**CURSO MEAN**

**FRONT-END**

# Instalación y entorno de desarrollo

## Instalar Nodejs

https://nodejs.org/es/download/

## Instalación de Angular5

Utilizar Angular CLI es muy sencillo, simplemente lo instalamos con este comando:

npm install -g @angular/cli

Y podemos generar una instalación de Angular 4 de esta forma:

ng new sueldoApp

cd sueldoApp

/\*

Creara el node modules

Npm install

Npm update

\*/

ng serve

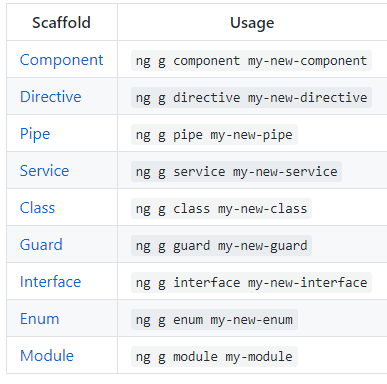
\*\* abrimos desde nuestro editor de código (visual estudio code)

https://code.visualstudio.com/download

# Componentes

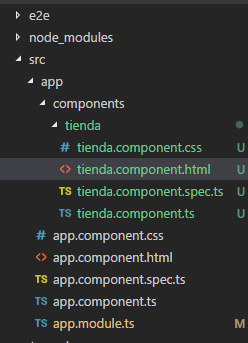
## Crear componentes, plantillas y estilos

<https://github.com/angular/angular-cli>



* Crear componentes en Angular
* ng generate component my-new-component
* ng g component my-new-component # using the alias

ng generate component component/tienda



tienda.component.ts

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

@Component({

selector: 'app-tienda',

templateUrl: './tienda.component.html',

styleUrls: ['./tienda.component.css']

})

export class TiendaComponent implements OnInit {

public titulo;

constructor() { }

ngOnInit() {

this.titulo = 'TIENDA ITEAMDEVS';

}

}

tienda.component.html

<p>

{{titulo}}

</p>

Se actualiza también:

app.module.ts

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

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

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

import { HttpModule } from '@angular/http';

// componentes para que esten dismonibles en toda la aplicacion

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

import { TiendaComponent } from './components/tienda/tienda.component';

@NgModule({

declarations: [

AppComponent,

TiendaComponent

],

imports: [

BrowserModule,

FormsModule,

HttpModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

## Creación de varios componentes

Ng generate component components/parque

Esto me genera un nuevo componete y agregara los import en app.modules y finalmente lo podemos llamar en el componente principal app.component.html

Parque.component.ts

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

@Component({

selector: 'app-parque',

templateUrl: './parque.component.html',

styleUrls: ['./parque.component.css']

})

export class ParqueComponent implements OnInit {

public nombre: string;

public metros: number;

public vegetacion: string;

public abierto: boolean;

constructor() {

this.nombre = 'parque de caballos';

this.metros = 450;

this.vegetacion = 'Alta';

this.abierto = true;

}

ngOnInit() {

}

}

Parque.component.html

<h2>{{nombre}}</h2>

<ol>

<li>Metros: {{metros}}</li>

<li>Vegetacion: {{vegetacion}}</li>

<li>

<span \*ngIf="abierto == true">Abierto</span>

<span \*ngIf="abierto == false">Cerrado</span>

</li>

</ol>

App.module.ts

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

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

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

import { HttpModule } from '@angular/http';

// componentes para que esten dismonibles en toda la aplicacion

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

import { TiendaComponent } from './components/tienda/tienda.component';

import { ParqueComponent } from './components/parque/parque.component';

@NgModule({

declarations: [

AppComponent,

TiendaComponent,

ParqueComponent

],

imports: [

BrowserModule,

FormsModule,

HttpModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

una vez registrado en app.modules.ts ya lo podemos utilizar en cualquier otro componete

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center">

<h1>

Welcome to {{ title }}!

</h1>

<img width="300" alt="Angular Logo" src="data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciIHZpZXdCb3g9IjAgMCAyNTAgMjUwIj4KICAgIDxwYXRoIGZpbGw9IiNERDAwMzEiIGQ9Ik0xMjUgMzBMMzEuOSA2My4ybDE0LjIgMTIzLjFMMTI1IDIzMGw3OC45LTQzLjcgMTQuMi0xMjMuMXoiIC8+CiAgICA8cGF0aCBmaWxsPSIjQzMwMDJGIiBkPSJNMTI1IDMwdjIyLjItLjFWMjMwbDc4LjktNDMuNyAxNC4yLTEyMy4xTDEyNSAzMHoiIC8+CiAgICA8cGF0aCAgZmlsbD0iI0ZGRkZGRiIgZD0iTTEyNSA1Mi4xTDY2LjggMTgyLjZoMjEuN2wxMS43LTI5LjJoNDkuNGwxMS43IDI5LjJIMTgzTDEyNSA1Mi4xem0xNyA4My4zaC0zNGwxNy00MC45IDE3IDQwLjl6IiAvPgogIDwvc3ZnPg==">

</div>

<app-tienda></app-tienda>

<hr>

<app-parque></app-parque>

<hr>



# Comunicación entre componentes

## Inputs

Sirve para comunicar componentes por medio de directivas parametrizadas estática o dinámicamente( comunicación de componente padre a componente hijo 🡪)

Ejemplo: componente tienda a componente parque

Tienda.component.ts

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

@Component({

selector: 'app-tienda',

templateUrl: './tienda.component.html',

styleUrls: ['./tienda.component.css']

})

export class TiendaComponent implements OnInit {

public titulo;

public nombreParque: string;

constructor() {

this.titulo = 'TIENDA ITEAMDEVS';

}

mostrarNombre() {

console.log(this.nombreParque);

}

ngOnInit() {

}

}

Tienda.componet.html

PADRE

<p>

{{titulo}}

</p>

<hr>

Nombre del parque:

<input type="text" [(ngModel)] = "nombreParque" (keyup)="mostrarNombre()" />

mostrar: {{nombreParque}}

<hr>

COMPONENTE

<div \*ngIf="nombreParque">

<app-parque [nombre] = "nombreParque" [ferestal]="'mala'"></app-parque>

</div>

<hr>

Parque.component.ts

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

@Component({

selector: 'app-parque',

templateUrl: './parque.component.html',

styleUrls: ['./parque.component.css']

})

export class ParqueComponent implements OnInit {

@Input() nombre: string;

public metros: number;

@Input('ferestal') vegetacion: string;

abierto: boolean;

constructor() {

this.nombre = 'nombre de parque';

this.metros = 450;

this.vegetacion = 'Alta';

this.abierto = true;

}

ngOnInit() {

}

}

Parque componet.html

<h2>nombre: {{nombre}}</h2>

<ol>

<li>Metros: {{metros}}</li>

<li>Vegetacion: {{vegetacion}}</li>

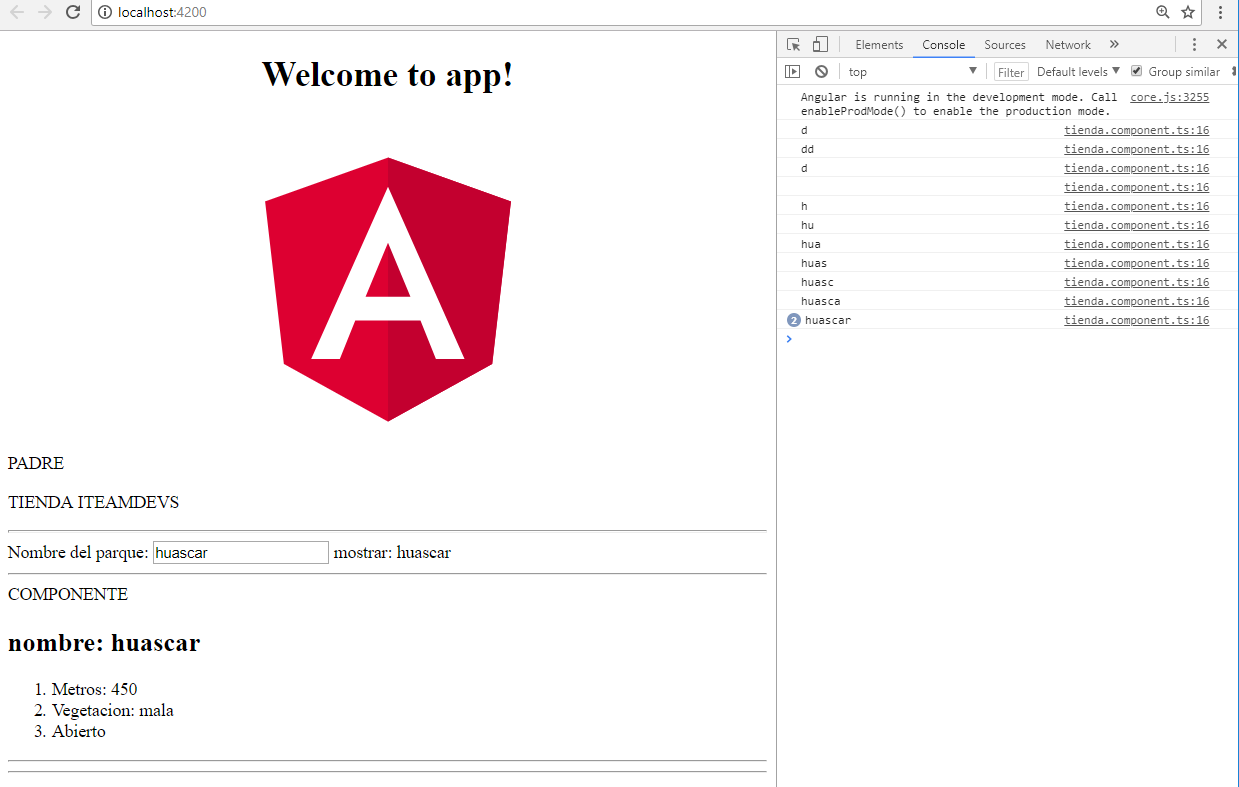
<li>

<span \*ngIf="abierto == true">Abierto</span>

<span \*ngIf="abierto == false">Cerrado</span>

</li>

</ol>



## Outputs

Comunicación inversa de un componente hico a un componete padre (<- )

Parque.componet.ts

import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';

@Component({

selector: 'app-parque',

templateUrl: './parque.component.html',

styleUrls: ['./parque.component.css']

})

export class ParqueComponent implements OnInit {

@Input() nombre: string;

public metros: number;

@Input('ferestal') vegetacion: string;

abierto: boolean;

@Output() pasarDatos = new EventEmitter();

constructor() {

this.nombre = 'nombre de parque';

this.metros = 450;

this.vegetacion = 'Alta';

this.abierto = true;

}

emitirEvento() {

this.pasarDatos.emit({

'nombre': this.nombre ,

'metros': this.metros ,

'vegetacion': this.vegetacion ,

'abierto': this.abierto

});

}

**Parque.componet.html**

<h2>nombre: {{nombre}}</h2>

<ol>

<li>Metros: {{metros}}</li>

<li>Vegetacion: {{vegetacion}}</li>

<li>

<span \*ngIf="abierto == true">Abierto</span>

<span \*ngIf="abierto == false">Cerrado</span>

</li>

</ol>

<button (click)="emitirEvento()">Mostrar en componente padre</button>

**Tienda.component.html**

COMPONENTE PADRE

<p>

{{titulo}}

</p>

<hr>

Nombre del parque:

<input type="text" [(ngModel)] = "nombreParque" (keyup)="mostrarNombre()" />

mostrar: {{nombreParque}}

<div \*ngIf="miParque">

Datos recibidos desde componente hijo:

<ul>

<li>{{miParque.nombre}}</li>

<li>{{miParque.metros}}</li>

<li>{{miParque.vegetacion}}</li>

<li>{{miParque.abierto}}</li>

</ul>

</div>

<hr>

COMPONENTE HIJO

<div \*ngIf="nombreParque">

<app-parque [nombre] = "nombreParque"

[ferestal]= "'mala'"

(pasarDatos)= "verDatosParque($event)"

></app-parque>

</div>

<hr>

**Tienda.component.ts**

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

@Component({

selector: 'app-tienda',

templateUrl: './tienda.component.html',

styleUrls: ['./tienda.component.css']

})

export class TiendaComponent implements OnInit {

public titulo;

public nombreParque: string;

// variable para recibir datos de componente hijo

public miParque;

constructor() {

this.titulo = 'TIENDA ITEAMDEVS';

}

mostrarNombre() {

console.log(this.nombreParque);

}

verDatosParque(event) {

// console.log(event);

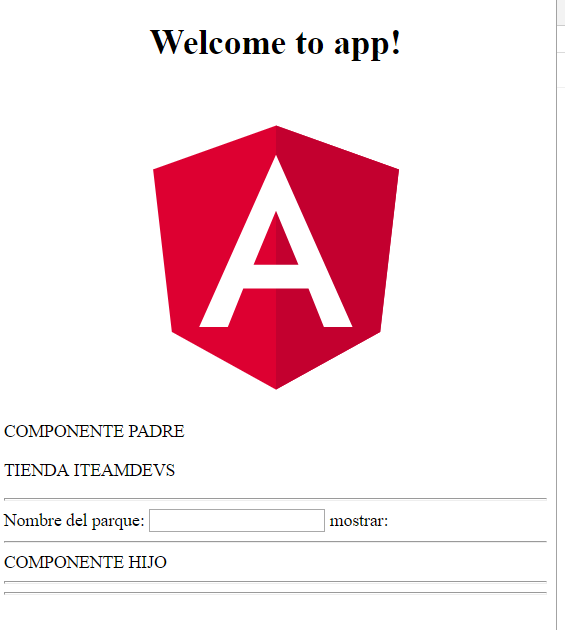
this.miParque = event;

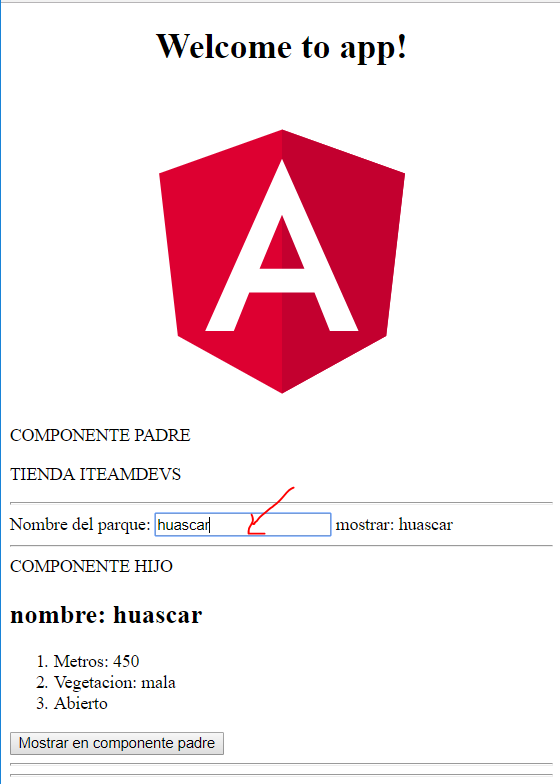
}

ngOnInit() {

}

}







# Hooks

Los hooks son eventos que se ejecutan en un momento dado del ciclo de vida del componente, es decir, son eventos que se van a lanzar dependiendo del estado del componente. Por ejemplo, cuando iniciemos el componente, cuando hagamos algún cambio en el componente, etc.

* https://angular.io/guide/lifecycle-hooks

## Onchanges

Este método va a ser el primero que se va a ejecutar, se va a ejecutar el primero de todos y también se va a ejecutar cuando se produce algún cambio en los valores de las propiedades de nuestro componente.

Ejemplo:

import { Component, OnInit, Input, Output, EventEmitter, OnChanges, SimpleChanges } from '@angular/core';

@Component({

selector: 'app-parque',

templateUrl: './parque.component.html',

styleUrls: ['./parque.component.css']

})

export class ParqueComponent implements OnInit OnChanges{

@Input() nombre: string;

public metros: number;

@Input('ferestal') vegetacion: string;

abierto: boolean;

@Output() pasarDatos = new EventEmitter();

constructor() {

this.nombre = 'nombre de parque';

this.metros = 450;

this.vegetacion = 'Alta';

this.abierto = true;

}

ngOnChanges(changes: SimpleChanges) {

console.log(changes);

}

ngOnInit() {

}

emitirEvento() {

this.pasarDatos.emit({

'nombre': this.nombre ,

'metros': this.metros ,

'vegetacion': this.vegetacion ,

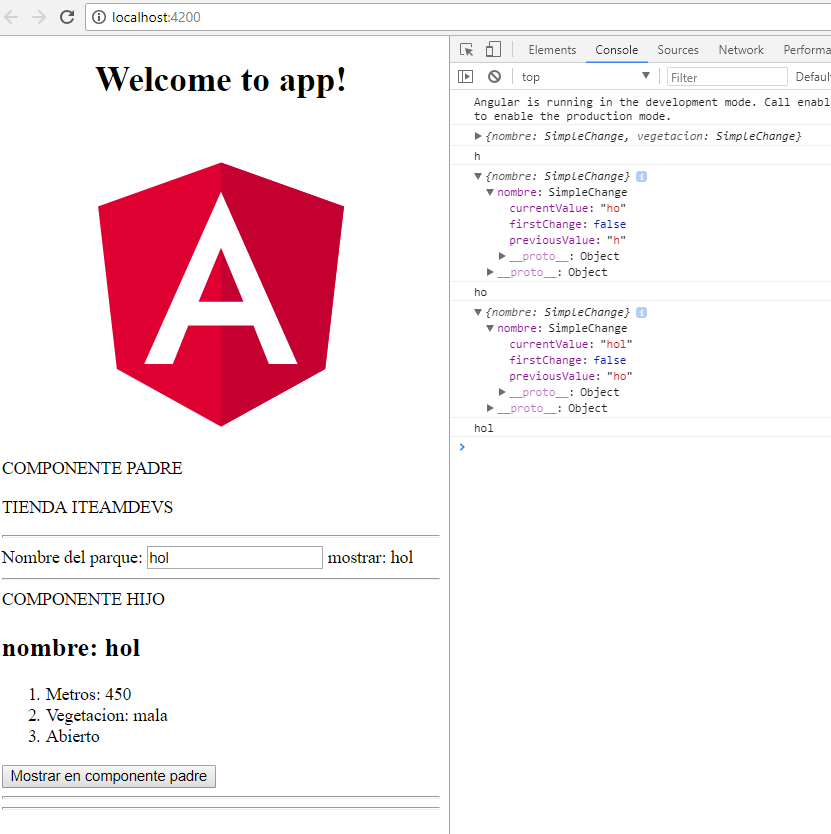
'abierto': this.abierto

});

}

}

Resultado:



## onInit

El hook ngOnInit se ejecuta cuando cargamos la directiva de nuestro componente, es el primer método que se ejecuta después de lanzar el constructor de la clase del componente.

Ejemplo:

import { Component, Input, Output, EventEmitter, OnChanges, SimpleChanges, OnInit } from '@angular/core';

@Component({

selector: 'app-parque',

templateUrl: './parque.component.html',

styleUrls: ['./parque.component.css']

})

export class ParqueComponent implements OnInit OnChanges{

@Input() nombre: string;

public metros: number;

@Input('ferestal') vegetacion: string;

abierto: boolean;

@Output() pasarDatos = new EventEmitter();

constructor() {

this.nombre = 'nombre de parque';

this.metros = 450;

this.vegetacion = 'Alta';

this.abierto = true;

}

ngOnChanges(changes: SimpleChanges) {

console.log(changes);

}

ngOnInit() {

console.log('metodo onInit lanzado');

}

emitirEvento() {

this.pasarDatos.emit({

'nombre': this.nombre ,

'metros': this.metros ,

'vegetacion': this.vegetacion ,

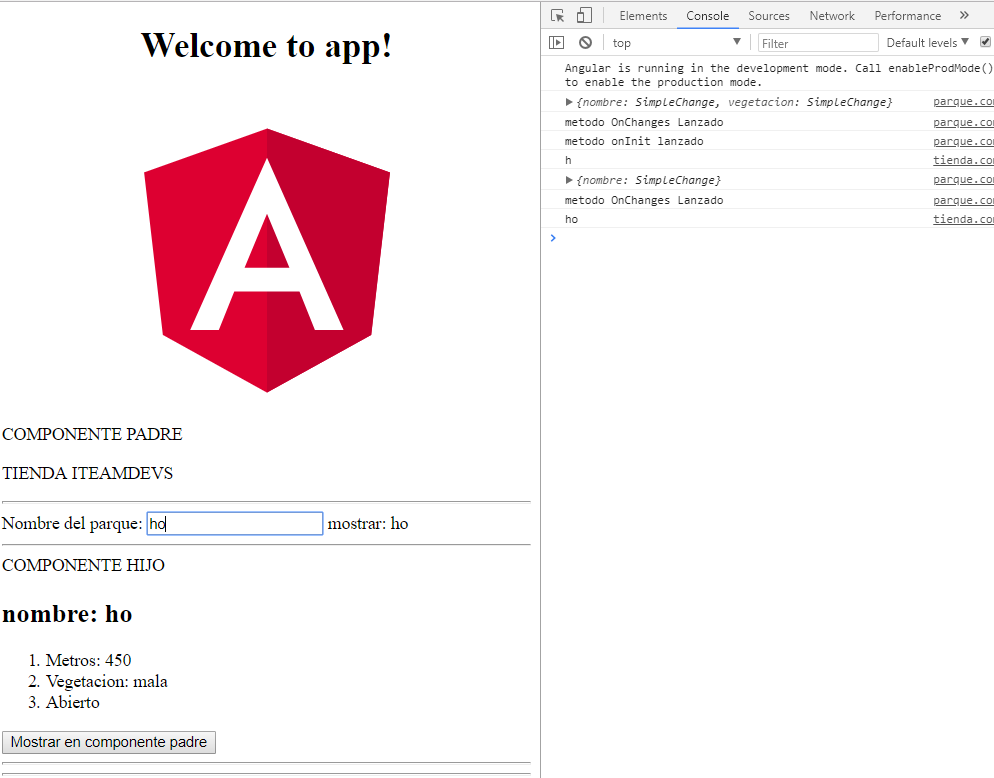
'abierto': this.abierto

});

}

}

Resultado:



## DoCheck

El método DoCheck se ejecuta cada vez que se produce algún cambio en el componente o en la aplicación de Angular y es un método que se va a estar ejecutando frecuentemente.

Ejemplo:

import { Component, Input, Output, EventEmitter, OnChanges, SimpleChanges, OnInit, DoCheck } from '@angular/core';

@Component({

selector: 'app-parque',

templateUrl: './parque.component.html',

styleUrls: ['./parque.component.css']

})

export class ParqueComponent implements OnInit, OnChanges, DoCheck {

@Input() nombre: string;

public metros: number;

@Input('ferestal') vegetacion: string;

abierto: boolean;

@Output() pasarDatos = new EventEmitter();

constructor() {

this.nombre = 'nombre de parque';

this.metros = 450;

this.vegetacion = 'Alta';

this.abierto = true;

}

ngOnChanges(changes: SimpleChanges) {

console.log(changes);

console.log('metodo OnChanges Lanzado');

}

ngOnInit() {

console.log('metodo onInit lanzado');

}

ngDoCheck() {

console.log('metodo DoCheck Lanzado');

}

emitirEvento() {

this.pasarDatos.emit({

'nombre': this.nombre ,

'metros': this.metros ,

'vegetacion': this.vegetacion ,

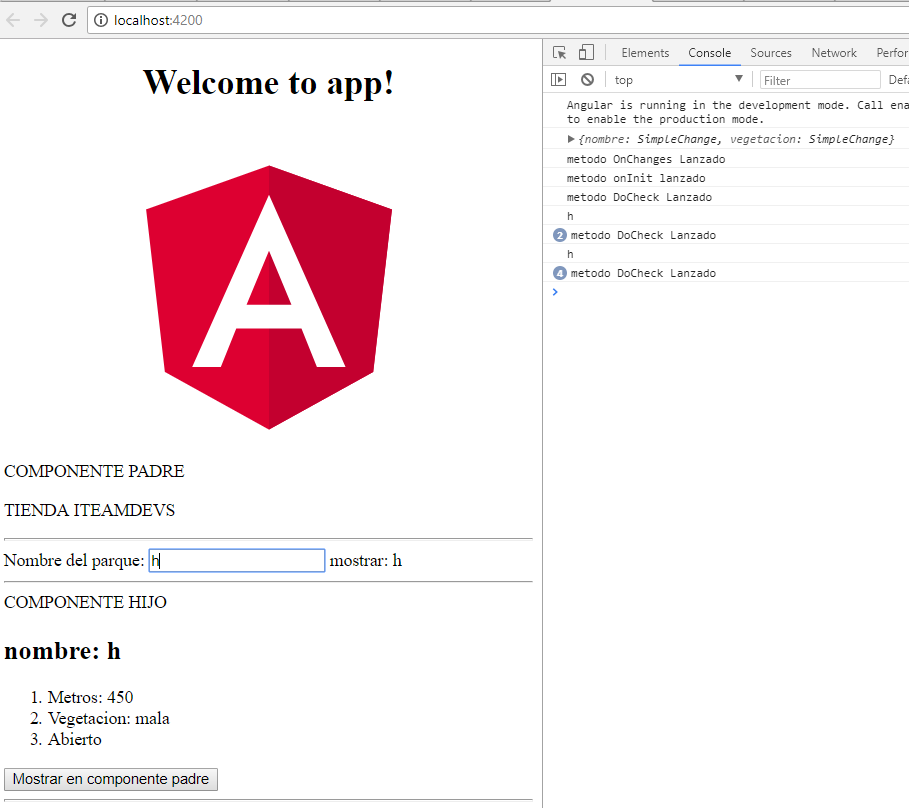
'abierto': this.abierto

});

}

}

Resultado:



## OnDestroy

Además tenemos otro evento Hook en Angular 4 que se ejecuta justo antes de eliminar la instancia de un componente, este es el método ngOnDestroy:

**Ejempo:**

import { Component, Input, Output, EventEmitter, OnChanges, SimpleChanges, OnInit, DoCheck, OnDestroy } from '@angular/core';

@Component({

selector: 'app-parque',

templateUrl: './parque.component.html',

styleUrls: ['./parque.component.css']

})

export class ParqueComponent implements OnInit, OnChanges, DoCheck, OnDestroy {

@Input() nombre: string;

public metros: number;

@Input('ferestal') vegetacion: string;

abierto: boolean;

@Output() pasarDatos = new EventEmitter();

constructor() {

this.nombre = 'nombre de parque';

this.metros = 450;

this.vegetacion = 'Alta';

this.abierto = true;

}

ngOnChanges(changes: SimpleChanges) {

console.log(changes);

console.log('metodo OnChanges Lanzado');

}

ngOnInit() {

console.log('metodo onInit lanzado');

}

ngDoCheck() {

console.log('metodo DoCheck Lanzado');

}

ngOnDestroy() {

console.log('metodo onDestroy Lanzado');

}

emitirEvento() {

this.pasarDatos.emit({

'nombre': this.nombre ,

'metros': this.metros ,

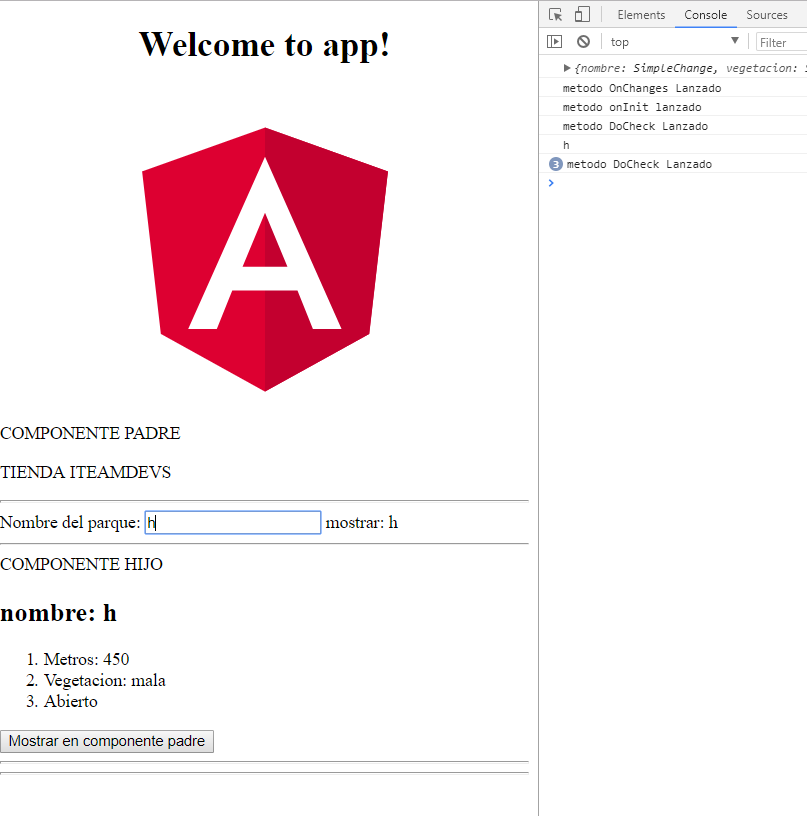
'vegetacion': this.vegetacion ,

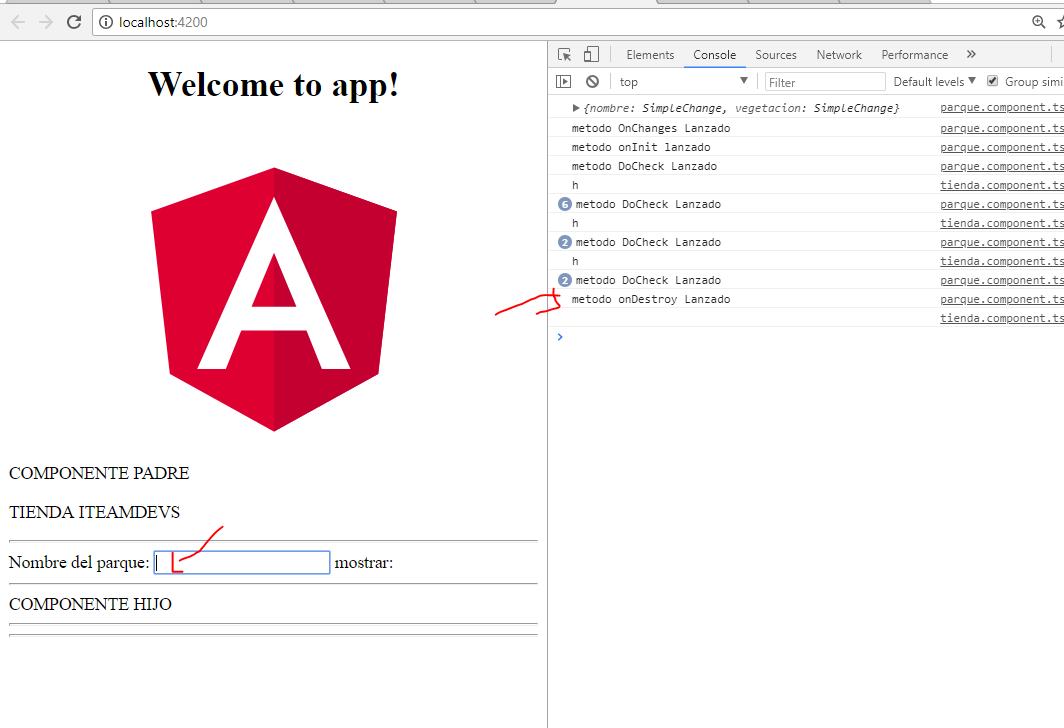
'abierto': this.abierto

});

}

}





# Routing y paginas

## Configurar el sistema de rutas

Creamos app.routing.ts

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

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

// componentes

import { TiendaComponent } from './components/tienda/tienda.component';

const appRoutes: Routes = [

{ path: '', component: TiendaComponent },

{ path: '', redirectTo: 'tienda', pathMatch: 'full' },

{ path: 'tienda', component: TiendaComponent },

{ path: '\*\*', component: TiendaComponent }

];

// \*\* es pagina 404 cuando no existe lapagina

export const appRoutingProviders: any[] = [];

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

configuramos en app.modules.ts

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

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

// importamos el routing

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

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

import { HttpModule } from '@angular/http';

// componentes para que esten dismonibles en toda la aplicacion

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

import { TiendaComponent } from './components/tienda/tienda.component';

import { ParqueComponent } from './components/parque/parque.component';

@NgModule({

declarations: [

AppComponent,

TiendaComponent,

ParqueComponent

],

imports: [

BrowserModule,

FormsModule,

HttpModule,

routing

],

providers: [

appRoutingProviders

],

bootstrap: [AppComponent]

})

export class AppModule { }

ahora ya podemos navegar, pero para que reconosca dinámicamente las rutas modificamos app.component.html

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center">

<h1>

Welcome to {{ title }}!

</h1>

<img width="300" alt="Angular Logo" src="data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciIHZpZXdCb3g9IjAgMCAyNTAgMjUwIj4KICAgIDxwYXRoIGZpbGw9IiNERDAwMzEiIGQ9Ik0xMjUgMzBMMzEuOSA2My4ybDE0LjIgMTIzLjFMMTI1IDIzMGw3OC45LTQzLjcgMTQuMi0xMjMuMXoiIC8+CiAgICA8cGF0aCBmaWxsPSIjQzMwMDJGIiBkPSJNMTI1IDMwdjIyLjItLjFWMjMwbDc4LjktNDMuNyAxNC4yLTEyMy4xTDEyNSAzMHoiIC8+CiAgICA8cGF0aCAgZmlsbD0iI0ZGRkZGRiIgZD0iTTEyNSA1Mi4xTDY2LjggMTgyLjZoMjEuN2wxMS43LTI5LjJoNDkuNGwxMS43IDI5LjJIMTgzTDEyNSA1Mi4xem0xNyA4My4zaC0zNGwxNy00MC45IDE3IDQwLjl6IiAvPgogIDwvc3ZnPg==">

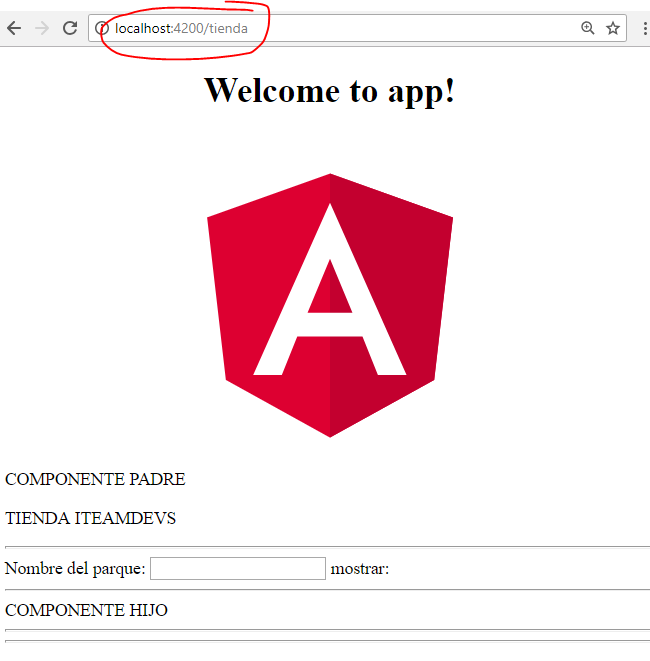
</div>

<!--app-tienda></app-tienda-->

<router-outlet></router-outlet>

<hr>

Resultado:



## Router y componentes

Haremos un aplicatifo de zoológico

Crear los siguientes componentes con angular cli

* Ng generate component components/home
* Ng generate component components/animals
* Ng generate component components/contact
* Ng generate component components/keeper

Hacemos los siguientes campos en cada componente:

[Home.component.ts](http://Home.component.ts)

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

@Component({

selector: 'app-home',

templateUrl: './home.component.html',

styleUrls: ['./home.component.css']

})

export class HomeComponent implements OnInit {

title = 'COMPONENTE HOME';

constructor() { }

ngOnInit() {

console.log('comopente home cargado');

}

}

[Home.component.html](http://Home.component.html)

<p>

{{title}}

</p>

Al final modificamos app.modules.ts

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

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

// importamos el routing

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

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

import { HttpModule } from '@angular/http';

// componentes para que esten dismonibles en toda la aplicacion

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

import { TiendaComponent } from './components/tienda/tienda.component';

import { ParqueComponent } from './components/parque/parque.component';

import { HomeComponent } from './components/home/home.component';

import { AnimalsComponent } from './components/animals/animals.component';

import { ContactComponent } from './components/contact/contact.component';

import { KeeperComponent } from './components/keeper/keeper.component';

import { NotfoundComponent } from './components/notfound/notfound.component';

@NgModule({

declarations: [

AppComponent,

TiendaComponent,

ParqueComponent,

HomeComponent,

AnimalsComponent,

ContactComponent,

KeeperComponent,

NotfoundComponent,

],

imports: [

BrowserModule,

FormsModule,

HttpModule,

routing

],

providers: [

appRoutingProviders

],

bootstrap: [AppComponent]

})

export class AppModule { }

y también configuramos app.routing.ts

import { HomeComponent } from './components/home/home.component';

import { AnimalsComponent } from './components/animals/animals.component';

import { ContactComponent } from './components/contact/contact.component';

import { KeeperComponent } from './components/keeper/keeper.component';

import { NotfoundComponent } from './components/notfound/notfound.component';

const appRoutes: Routes = [

{ path: '', component: HomeComponent },

{ path: '', redirectTo: 'home', pathMatch: 'full' },

{ path: 'home', component: HomeComponent },

{ path: '404', component: NotfoundComponent },

{ path: 'animales', component: AnimalsComponent },

{ path: 'contacto', component: ContactComponent },

{ path: 'cuidadores', component: KeeperComponent },

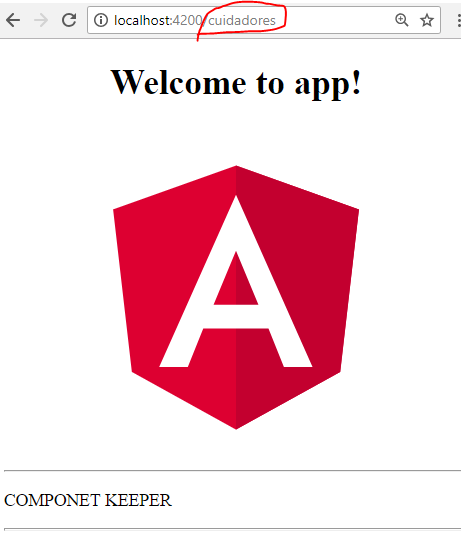
];

// \*\* es pagina 404 cuando no existe lapagina

export const appRoutingProviders: any[] = [];

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

resultado: ya podemos navegar por los componentes



## Navegación entre rutas

Vamos a crear nuestro menú de navegación:

En app.component.html

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center">

<h1>

{{ title }}!

</h1>

<img width="300" alt="Angular Logo" src="data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciIHZpZXdCb3g9IjAgMCAyNTAgMjUwIj4KICAgIDxwYXRoIGZpbGw9IiNERDAwMzEiIGQ9Ik0xMjUgMzBMMzEuOSA2My4ybDE0LjIgMTIzLjFMMTI1IDIzMGw3OC45LTQzLjcgMTQuMi0xMjMuMXoiIC8+CiAgICA8cGF0aCBmaWxsPSIjQzMwMDJGIiBkPSJNMTI1IDMwdjIyLjItLjFWMjMwbDc4LjktNDMuNyAxNC4yLTEyMy4xTDEyNSAzMHoiIC8+CiAgICA8cGF0aCAgZmlsbD0iI0ZGRkZGRiIgZD0iTTEyNSA1Mi4xTDY2LjggMTgyLjZoMjEuN2wxMS43LTI5LjJoNDkuNGwxMS43IDI5LjJIMTgzTDEyNSA1Mi4xem0xNyA4My4zaC0zNGwxNy00MC45IDE3IDQwLjl6IiAvPgogIDwvc3ZnPg==">

</div>

<hr>

<a [routerLink] = "['/home']">Inicio </a>

<a [routerLink] = "['/animales']">Animales </a>

<a [routerLink] = "['/contacto']">Contacto </a>

<a [routerLink] = "['/cuidadores']">Cuidadores </a>

<a [routerLink] = "['/tienda']">Tienda</a>

<hr>

<!--app-tienda></app-tienda-->

<router-outlet></router-outlet>

Resultado:



# Seciones y localStorage

## localStorage y Docheck

guardaremos una variable en localstorage en el componete contato y la recuperaremos en el componete principal app.component

contact.componet.html

<p>

{{title}}

</p>

<input type="text" [(ngModel)]="emailContacto">

<button (click)="guardarEmail()">Guardar Email</button>

Contact.component.ts

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

@Component({

selector: 'app-contact',

templateUrl: './contact.component.html',

styleUrls: ['./contact.component.css']

})

export class ContactComponent implements OnInit {

title = 'COMPONETE CONTAC';

public emailContacto: string;

constructor() { }

ngOnInit() {

console.log('componet contact cargado');

}

guardarEmail() {

// console.log(this.emailContacto);

// guardamos en localstorage

localStorage.setItem('emailContato', this.emailContacto);

// recupero del localstorage:

console.log(localStorage.getItem('emailContato'));

}

}

Hasta el momento ya se guargo y esta en la memoria local. Ahora la recuperaremos desde otro componete

App.componet.ts

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

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit, DoCheck {

title = 'ESCUELA FULL STACK';

emailContacto: string;

ngOnInit() {

console.log(localStorage.getItem('emailContato'));

this.emailContacto = localStorage.getItem('emailContato');

console.log('cargo email del localstorage: ' + this.emailContacto);

}

ngDoCheck() {

// para que actualise automaticamete sin recargar la pagina el email que se guarla en localstorage cuando se le ha cambiado

this.emailContacto = localStorage.getItem('emailContato');

}

}

App.component.html

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center">

<h1>

{{ title }}!

</h1>

<img width="300" alt="Angular Logo" src="data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciIHZpZXdCb3g9IjAgMCAyNTAgMjUwIj4KICAgIDxwYXRoIGZpbGw9IiNERDAwMzEiIGQ9Ik0xMjUgMzBMMzEuOSA2My4ybDE0LjIgMTIzLjFMMTI1IDIzMGw3OC45LTQzLjcgMTQuMi0xMjMuMXoiIC8+CiAgICA8cGF0aCBmaWxsPSIjQzMwMDJGIiBkPSJNMTI1IDMwdjIyLjItLjFWMjMwbDc4LjktNDMuNyAxNC4yLTEyMy4xTDEyNSAzMHoiIC8+CiAgICA8cGF0aCAgZmlsbD0iI0ZGRkZGRiIgZD0iTTEyNSA1Mi4xTDY2LjggMTgyLjZoMjEuN2wxMS43LTI5LjJoNDkuNGwxMS43IDI5LjJIMTgzTDEyNSA1Mi4xem0xNyA4My4zaC0zNGwxNy00MC45IDE3IDQwLjl6IiAvPgogIDwvc3ZnPg==">

</div>

<span \*ngIf="emailContacto"><strong> Email de contacto:</strong> {{ emailContacto }}</span>

<hr>

<a [routerLink] = "['/home']">Inicio </a>

<a [routerLink] = "['/animales']">Animales </a>

<a [routerLink] = "['/contacto']">Contacto </a>

<a [routerLink] = "['/cuidadores']">Cuidadores </a>

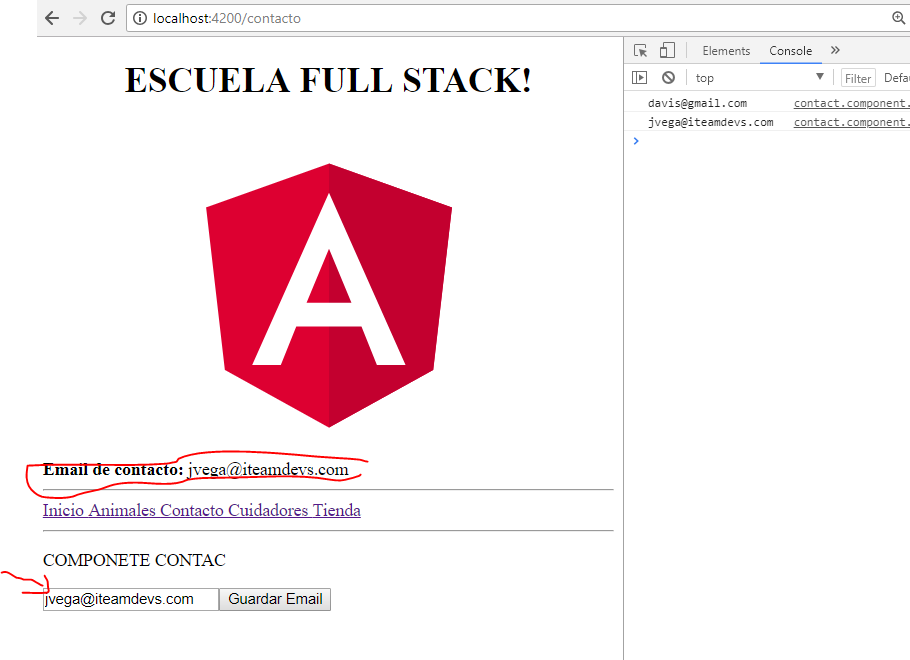
<a [routerLink] = "['/tienda']">Tienda</a>

<hr>

<!--app-tienda></app-tienda-->

<router-outlet></router-outlet>

Resultado:



## eliminar elementos de localStorage

creamos un botón para limpiar el localstirage:

app.componet.html

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center">

<h1>

{{ title }}!

</h1>

<img width="300" alt="Angular Logo" src="data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciIHZpZXdCb3g9IjAgMCAyNTAgMjUwIj4KICAgIDxwYXRoIGZpbGw9IiNERDAwMzEiIGQ9Ik0xMjUgMzBMMzEuOSA2My4ybDE0LjIgMTIzLjFMMTI1IDIzMGw3OC45LTQzLjcgMTQuMi0xMjMuMXoiIC8+CiAgICA8cGF0aCBmaWxsPSIjQzMwMDJGIiBkPSJNMTI1IDMwdjIyLjItLjFWMjMwbDc4LjktNDMuNyAxNC4yLTEyMy4xTDEyNSAzMHoiIC8+CiAgICA8cGF0aCAgZmlsbD0iI0ZGRkZGRiIgZD0iTTEyNSA1Mi4xTDY2LjggMTgyLjZoMjEuN2wxMS43LTI5LjJoNDkuNGwxMS43IDI5LjJIMTgzTDEyNSA1Mi4xem0xNyA4My4zaC0zNGwxNy00MC45IDE3IDQwLjl6IiAvPgogIDwvc3ZnPg==">

</div>

<span \*ngIf="emailContacto"><strong> Email de contacto:</strong> {{ emailContacto }}</span>

<button (click) = "borrarEmail()">Eliminar email de contacto</button>

<hr>

<a [routerLink] = "['/home']">Inicio </a>

<a [routerLink] = "['/animales']">Animales </a>

<a [routerLink] = "['/contacto']">Contacto </a>

<a [routerLink] = "['/cuidadores']">Cuidadores </a>

<a [routerLink] = "['/tienda']">Tienda</a>

<hr>

<!--app-tienda></app-tienda-->

<router-outlet></router-outlet>

App.componet.ts

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

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit, DoCheck {

title = 'ESCUELA FULL STACK';

emailContacto: string;

ngOnInit() {

console.log(localStorage.getItem('emailContato'));

this.emailContacto = localStorage.getItem('emailContato');

console.log('cargo email del localstorage: ' + this.emailContacto);

}

ngDoCheck() {

// para que actualise automaticamete sin recargar la pagina el email que se guarla en localstorage cuando se le ha cambiado

this.emailContacto = localStorage.getItem('emailContato');

}

borrarEmail() {

// remover solo un elemento del LS

localStorage.removeItem('emailContato');

// remover todo el LS

localStorage.clear();

// limpío la variable emailcontacto

this.emailContacto = null;

}

}

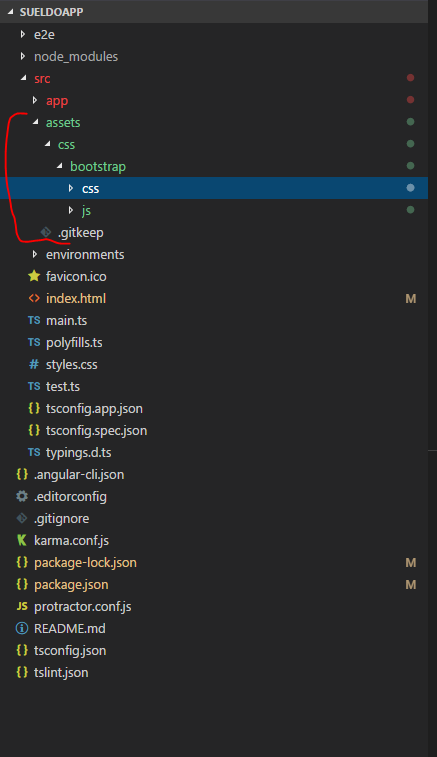




# Bootstrap y maquetación web

## Instalar Bootstrap

Descargamos las librerías de bootstrap y la pegamos en el directorio assets



En el archivo index.html linkeamos los estilos de Bootstrap

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>SueldoApp</title>

<base href="/">

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

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

<!-- CARGAMOS LIBRERIAS -->

<link rel="stylesheet" type="text/css" href="assets/css/bootstrap/css/bootstrap.min.css">

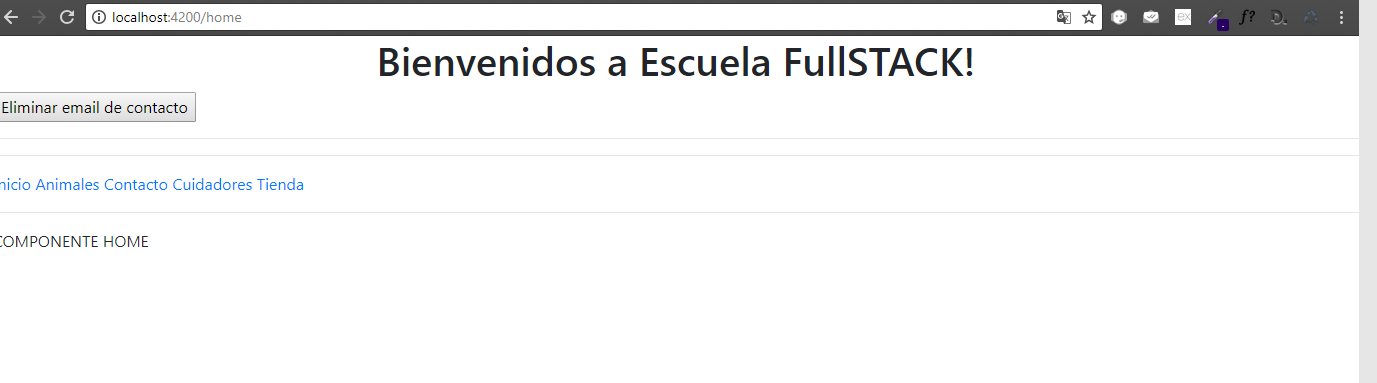
</head>

<body>

<app-root></app-root>

</body>

</html>



## Maquetar el menú

App.component.html

<!--The content below is only a placeholder and can be replaced.-->

<nav class="navbar navbar-expand-lg navbar-dark bg-dark " style="margin-top: 5px">

<div class="container">

<a class="navbar-brand" href="#">

<img src="https://escuelafullstack.com/web/image/res.company/1/logo?unique=fa956a0" width="30" height="30" alt="">

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavDropdown">

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" [routerLink]="['/home']">Inicio <span class="sr-only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/animales']">Animales</a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/cuidadores']">Cuidadores</a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/contacto']">Contacto</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

Tienda

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">

<a class="dropdown-item" href="#">SubMenu1</a>

<a class="dropdown-item" href="#">SubMenu2</a>

<a class="dropdown-item" href="#">SubMenu3</a>

</div>

</li>

</ul>

</div>

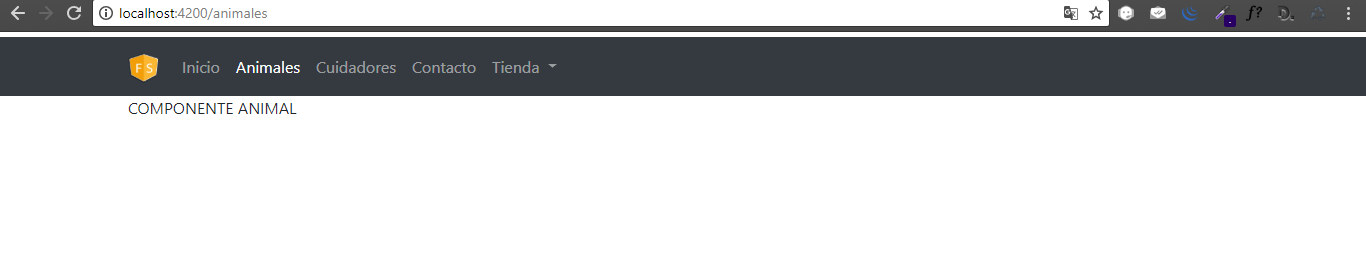
</div>

</nav>

<div class="container">

<router-outlet></router-outlet>

</div>



## Maquetacion de Home Page

[Home.component.html](http://Home.component.html)

<div class="jumbotron">

<h1 class="display-4">BIENVENIDO A {{title}}</h1>

<p class="lead">Controla los ingresos y egresos diarios</p>

<hr class="my-4">

<p>obten reportes de tus movimientos</p>

<p>Descubre en que gastas mas de lo devido</p>

</div>



## La ruta actual (routerLinkActive)

<!--The content below is only a placeholder and can be replaced.-->

<nav class="navbar navbar-expand-lg navbar-dark bg-dark " style="margin-top: 5px">

<div class="container">

<a class="navbar-brand" href="#">

<img src="https://escuelafullstack.com/web/image/res.company/1/logo?unique=fa956a0" width="30" height="30" alt="">

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavDropdown">

<ul class="navbar-nav">

<li class="nav-item ">

<a class="nav-link" [routerLink]="['/home']" [routerLinkActive]="['active']">Inicio </a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/animales']" [routerLinkActive]="['active']">Animales</a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/cuidadores']" [routerLinkActive]="['active']">Cuidadores</a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/contacto']" [routerLinkActive]="['active']">Contacto</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" [routerLinkActive]="['active']">

Tienda

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">

<a class="dropdown-item" href="#">SubMenu1</a>

<a class="dropdown-item" href="#">SubMenu2</a>

<a class="dropdown-item" href="#">SubMenu3</a>

</div>

</li>

</ul>

</div>

</div>

</nav>

<div class="container">

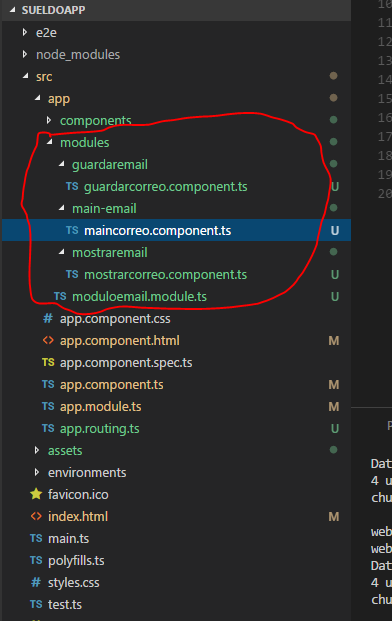
<router-outlet></router-outlet>

</div>

# Modulos

## Modulo simple

Creamos la siguiente estructura de archivos



**guardarcorreo.component.ts**

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

@Component({

selector: 'app-guardarcorreo',

template: `<h4> {{title}} </h4>

<input type="text" [(ngModel)]="emailContacto">

<button (click)="guardarEmail()">Guardar Email</button>`

})

export class GuardarCorreoComponent implements OnInit, DoCheck {

title = 'Guardar correo';

public emailContacto: string;

constructor() { }

ngOnInit() {

}

ngDoCheck() {

}

guardarEmail() {

// console.log(this.emailContacto);

// guardamos en localstorage

localStorage.setItem('emailContato', this.emailContacto);

// recupero del localstorage:

console.log(localStorage.getItem('emailContato'));

}

}

**mostrarcorreo.component.ts**

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

@Component({

selector: 'app-mostrarcorreo',

template: `

<div \*ngIf="emailContacto">

<h4> {{title}} </h4>

<strong> Email de contacto:</strong>

{{ emailContacto }}

<button (click) = "borrarEmail()">Eliminar email de contacto</button>

</div>

`

})

export class MostrarCorreoComponent implements OnInit, DoCheck {

title = 'Mostrar correo';

public emailContacto: string;

constructor() { }

ngOnInit() {

console.log('metodo onInit lanzado');

this.emailContacto = localStorage.getItem('emailContato');

}

ngDoCheck() {

console.log('metodo DoCheck Lanzado');

this.emailContacto = localStorage.getItem('emailContato');

}

borrarEmail() {

localStorage.removeItem('emailContato');

localStorage.clear();

this.emailContacto = null;

}

}

**maincorreo.component.ts**

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

@Component({

selector: 'app-maincorreo',

template: `

<h1>{{title}}</h1>

<app-mostrarcorreo></app-mostrarcorreo>

<app-guardarcorreo></app-guardarcorreo>

`

})

export class MainCorreoComponent implements OnInit {

title = 'Modulo de correo';

public emailContacto: string;

constructor() { }

ngOnInit() {

console.log('componente principal de modulo cargado');

}

}

**moduloemail.module.ts**

// tslint:disable-next-line:eofline

// importar modulos necesarios para crear modulos

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

import { CommonModule } from '@angular/common';

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

// importat componentes

import { GuardarCorreoComponent } from './guardaremail/guardarcorreo.component';

import { MostrarCorreoComponent } from './mostraremail/mostrarcorreo.component';

import { MainCorreoComponent } from './main-email/maincorreo.component';

// decorador ngModules para cargar componentes y configuracion de modulos

@NgModule({

imports: [CommonModule, FormsModule],

declarations: [

GuardarCorreoComponent,

MostrarCorreoComponent,

MainCorreoComponent

],

exports: [MainCorreoComponent]

})

export class ModuloEmailModule { }

**modificamos nuestro app.modules.ts**

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

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

// importamos el routing

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

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

import { HttpModule } from '@angular/http';

// importar nuestro modulos creados

import { ModuloEmailModule } from './modules/moduloemail.module';

// componentes para que esten dismonibles en toda la aplicacion

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

import { TiendaComponent } from './components/tienda/tienda.component';

import { ParqueComponent } from './components/parque/parque.component';

import { HomeComponent } from './components/home/home.component';

import { AnimalsComponent } from './components/animals/animals.component';

import { ContactComponent } from './components/contact/contact.component';

import { KeeperComponent } from './components/keeper/keeper.component';

import { NotFoundComponent } from './components/not-found/not-found.component';

@NgModule({

declarations: [

AppComponent,

TiendaComponent,

ParqueComponent,

HomeComponent,

AnimalsComponent,

ContactComponent,

KeeperComponent,

NotFoundComponent

],

imports: [

BrowserModule,

FormsModule,

HttpModule,

routing,

ModuloEmailModule

],

providers: [

appRoutingProviders

],

bootstrap: [AppComponent]

})

export class AppModule { }

Ahora ya podremos utilizar este modulo para el ejemplo lo llamaremos en nuestro componente de contacto.

contact.component.html

<p>

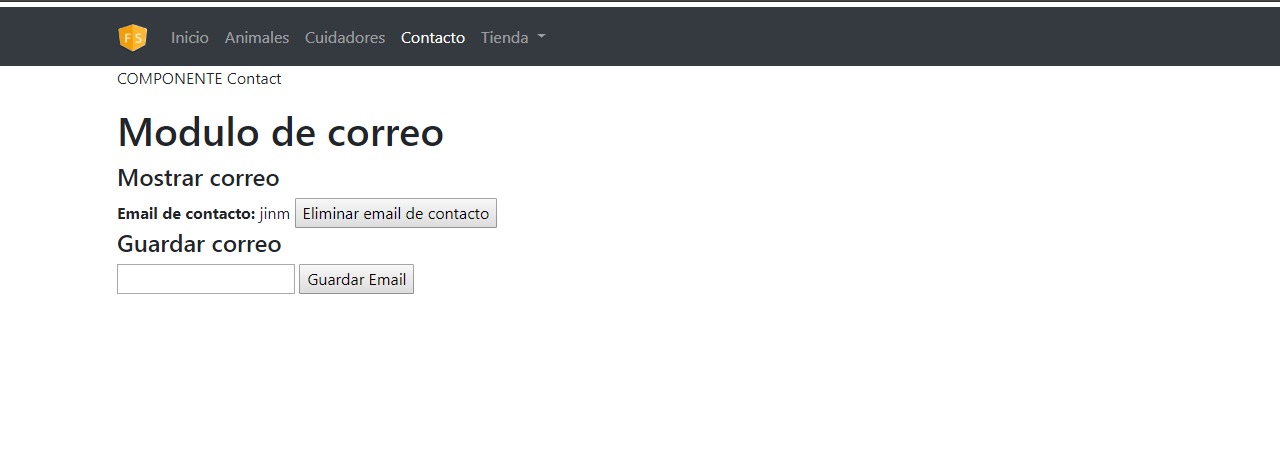
{{title}}

</p>

<!--input type="text" [(ngModel)]="emailContacto"-->

<!--button (click)="guardarEmail()">Guardar Email</button-->

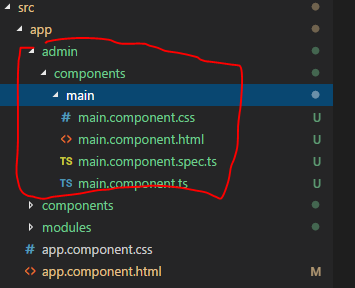
<app-maincorreo></app-maincorreo>



## Crear Modulo avanzado con rutas

Creamos componentes:

ng g component admin/components/main



Main.component.ts

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

@Component({

selector: 'app-main',

templateUrl: './main.component.html',

styleUrls: ['./main.component.css']

})

export class MainComponent implements OnInit {

title = 'Component MAIN';

constructor() { }

ngOnInit() {

}

}

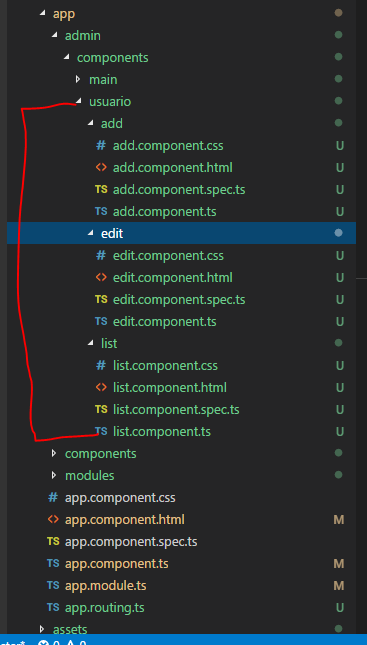
**Main.componet.html**

<p>

{{title}}

</p>

**Creamos los demás modulos para operaciones crud**



Comandos;

ng g component admin/components/usuario/add

ng g component admin/components/usuario/edit

ng g component admin/components/usuario/list

Add.component.ts

@Component({

selector: 'app-add',

templateUrl: './add.component.html',

styleUrls: ['./add.component.css']

})

export class AddComponent implements OnInit {

title = 'Añadir usuario';

constructor() { }

ngOnInit() {

}

}

Add.component.html

<p>

{{title}}

</p>

Verificamos que nuestr app.modules este actualizado

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

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

// importamos el routing

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

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

import { HttpModule } from '@angular/http';

// importar nuestro modulos creados

import { ModuloEmailModule } from './modules/moduloemail.module';

// componentes para que esten dismonibles en toda la aplicacion

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

import { TiendaComponent } from './components/tienda/tienda.component';

import { ParqueComponent } from './components/parque/parque.component';

import { HomeComponent } from './components/home/home.component';

import { AnimalsComponent } from './components/animals/animals.component';

import { ContactComponent } from './components/contact/contact.component';

import { KeeperComponent } from './components/keeper/keeper.component';

import { NotFoundComponent } from './components/not-found/not-found.component';

import { MainComponent } from './admin/components/main/main.component';

import { AddComponent } from './admin/components/usuario/add/add.component';

import { EditComponent } from './admin/components/usuario/edit/edit.component';

import { ListComponent } from './admin/components/usuario/list/list.component';

@NgModule({

declarations: [

AppComponent,

TiendaComponent,

ParqueComponent,

HomeComponent,

AnimalsComponent,

ContactComponent,

KeeperComponent,

NotFoundComponent,

MainComponent,

AddComponent,

EditComponent,

ListComponent

],

imports: [

BrowserModule,

FormsModule,

HttpModule,

routing,

ModuloEmailModule

],

providers: [

appRoutingProviders

],

bootstrap: [AppComponent]

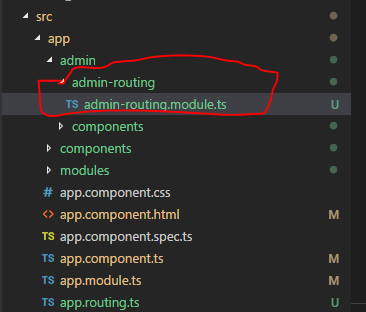
})

export class AppModule { }

## Modulo funcionalidad. Rutas hijas

Creamos un module para rutear:

ng g module admin/adminRouting



Modificamos el archivo

Admin-routing.module.ts

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

// import { CommonModule } from '@angular/common';

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

// importamos componentes

import { MainComponent } from '../components/main/main.component';

import { AddComponent } from '../components/usuario/add/add.component';

import { EditComponent } from '../components/usuario/edit/edit.component';

import { ListComponent } from '../components/usuario/list/list.component';

const adminRoutes: Routes = [

{

path: 'admin-panel',

component: MainComponent,

children: [

{ path: 'listado', component: ListComponent },

{ path: 'crear', component: AddComponent },

{ path: 'editar', component: EditComponent },

]

}

];

@NgModule({

imports: [

RouterModule.forChild(adminRoutes)

],

exports: [

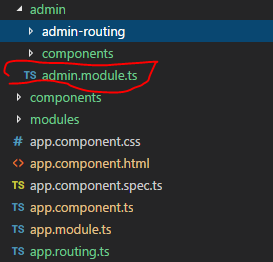
RouterModule

]

})

export class AdminRoutingModule { }

Creamos el fichero admin.module.ts



// modulos

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

import { CommonModule } from '@angular/common';

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

import { HttpModule } from '@angular/http';

import { AdminRoutingModule } from './admin-routing/admin-routing.module';

// importamos componentes

import { MainComponent } from './components/main/main.component';

import { AddComponent } from './components/usuario/add/add.component';

import { EditComponent } from './components/usuario/edit/edit.component';

import { ListComponent } from './components/usuario/list/list.component';

@NgModule({

declarations: [

MainComponent,

AddComponent,

EditComponent,

ListComponent

],

imports: [

CommonModule,

FormsModule,

HttpModule,

AdminRoutingModule

],

exports: [

],

providers: []

})

export class AdminModule { }

verificamos app.modules.ts

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

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

// importamos el routing

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

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

import { HttpModule } from '@angular/http';

// importar nuestro modulos creados

import { ModuloEmailModule } from './modules/moduloemail.module';

import { AdminModule } from './admin/admin.module';

// componentes para que esten dismonibles en toda la aplicacion

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

import { TiendaComponent } from './components/tienda/tienda.component';

import { ParqueComponent } from './components/parque/parque.component';

import { HomeComponent } from './components/home/home.component';

import { AnimalsComponent } from './components/animals/animals.component';

import { ContactComponent } from './components/contact/contact.component';

import { KeeperComponent } from './components/keeper/keeper.component';

import { NotFoundComponent } from './components/not-found/not-found.component';

/\*

import { MainComponent } from './admin/components/main/main.component';

import { AddComponent } from './admin/components/usuario/add/add.component';

import { EditComponent } from './admin/components/usuario/edit/edit.component';

import { ListComponent } from './admin/components/usuario/list/list.component';

\*/

@NgModule({

declarations: [

AppComponent,

TiendaComponent,

ParqueComponent,

HomeComponent,

AnimalsComponent,

ContactComponent,

KeeperComponent,

NotFoundComponent

/\*MainComponent,

AddComponent,

EditComponent,

ListComponent

\*/

],

imports: [

BrowserModule,

FormsModule,

HttpModule,

routing,

ModuloEmailModule,

AdminModule

],

providers: [

appRoutingProviders

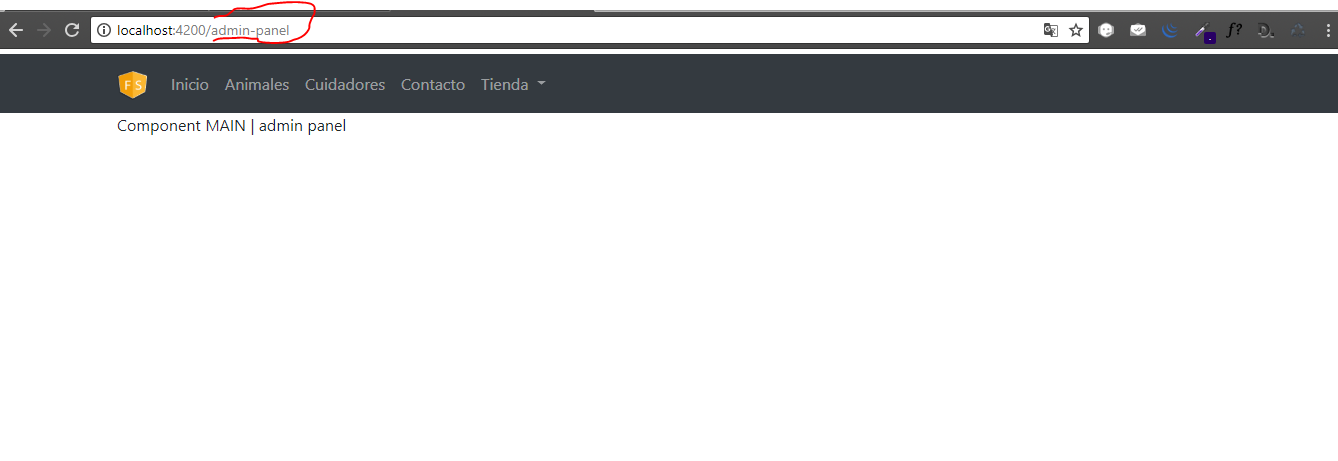
],

bootstrap: [AppComponent]

})

export class AppModule { }

Resultado



Modificamos admin-routing.module.ts

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

// import { CommonModule } from '@angular/common';

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

// importamos componentes

import { MainComponent } from '../components/main/main.component';

import { AddComponent } from '../components/usuario/add/add.component';

import { EditComponent } from '../components/usuario/edit/edit.component';

import { ListComponent } from '../components/usuario/list/list.component';

const adminRoutes: Routes = [

{

path: 'admin-panel', component: MainComponent,

children: [

{ path: '', redirectTo: 'listado', pathMatch: 'full' }, // para que redirecione cuando la ruta esta vacia

{ path: 'listado', component: ListComponent },

{ path: 'crear', component: AddComponent },

{ path: 'editar', component: EditComponent }

]

}

];

@NgModule({

imports: [

RouterModule.forChild(adminRoutes)

],

exports: [

RouterModule

]

})

export class AdminRoutingModule { }

modificar main.component.html

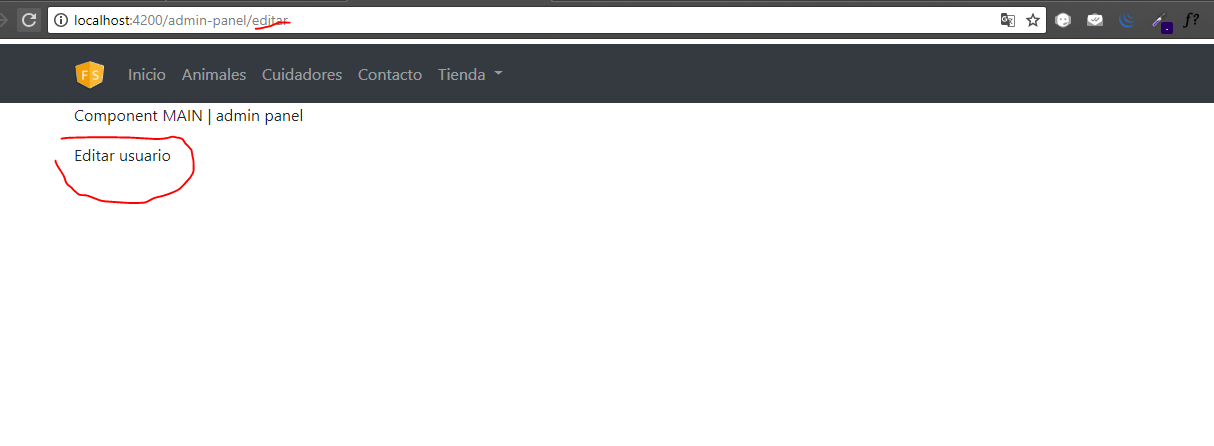
<p>

{{title}}

</p>

<router-outlet></router-outlet>

Resultado:



\*ya podemos navegar por las rutas hijas

Modificamos admin-routing.module.ts

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

// import { CommonModule } from '@angular/common';

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

// importamos componentes

import { MainComponent } from '../components/main/main.component';

import { AddComponent } from '../components/usuario/add/add.component';

import { EditComponent } from '../components/usuario/edit/edit.component';

import { ListComponent } from '../components/usuario/list/list.component';

const adminRoutes: Routes = [

{

path: 'admin-panel', component: MainComponent,

children: [

{ path: '', redirectTo: 'listado', pathMatch: 'full' }, // para que redirecione cuando la ruta esta vacia

{ path: 'listado', component: ListComponent },

{ path: 'crear', component: AddComponent },

{ path: 'editar', component: EditComponent }

]

},

{path: 'listado-del-panel', component: ListComponent }

];

@NgModule({

imports: [

RouterModule.forChild(adminRoutes)

],

exports: [

RouterModule

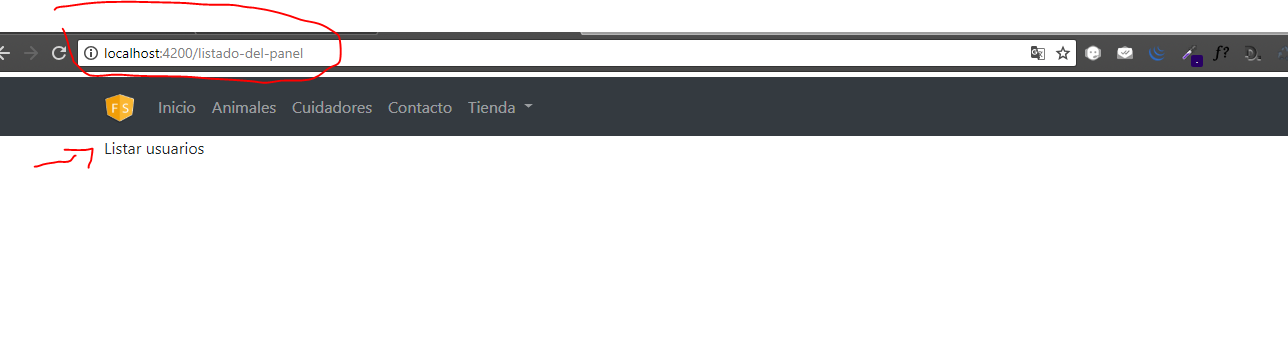
]

})

export class AdminRoutingModule { }

Resultado

<http://localhost:4200/listado-del-panel>



\*ya no sale como componente hijo se muestra como principal

# Modulo de panel de administracion

## Maquetación del modulo del panel de administración

Modificamos main.component.html

<p>

{{title}}

</p>

<div >

<div class="row wrapper">

<aside class="col-12 col-sm-3 p-0 bg-dark">

<nav class="navbar navbar-expand-sm navbar-dark bg-dark align-items-start flex-sm-column flex-row">

<span class="navbar-brand"> Panel Administracion</span>

<div class="collapse navbar-collapse sidebar">

<ul class="flex-column navbar-nav w-100 justify-content-between">

<li class="nav-item " [routerLinkActive]="['active']">

<a class="nav-link pl-0" [routerLink]="['crear']"> crear</a>

</li>

<li class="nav-item" [routerLinkActive]="['active']">

<a class="nav-link pl-0" [routerLink]="['listado']"> <span class="">listado</span></a>

</li>

</ul>

</div>

</nav>

</aside>

<main class="col bg-faded ">

<router-outlet></router-outlet>

</main>

</div>

</div>

Modificamos main.component.css

.wrapper {

min-height: 100vh;

}

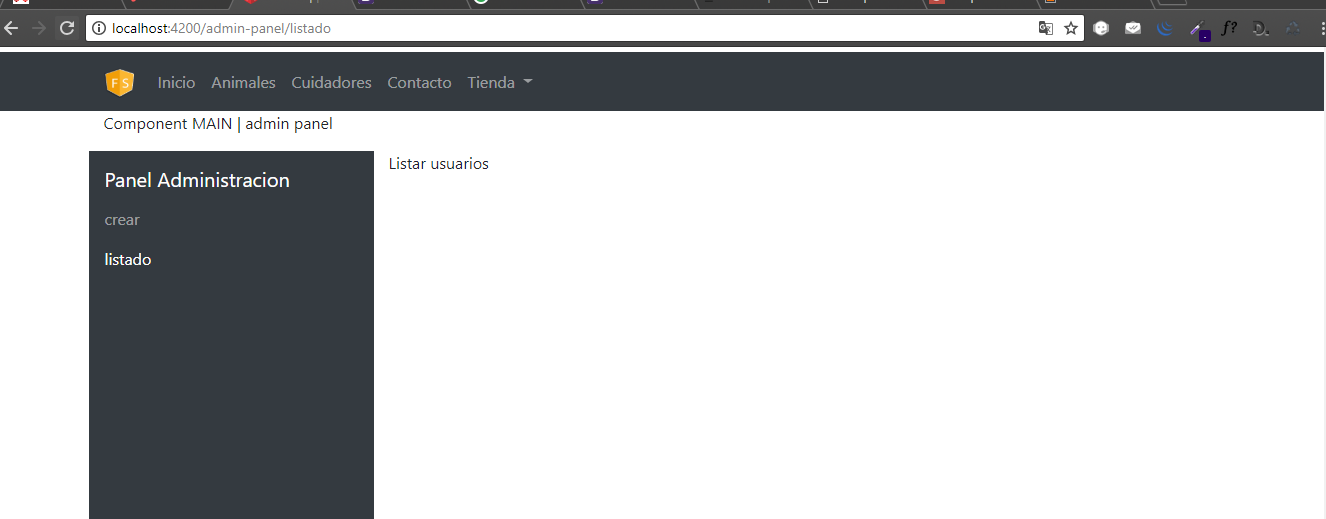
main {

max-height: 100%;

overflow-y: auto;

}

Resultado:



Maquetamos componente listado

List.component.html

<div class="col-lg-12">

<h3>

{{title}}

</h3>

<table class="table table-hover">

<thead class="thead-default">

<tr>

<th>Nombres</th>

<th>Apellidos</th>

<th>Email</th>

<th>Acciones</th>

</tr>

</thead>

<tbody>

<tr>

<td>jinmi lidonil</td>

<td>Vega Vega</td>

<td>jinmi16@gmail.com</td>

<td>

<button class="btn btn-sm btn-success">

Ver

</button>

<button class="btn btn-sm btn-primary">

Editar

</button>

<button class="btn btn-sm btn-danger">

Borrar</button>

</td>

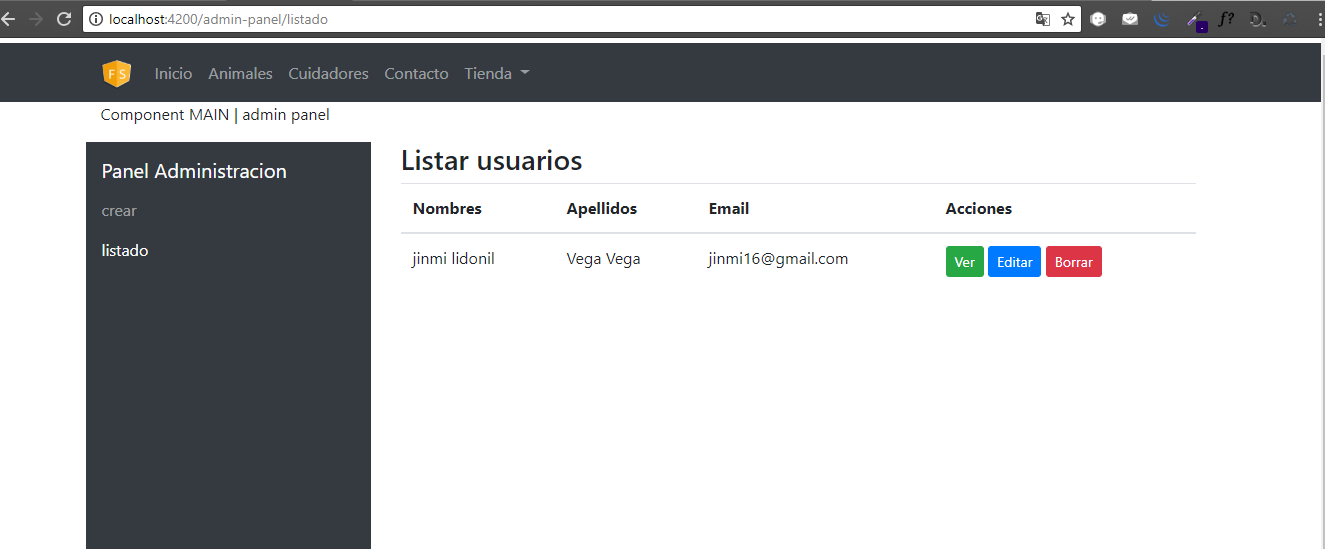
</tr>

</tbody>

</table>

</div>

Resultado:



También podemos usar ngForm para llenar la tabla:

List.component.html

<div class="col-lg-12">

<h3>

{{title}}

</h3>

<table class="table table-hover">

<thead class="thead-default">

<tr>

<th>Nombres</th>

<th>Apellidos</th>

<th>Email</th>

<th>Acciones</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let n of numbers; let i = index">

<td>jinmi lidonil {{ i + 1}}</td>

<td>Vega Vega</td>

<td>jinmi16@gmail.com</td>

<td>

<button class="btn btn-sm btn-success">

Ver

</button>

<button class="btn btn-sm btn-primary">

Editar

</button>

<button class="btn btn-sm btn-danger">

Borrar</button>

</td>

</tr>

</tbody>

</table>

</div>

list.component.ts

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

@Component({

selector: 'app-list',

templateUrl: './list.component.html',

styleUrls: ['./list.component.css']

})

export class ListComponent implements OnInit {

title = 'Listar usuarios';

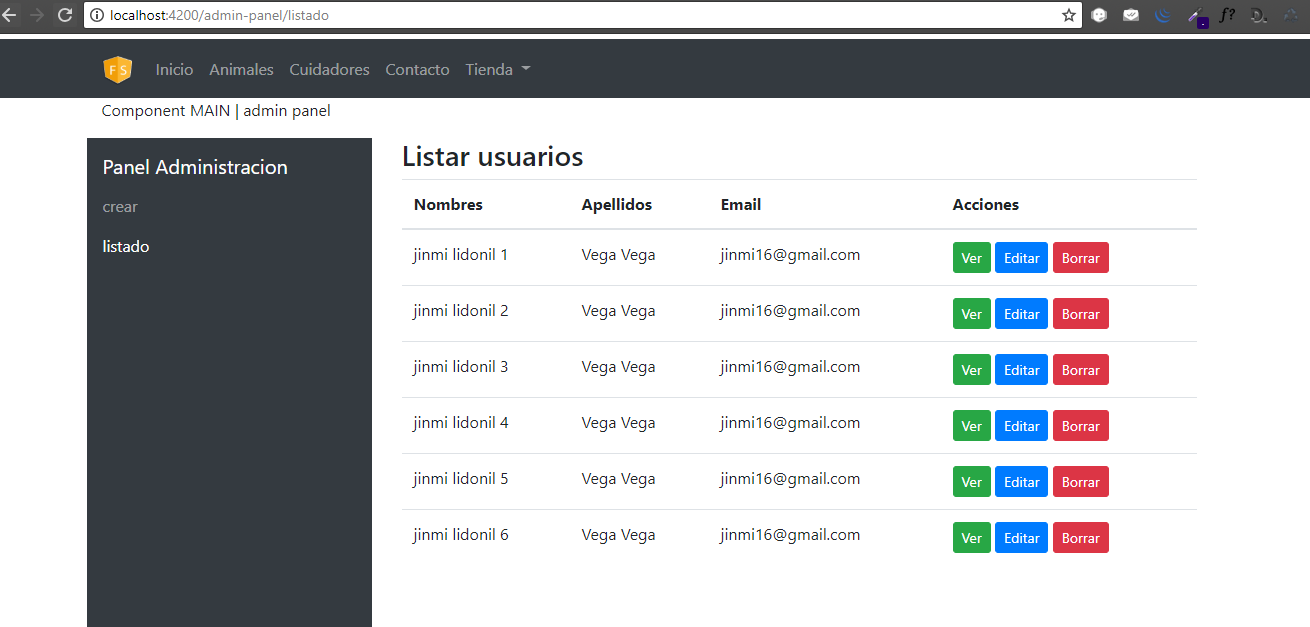
numbers = [0, 1, 2, 3, 4, 5];

constructor() { }

ngOnInit() {

}

}



Otra manera

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

@Component({

selector: 'app-list',

templateUrl: './list.component.html',

styleUrls: ['./list.component.css']

})

export class ListComponent implements OnInit {

title = 'Listar usuarios';

// numbers = [0, 1, 2, 3, 4, 5];

numbers = new Array(25);

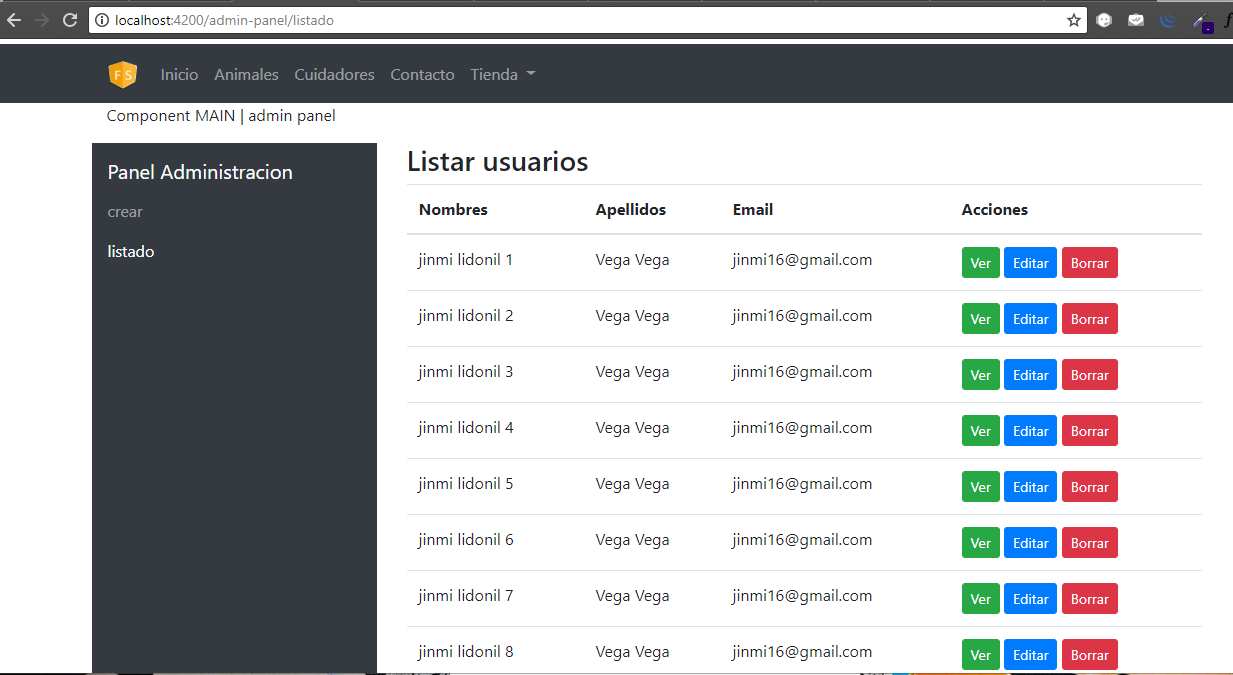
constructor() { }

ngOnInit() {

}

}

Resultado



## Formulario para añadir usuarios

Modificamos nuestro componente add.component.html

<div class="col-lg-12">

<h3>

{{title}}

</h3>

<form action="">

<p>

<label for="">Nombres</label>

<input type="text" class="form-control">

</p>

<p>

<label for="">Apellidos</label>

<input type="text" class="form-control">

</p>

<p>

<label for="">Usuario</label>

<input type="text" class="form-control">

</p>

<p>

<label for="">Contraseña</label>

<input type="text" class="form-control">

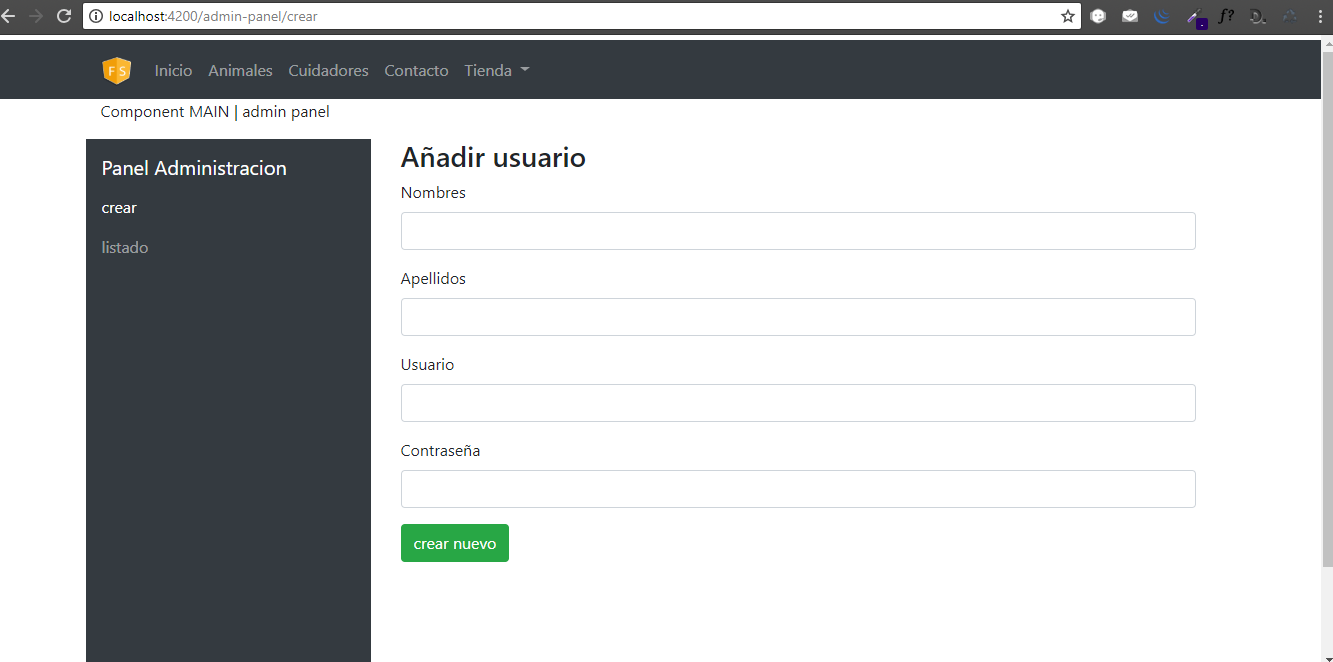
</p>

<input type="submit" value="crear nuevo" class="btn btn-success">

</form>

</div>

Resultado :



## Elementos del menú para login y registro

Modificamos app.component.html

<!--The content below is only a placeholder and can be replaced.-->

<nav class="navbar navbar-expand-lg navbar-dark bg-dark " style="margin-top: 5px">

<div class="container">

<a class="navbar-brand" href="#">

<img src="https://escuelafullstack.com/web/image/res.company/1/logo?unique=fa956a0" width="30" height="30" alt="">

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavDropdown">

<ul class="navbar-nav">

<li class="nav-item ">

<a class="nav-link" [routerLink]="['/home']" [routerLinkActive]="['active']">Inicio </a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/animales']" [routerLinkActive]="['active']">Animales</a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/cuidadores']" [routerLinkActive]="['active']">Cuidadores</a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/contacto']" [routerLinkActive]="['active']">Contacto</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true"

aria-expanded="false" [routerLinkActive]="['active']">

Tienda

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">

<a class="dropdown-item" href="#">SubMenu1</a>

<a class="dropdown-item" href="#">SubMenu2</a>

<a class="dropdown-item" href="#">SubMenu3</a>

</div>

</li>

</ul>

<ul class="nav navbar-nav ml-auto">

<li class="nav-item">

<a class="nav-link" [routerLink]="['/']"> registro</a>

</li>

<li class="nav-item">

<a class="nav-link " [routerLink]="['/']"> login</a>

</li>

</ul>

</div>

</div>

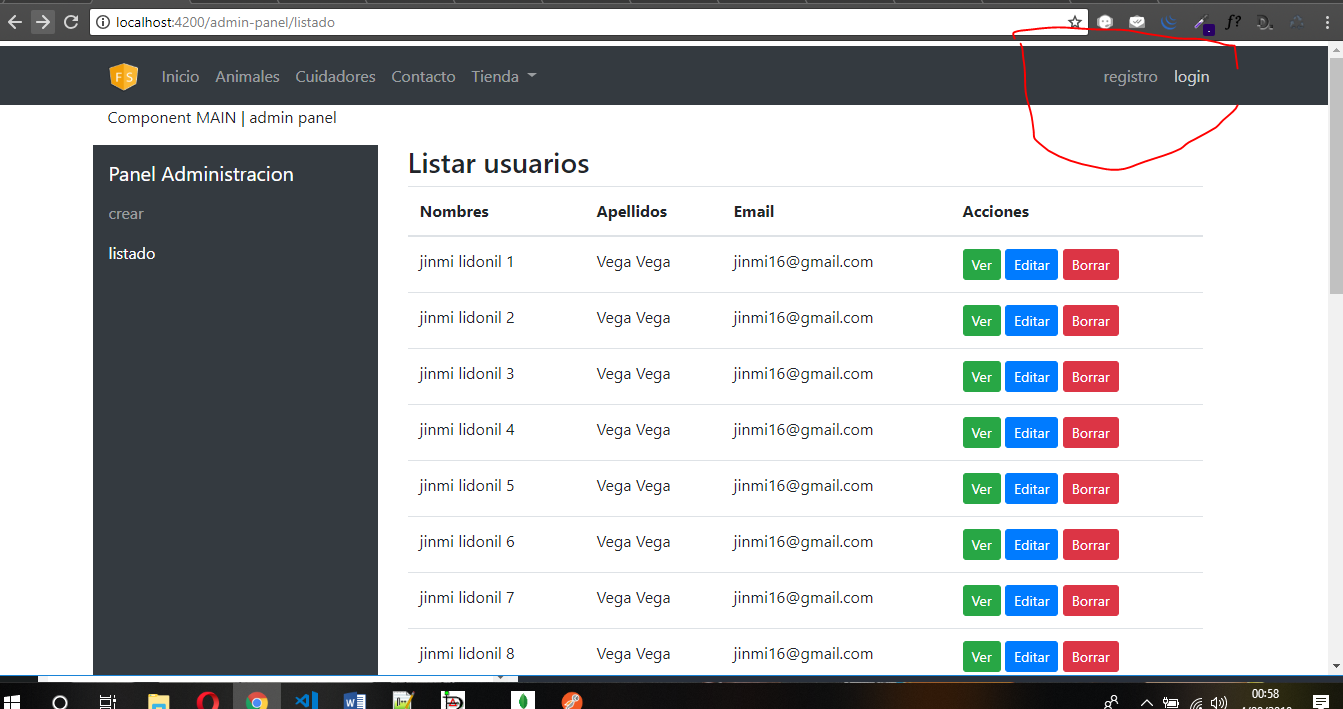
</nav>

<div class="container">

<router-outlet></router-outlet>

</div>

Resultado:

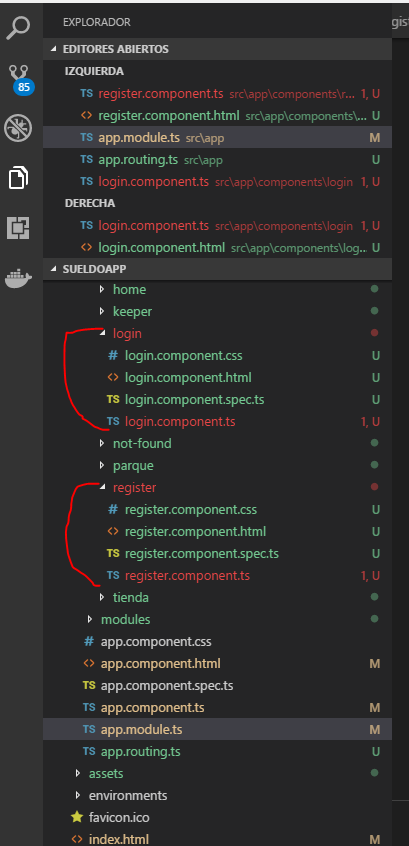


## Componentes y rutas para login y registro

Creamos los componentes

ng g component components/login

ng g component components/register



Modificamos:

Register.component.ts

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

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

@Component({

selector: 'app-register',

templateUrl: './register.component.html',

styleUrls: ['./register.component.css']

})

export class RegisterComponent implements OnInit {

public title: String;

constructor(

private \_route: ActivatedRoute,

private \_router: Router

) {

this.title = 'Registro';

}

ngOnInit() {

console.log('register.componet cargado !!');

}

}

Register.component.html

<div class="col-lg-12">

<h1>

{{title}}

</h1>

<form action="">

<p>

<label for="">Nombres</label>

<input type="text" class="form-control">

</p>

<p>

<label for="">Apellidos</label>

<input type="text" class="form-control">

</p>

<p>

<label for="">Usuario Email</label>

<input type="text" class="form-control">

</p>

<p>

<label for="">Contraseña</label>

<input type="text" class="form-control">

</p>

<p>

<label for="">Nickname</label>

<input type="text" class="form-control">

</p>

<p>

<label for="">Telefono</label>

<input type="text" class="form-control">

</p>

<p>

<label for="">Edad</label>

<input type="text" class="form-control">

</p>

<input type="submit" value="crear nuevo" class="btn btn-success">

</form>

</div>

Login.component.ts

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

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

@Component({

selector: 'app-login',

templateUrl: './login.component.html',

styleUrls: ['./login.component.css']

})

export class LoginComponent implements OnInit {

public title: String;

constructor(

private \_route: ActivatedRoute,

private \_router: Router

) {

this.title = 'Login';

}

ngOnInit() {

console.log('login.componet cargado !!');

}

}

Login.componet.html

<div id="login">

<div class="container">

<div id="login-row" class="row justify-content-center align-items-center">

<div id="login-column" class="col-md-6">

<div id="login-box" class="col-md-12">

<form id="login-form" class="form" action="" method="post">

<h3 class="text-center text-info">Identificate</h3>

<div class="form-group">

<label for="username" class="text-info">usuario:</label>

<br>

<input type="text" name="username" id="username" class="form-control" placeholder="ejemplo@gmail.com">

</div>

<div class="form-group">

<label for="password" class="text-info">Password:</label>

<br>

<input type="text" name="password" id="password" class="form-control">

</div>

<div class="form-group" style="text-align: center;">

<input type="submit" name="submit" class="btn btn-info btn-md" value="Ingresar">

</div>

</form>

</div>

</div>

</div>

</div>

</div>

Login.component.css

body {

margin: 0;

padding: 0;

background-color: #17a2b8;

height: 100vh;

}

#login .container #login-row #login-column #login-box {

margin-top: 20px;

max-width: 600px;

height: 320px;

border: 1px solid #9C9C9C;

background-color: #EAEAEA;

}

#login .container #login-row #login-column #login-box #login-form {

padding: 20px;

}

#login .container #login-row #login-column #login-box #login-form #register-link {

margin-top: -85px;

}

Ahora modificamos app.component.html

<!--The content below is only a placeholder and can be replaced.-->

<nav class="navbar navbar-expand-lg navbar-dark bg-dark " style="margin-top: 5px">

<div class="container">

<a class="navbar-brand" href="#">

<img src="https://escuelafullstack.com/web/image/res.company/1/logo?unique=fa956a0" width="30" height="30" alt="">

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavDropdown">

<ul class="navbar-nav">

<li class="nav-item ">

<a class="nav-link" [routerLink]="['/home']" [routerLinkActive]="['active']">Inicio </a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/animales']" [routerLinkActive]="['active']">Animales</a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/cuidadores']" [routerLinkActive]="['active']">Cuidadores</a>

</li>

<li class="nav-item">

<a class="nav-link" [routerLink]="['/contacto']" [routerLinkActive]="['active']">Contacto</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true"

aria-expanded="false" [routerLinkActive]="['active']">

Tienda

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">

<a class="dropdown-item" href="#">SubMenu1</a>

<a class="dropdown-item" href="#">SubMenu2</a>

<a class="dropdown-item" href="#">SubMenu3</a>

</div>

</li>

</ul>

<ul class="nav navbar-nav ml-auto">

<li class="nav-item">

<a class="nav-link" [routerLink]="['registro']"> registro</a>

</li>

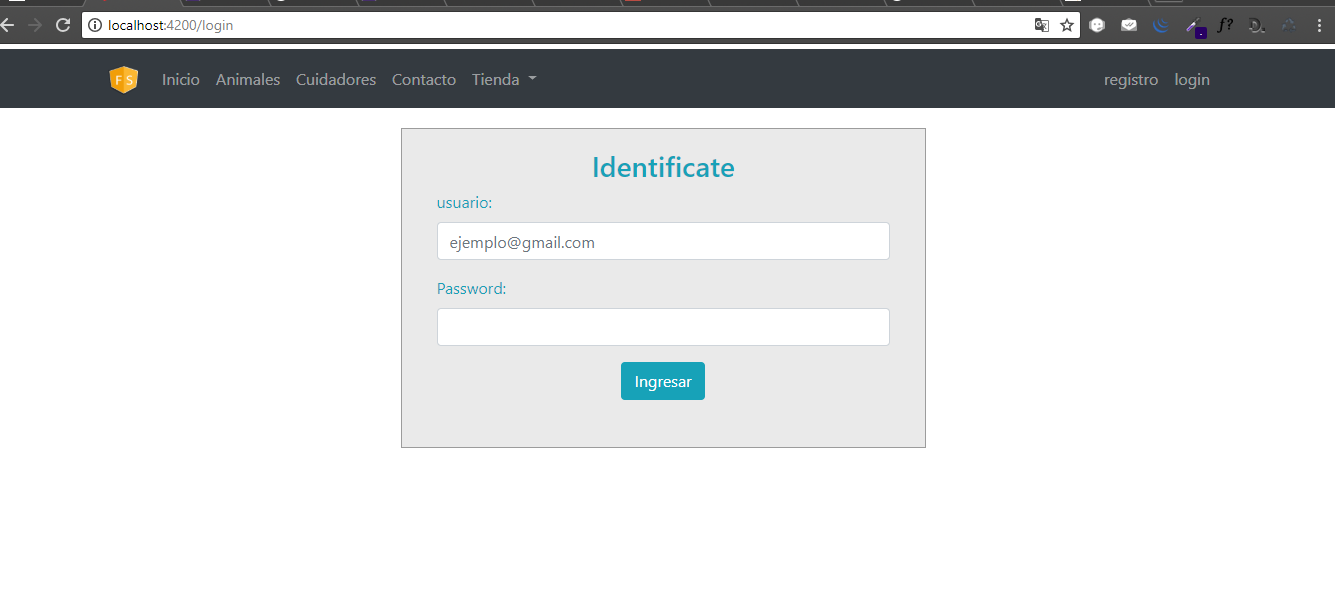
<li class="nav-item">

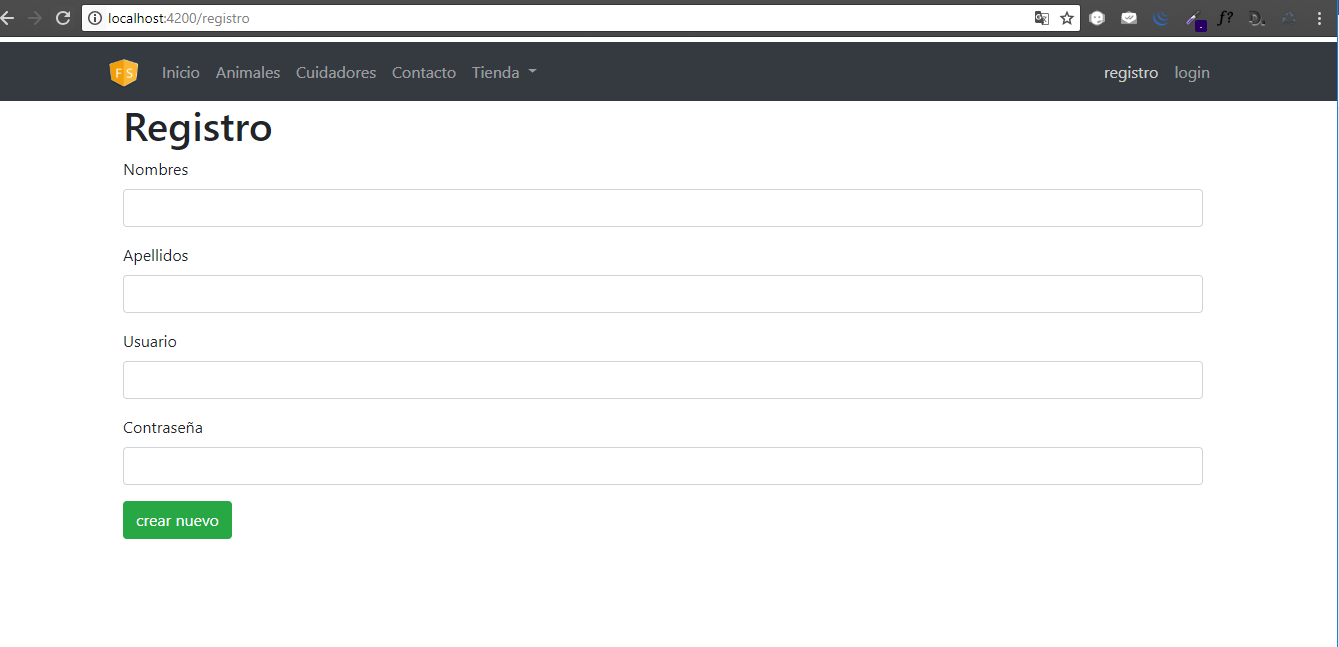
<a class="nav-link " [routerLink]="['login']"> login</a>

</li>

</ul>

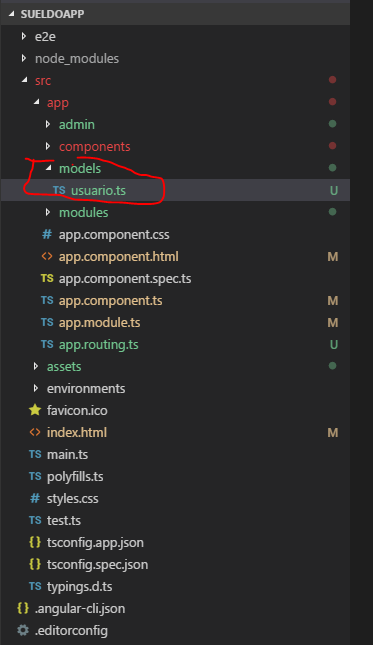
Resultado ahora ya podemos acceder dede el menú login y registro





# Crear Modelos

## Modelo de usuario



Usuario.ts

export class Usuario {

constructor(

public \_id: string,

public nikname: string,

public password: string,

public nombre: string,

public apellido: string,

public email: string,

public telefono: string,

public edad: string

) { }

}

## Recibir datos de formulario de registro

Modificamos register.component.ts

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

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

// importamos modelo de usuario

import { Usuario } from '../../models/usuario';

@Component({

selector: 'app-register',

templateUrl: './register.component.html',

styleUrls: ['./register.component.css']

})

export class RegisterComponent implements OnInit {

public title: String;

public usuario: Usuario;

constructor(

private \_route: ActivatedRoute,

private \_router: Router

) {

this.title = 'Registro';

this.usuario = new Usuario('', '', '', '', '', '', '', '');

}

ngOnInit() {

console.log('register.componet cargado !!');

}

}

Modificamos register.component.html

<div class="col-lg-12">

<h1>

{{title}}

</h1>

<form #registerForm="ngForm" (ngSubmit)="onSubmit()">

<p>

<label for="">Nombres</label>

<input type="text" class="form-control" required name="nombre" #nombre="ngModel" [(ngModel)]="usuario.nombre">

<span \*ngIf="!nombre.valid && nombre.touched">

El nombre es obligatorio

</span>

</p>

<p>

<label for="">Apellidos</label>

<input type="text" class="form-control" required name="apellido" #apellido="ngModel" [(ngModel)]="usuario.apellido">

<span \*ngIf="!apellido.valid && apellido.touched">

El apellido es obligatorio

</span>

</p>

<p>

<label for="">Usuario Email</label>

<input type="text" class="form-control" required name="email" #email="ngModel" [(ngModel)]="usuario.email">

<span \*ngIf="!email.valid && email.touched">

Usuario email es obligatorio

</span>

</p>

<p>

<label for="">Contraseña</label>

<input type="text" class="form-control" required name="password" #password="ngModel" [(ngModel)]="usuario.password">

<span \*ngIf="!password.valid && password.touched">

Contraseñaes obligatoria

</span>

</p>

<p>

<label for="">Nickname</label>

<input type="text" class="form-control" name="nikname" #nikname="ngModel" [(ngModel)]="usuario.nikname">

</p>

<p>

<label for="">Telefono</label>

<input type="text" class="form-control" name="telefono" #telefono="ngModel" [(ngModel)]="usuario.telefono">

</p>

<p>

<label for="">Edad</label>

<input type="text" class="form-control" name="edad" #edad="ngModel" [(ngModel)]="usuario.edad">

</p>

<input type="submit" value="crear nuevo" class="btn btn-primary" [disabled]="!registerForm.form.valid">

</form>

</div>

Modificamos register.component.ts

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

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

// importamos modelo de usuario

import { Usuario } from '../../models/usuario';

@Component({

selector: 'app-register',

templateUrl: './register.component.html',

styleUrls: ['./register.component.css']

})

export class RegisterComponent implements OnInit {

public title: String;

public usuario: Usuario;

constructor(

private \_route: ActivatedRoute,

private \_router: Router

) {

this.title = 'Registro';

this.usuario = new Usuario('', '', '', '', '', '', '', '');

}

ngOnInit() {

console.log('register.componet cargado !!');

}

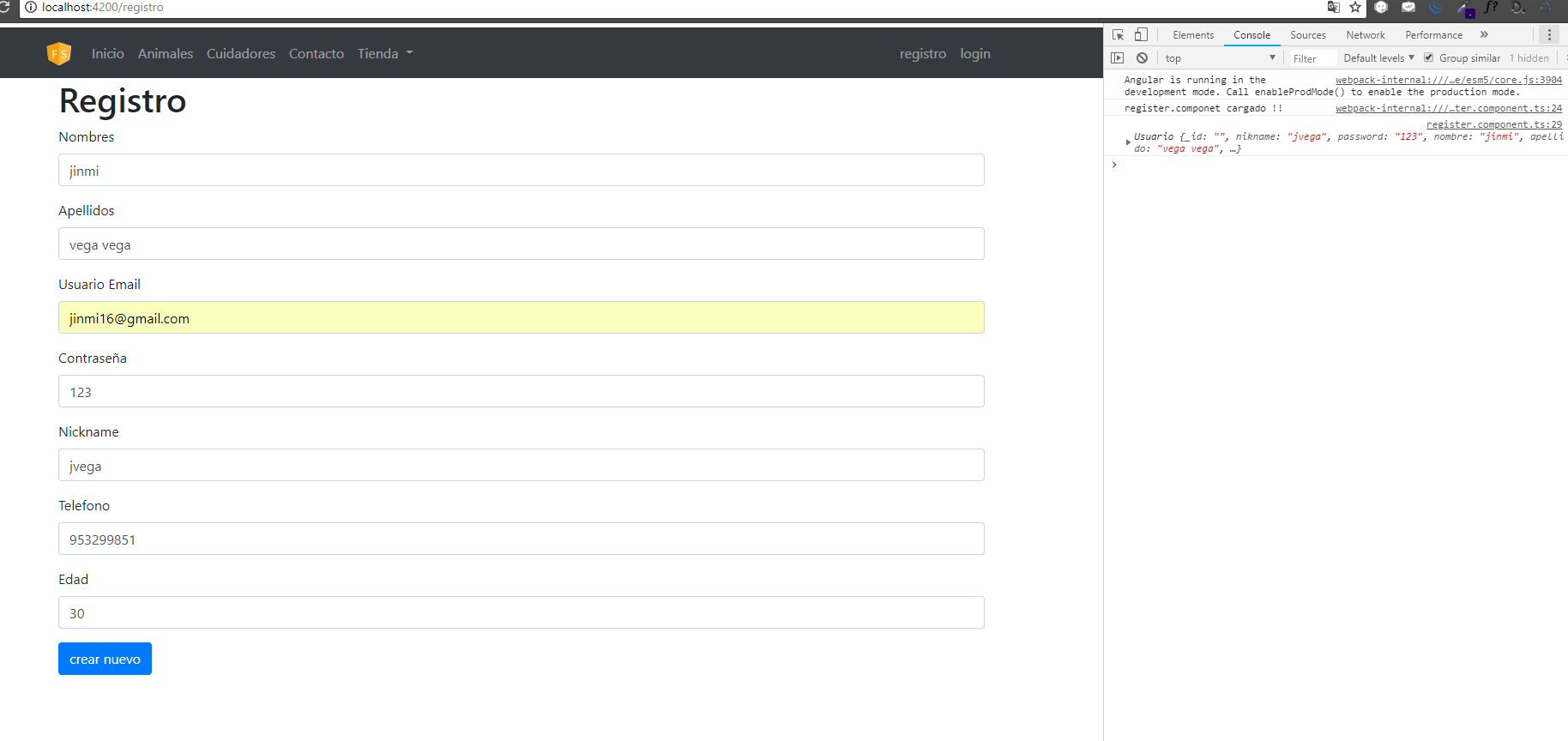
onSubmit() {

console.log(this.usuario);

}

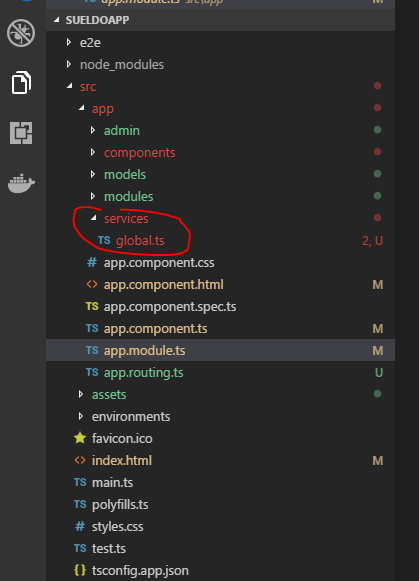
}

Resultado:



## Servicios para usuario

Creamos un fichero para los servicios



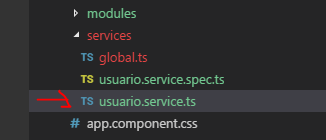
Global.ts

export const GLOBAL = {

url: 'http://localhost:3789/api/'

}

Creamos el fichero sercices/usuario.services.ts



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

import { Http, Response, Headers } from '@angular/http';

import 'rxjs/add/operator/map';

import { Observable } from 'rxjs/observable';

import { GLOBAL } from './global';

@Injectable()

export class UsuarioService {

public url: String = '';

constructor(

private \_http: Http

) {

this.url = GLOBAL.url;

}

register() {

return 'Texto desde el servicio';

}

}

Ahora llamamos el servicio en register.component.ts

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

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

// importamos modelo de usuario

import { Usuario } from '../../models/usuario';

// importamos servicios

import { GLOBAL } from '../../services/global';

import { UsuarioService } from '../../services/usuario.service';

@Component({

selector: 'app-register',

templateUrl: './register.component.html',

styleUrls: ['./register.component.css'],

providers: [UsuarioService]

})

export class RegisterComponent implements OnInit {

public title: String;

public usuario: Usuario;

constructor(

private \_route: ActivatedRoute,

private \_router: Router,

private \_userService: UsuarioService

) {

this.title = 'Registro';

this.usuario = new Usuario('', '', '', '', '', '', '', '');

}

ngOnInit() {

console.log('register.componet cargado !!');

console.log(this.\_userService.register());

}

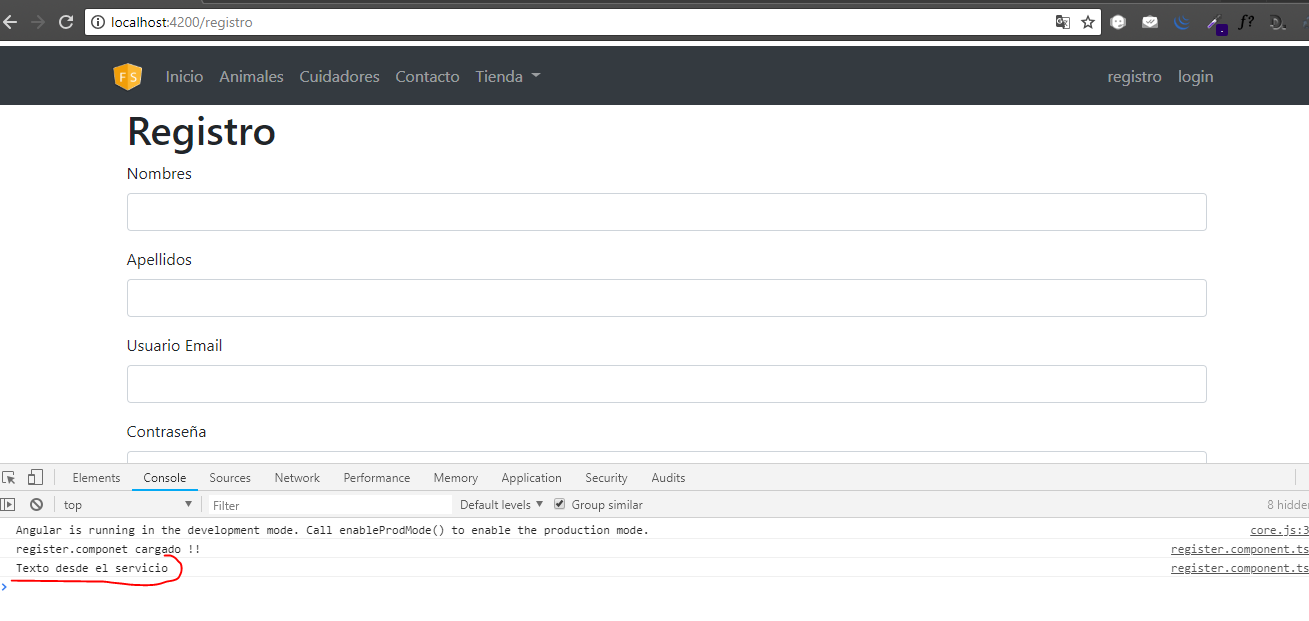
onSubmit() {

console.log(this.usuario);

}

}

Resultado:



También podemos cargar el servicio de manera global

Register.component.ts

// importamos servicios

import { GLOBAL } from '../../services/global';

import { UsuarioService } from '../../services/usuario.service';

@Component({

selector: 'app-register',

templateUrl: './register.component.html',

styleUrls: ['./register.component.css']

// providers: [UsuarioService]

})

export class RegisterComponent implements OnInit {

public title: String;

public usuario: Usuario;

constructor(

private \_route: ActivatedRoute,

private \_router: Router,

private \_userService: UsuarioService

) {

this.title = 'Registro';

this.usuario = new Usuario('', '', '', '', '', '', '', '');

}

ngOnInit() {

console.log('register.componet cargado !!');

console.log(this.\_userService.register());

}

onSubmit() {

console.log(this.usuario);

}

}

App.module.ts

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

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

// importamos el routing

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

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

import { HttpModule } from '@angular/http';

// importar nuestro modulos creados

import { ModuloEmailModule } from './modules/moduloemail.module';

import { AdminModule } from './admin/admin.module';

// importamos servicios

import { UsuarioService } from './services/usuario.service';

// componentes para que esten dismonibles en toda la aplicacion

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

import { TiendaComponent } from './components/tienda/tienda.component';

import { ParqueComponent } from './components/parque/parque.component';

import { HomeComponent } from './components/home/home.component';

import { AnimalsComponent } from './components/animals/animals.component';

import { ContactComponent } from './components/contact/contact.component';

import { KeeperComponent } from './components/keeper/keeper.component';

import { NotFoundComponent } from './components/not-found/not-found.component';

import { RegisterComponent } from './components/register/register.component';

import { LoginComponent } from './components/login/login.component';

/\*

import { MainComponent } from './admin/components/main/main.component';

import { AddComponent } from './admin/components/usuario/add/add.component';

import { EditComponent } from './admin/components/usuario/edit/edit.component';

import { ListComponent } from './admin/components/usuario/list/list.component';

\*/

@NgModule({

declarations: [

AppComponent,

TiendaComponent,

ParqueComponent,

HomeComponent,

AnimalsComponent,

ContactComponent,

KeeperComponent,

NotFoundComponent,

RegisterComponent,

LoginComponent

/\*MainComponent,

AddComponent,

EditComponent,

ListComponent

\*/

],

imports: [

BrowserModule,

FormsModule,

HttpModule,

routing,

ModuloEmailModule,

AdminModule

],

providers: [

appRoutingProviders,

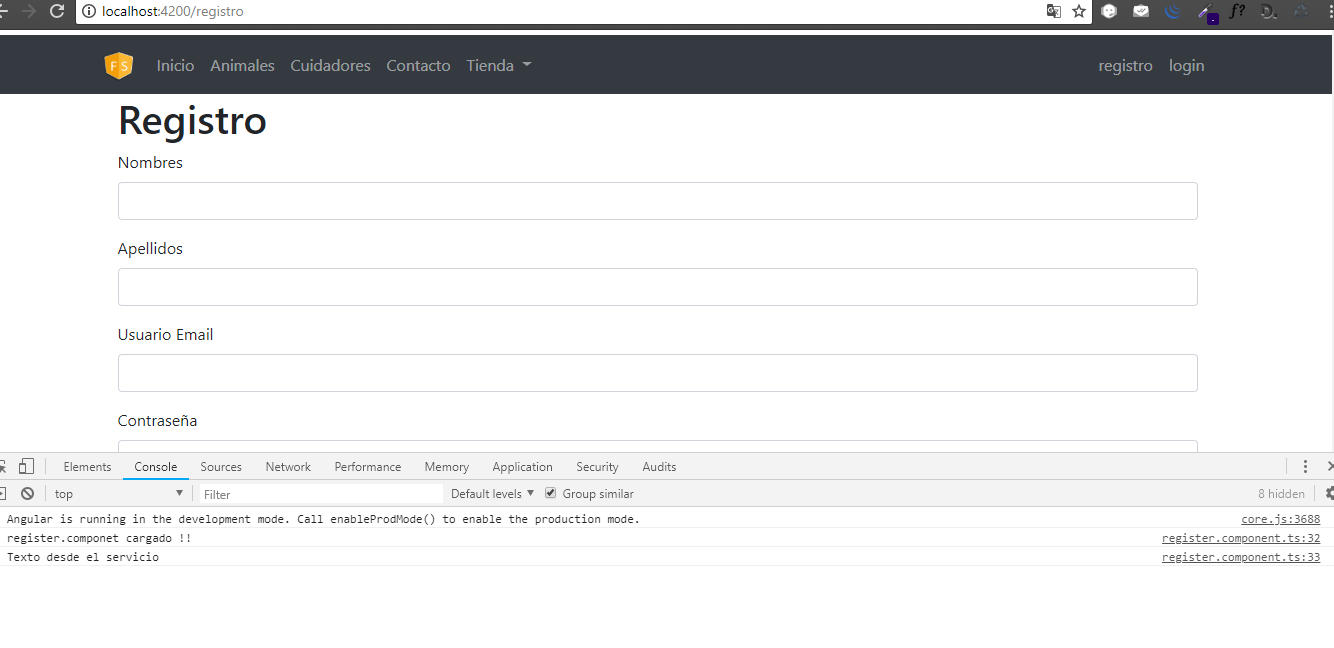
UsuarioService

],

bootstrap: [AppComponent]

})

export class AppModule { }



## Registrar usuario

Usuario.service.ts

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

import { Http, Response, Headers } from '@angular/http';

import 'rxjs/add/operator/map';

import { Observable } from 'rxjs/observable';

import { GLOBAL } from './global';

@Injectable()

export class UsuarioService {

public url: String = '';

constructor(

private \_http: Http

) {

this.url = GLOBAL.url;

}

register(user\_to\_register) {

const params = JSON.stringify(user\_to\_register);

const headers = new Headers({ 'Content-Type': 'application/json' });

return this.\_http.post(this.url + 'register', params, { headers: headers }).map(res => res.json());

}

}

Register.component.ts

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

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

// importamos modelo de usuario

import { Usuario } from '../../models/usuario';

// importamos servicios

import { GLOBAL } from '../../services/global';

import { UsuarioService } from '../../services/usuario.service';

@Component({

selector: 'app-register',

templateUrl: './register.component.html',

styleUrls: ['./register.component.css']

// providers: [UsuarioService]

})

export class RegisterComponent implements OnInit {

public title: String;

public status: String;

public usuario: Usuario;

constructor(

private \_route: ActivatedRoute,

private \_router: Router,

private \_userService: UsuarioService

) {

this.title = 'Registro';

this.usuario = new Usuario('', '', '', '', '', '', '', '');

}

ngOnInit() {

console.log('register.componet cargado !!');

// console.log(this.\_userService.register());

}

onSubmit() {

console.log(this.usuario);

this.\_userService.register(this.usuario).subscribe(

response => {

console.log(JSON.stringify(response));

if (response.user.\_id) {

this.status = 'success';

} else {

this.status = 'error';

}

this.usuario = new Usuario('', '', '', '', '', '', '', '');

},

error => {

console.log(<any>error);

}

);

}

}

Register.componet.html

<div class="col-lg-12">

<h1>

{{title}}

</h1>

<div class="alert alert-success" \*ngIf = "status == 'success'">

Registro completado exitosamente

</div>

<div class="alert alert-danger" \*ngIf = "status == 'error'">

No te has registrado correctamente

</div>

<form #registerForm="ngForm" (ngSubmit)="onSubmit()">

<p>

<label for="">Nombres</label>

<input type="text" class="form-control" required name="nombre" #nombre="ngModel" [(ngModel)]="usuario.nombre">

<span \*ngIf="!nombre.valid && nombre.touched">

El nombre es obligatorio

</span>

</p>

<p>

<label for="">Apellidos</label>

<input type="text" class="form-control" required name="apellido" #apellido="ngModel" [(ngModel)]="usuario.apellido">

<span \*ngIf="!apellido.valid && apellido.touched">

El apellido es obligatorio

</span>

</p>

<p>

<label for="">Usuario Email</label>

<input type="text" class="form-control" required name="email" #email="ngModel" [(ngModel)]="usuario.email">

<span \*ngIf="!email.valid && email.touched">

Usuario email es obligatorio

</span>

</p>

<p>

<label for="">Contraseña</label>

<input type="text" class="form-control" required name="password" #password="ngModel" [(ngModel)]="usuario.password">

<span \*ngIf="!password.valid && password.touched">

Contraseñaes obligatoria

</span>

</p>

<p>

<label for="">Nickname</label>

<input type="text" class="form-control" name="nikname" #nikname="ngModel" [(ngModel)]="usuario.nikname">

</p>

<p>

<label for="">Telefono</label>

<input type="text" class="form-control" name="telefono" #telefono="ngModel" [(ngModel)]="usuario.telefono">

</p>

<p>

<label for="">Edad</label>

<input type="text" class="form-control" name="edad" #edad="ngModel" [(ngModel)]="usuario.edad">

</p>

<input type="submit" value="crear nuevo" class="btn btn-primary" [disabled]="!registerForm.form.valid">

</form>

</div>