button>

</form>

**DETALLESDEPARTAMENTO.COMPONENT.TS**

import { Component, OnInit } from "@angular/core";

import { DepartamentoService } from "./../../services/departamento.service";

import { Router, ActivatedRoute, Params } from "@angular/router";

import { Departamento } from "./../../models/departamento";

@Component({

  selector: "app-detalles-departamento",

  templateUrl: "./detalles-departamento.component.html",

  styleUrls: ["./detalles-departamento.component.css"],

  providers: [DepartamentoService]

})

export class DetallesDepartamentoComponent implements OnInit {

  public departamento: Departamento;

  //private numero: number;

  constructor(

    private \_departamentoService: DepartamentoService,

    private \_route: ActivatedRoute,

    private \_router: Router

  ) {}

  buscarDepartamento(deptno) {

    this.\_departamentoService.buscarDepartamento(deptno).subscribe(

      response => {

        this.departamento = response;

      },

      error => {

        console.log(error);

      }

    );

  }

  eliminarDepartamento(deptno) {

    this.\_departamentoService.eliminarDepartamento(deptno).subscribe(

      response => {

        this.\_router.navigate(["/"]);

      },

      error => {

        console.log(error);

      }

    );

  }

  ngOnInit(): void {

    //SE TRABAJA CON OBSERVABLE PORQUE

    //PODRIAN LLEGAR PARAMETROS QUE TARDASEN

    //EN SER RECIBIDOS

    this.\_route.params.subscribe((params: Params) => {

      //COMPROBAMOS QUE TENEMOS PARÁMETROS EN LA PAGINA

      if (params.iddepartamento != null) {

        //SI TENEMOS NUMERO, ASIGNAMOS LA PROPIEDAD NUMERO

        //DE LA CLASE AL VALOR DEL PARAMETRO

        //A PESAR DE QUE SEA DECLARADO COMO

        //NUMERO EN TYPE SCRIPT, SIGUE VINIENDO

        //UN STRING, POR LO QUE CONVERTIMOS

        var numero = params.iddepartamento;

        this.buscarDepartamento(numero);

      } else {

        console.log("No hay parámetros");

      }

    });

  }

}

Creamos un componente llamado **ModificarDepartamento**

**MODIFICARDEPARTAMENTO.COMPONENT.HTML**

<form #updateForm="ngForm">

  <div class="form-group" style="width:50%">

    <label>Número: </label>

    <input

      disabled

      type="number"

      class="form-control"

      #cajanumero

      value="{{ departamento.numero }}"

    />

  </div>

  <div class="form-group" style="width:50%">

    <label>Número: </label>

    <input

      type="text"

      class="form-control"

      #cajanombre

      value="{{ departamento.nombre }}"

    />

  </div>

  <div class="form-group" style="width:50%">

    <label>Número: </label>

    <input

      type="text"

      class="form-control"

      #cajalocalidad

      value="{{ departamento.localidad }}"

    />

  </div>

  <button (click)="modificarDepartamento()" class="btn btn-success">

    Modificar departamento

  </button>

</form>

**MODIFICARDEPARTAMENTO.COMPONENT.TS**

import { Component, OnInit, ViewChild, ElementRef } from "@angular/core";

import { DepartamentoService } from "./../../services/departamento.service";

import { Router, ActivatedRoute, Params } from "@angular/router";

import { Departamento } from "src/app/models/departamento";

@Component({

  selector: "app-modificar-departamento",

  templateUrl: "./modificar-departamento.component.html",

  styleUrls: ["./modificar-departamento.component.css"],

  providers: [DepartamentoService]

})

export class ModificarDepartamentoComponent implements OnInit {

  public departamento: Departamento;

  @ViewChild("cajanumero") cajanumero: ElementRef;

  @ViewChild("cajanombre") cajanombre: ElementRef;

  @ViewChild("cajalocalidad") cajalocalidad: ElementRef;

  constructor(

    private \_departamentoService: DepartamentoService,

    private \_route: ActivatedRoute,

    private \_router: Router

  ) {}

  modificarDepartamento() {

    var numero = this.cajanumero.nativeElement.value;

    var nombre = this.cajanombre.nativeElement.value;

    var localidad = this.cajalocalidad.nativeElement.value;

    var departamento = {

      numero: numero,

      nombre: nombre,

      localidad: localidad

    };

    this.\_departamentoService.modificarDepartamento(departamento).subscribe(

      response => {

        this.\_router.navigate(["/detalles/" + departamento.numero]);

      },

      error => {

        console.log(error);

      }

    );

  }

  ngOnInit(): void {

    //SE TRABAJA CON OBSERVABLE PORQUE

    //PODRIAN LLEGAR PARAMETROS QUE TARDASEN

    //EN SER RECIBIDOS

    this.\_route.params.subscribe((params: Params) => {

      //COMPROBAMOS QUE TENEMOS PARÁMETROS EN LA PAGINA

      if (params.numero != null) {

        console.log("Numero: " + params.numero);

        //SI TENEMOS NUMERO, ASIGNAMOS LA PROPIEDAD NUMERO

        //DE LA CLASE AL VALOR DEL PARAMETRO

        //A PESAR DE QUE SEA DECLARADO COMO

        //NUMERO EN TYPE SCRIPT, SIGUE VINIENDO

        //UN STRING, POR LO QUE CONVERTIMOS

        var numero = params.numero;

        var nombre = params.nombre;

        var localidad = params.localidad;

        this.departamento = {

          numero: numero,

          nombre: nombre,

          localidad: localidad

        };

        console.log(this.departamento);

      } else {

        console.log("No hay parámetros");

      }

    });

  }

}

Modificamos la clase **Routing**

//IMPORTAMOS LOS MODULOS DEL ROUTER DE ANGULAR

import { ModuleWithProviders } from "@angular/core";

import { Routes, RouterModule } from "@angular/router";

//AHORA DEBEMOS IMPORTAR LOS COMPONENTES QUE VAYAN A FORMAR PARTE

//DE LA NAVEGACION, ES DECIR, LOS QUE VISUALIZAREMOS

//DENTRO DE LAS ZONAS DE NUESTRO SITE

//IMPORTAMOS EL COMPONENTE DE NUMERO DOBLE

import { DepartamentosComponent } from "./components/departamentos/departamentos.component";

import { InsertarDepartamentoComponent } from "./components/insertar-departamento/insertar-departamento.component";

import { DetallesDepartamentoComponent } from "./components/detalles-departamento/detalles-departamento.component";

import { ModificarDepartamentoComponent } from "./components/modificar-departamento/modificar-departamento.component";

//DEFINIMOS EL ARRAY DE RUTAS, ES DECIR, LA CONFIGURACION DE TODAS

//LAS RUTAS PARA NUESTRA APP, QUE SERA NUMERO DOBLE SIN PARAMETROS Y CON //PARAMETRO OPCIONAL.

const appRoutes: Routes = [

  { path: "", component: DepartamentosComponent },

  { path: "insertar", component: InsertarDepartamentoComponent },

  {

    path: "detalles/:iddepartamento",

    component: DetallesDepartamentoComponent

  },

  {

    path: "update/:numero/:nombre/:localidad",

    component: ModificarDepartamentoComponent

  }

];

//AHORA EXPORTAMOS EL MODULO DEL ROUTER PARA PODER

//UTILIZARLO DENTRO DE ANGULAR

export const appRoutingProviders: any[] = [];

export const routing: ModuleWithProviders = RouterModule.forRoot(appRoutes);

Y realizamos el enlace dentro de la clase **Departamentos**

<table class="table table-bordered">

  <thead>

    <tr>

      <th>NUMERO</th>

      <th>NOMBRE</th>

      <th>LOCALIDAD</th>

      <th></th>

      <th></th>

    </tr>

  </thead>

  <tbody>

    <tr \*ngFor="let dept of departamentos">

      <td>{{ dept.numero }}</td>

      <td>{{ dept.nombre }}</td>

      <td>{{ dept.localidad }}</td>

      <td>

        <a href="detalles/{{ dept.numero }}">

          Detalles

        </a>

      </td>

      <td>

        <a

          href="update/{{ dept.numero }}/{{ dept.nombre }}/{{ dept.localidad }}"

          class="btn btn-info"

        >

          Edit

        </a>

      </td>

    </tr>

  </tbody>

</table>