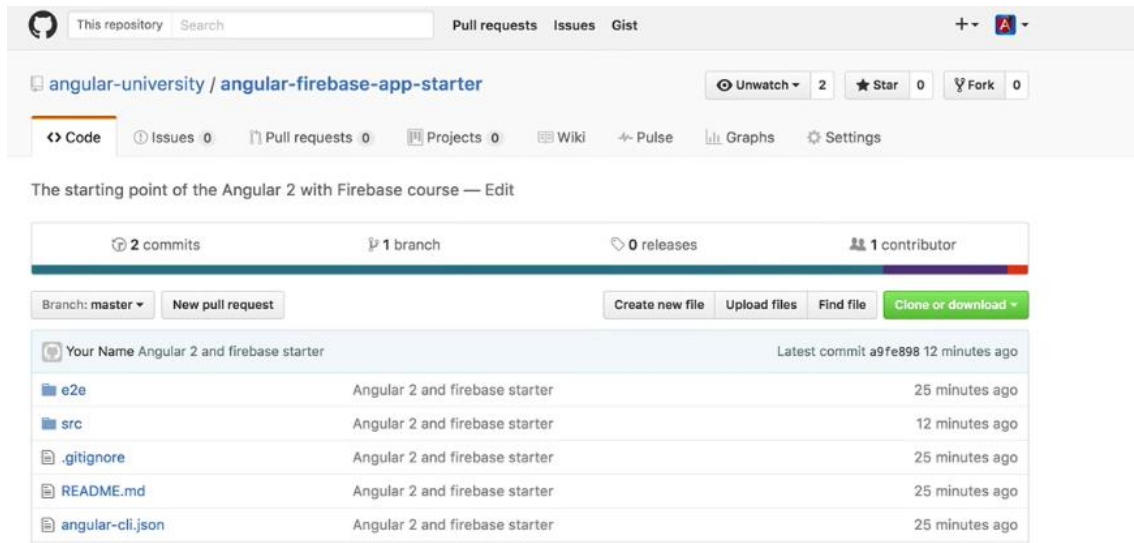
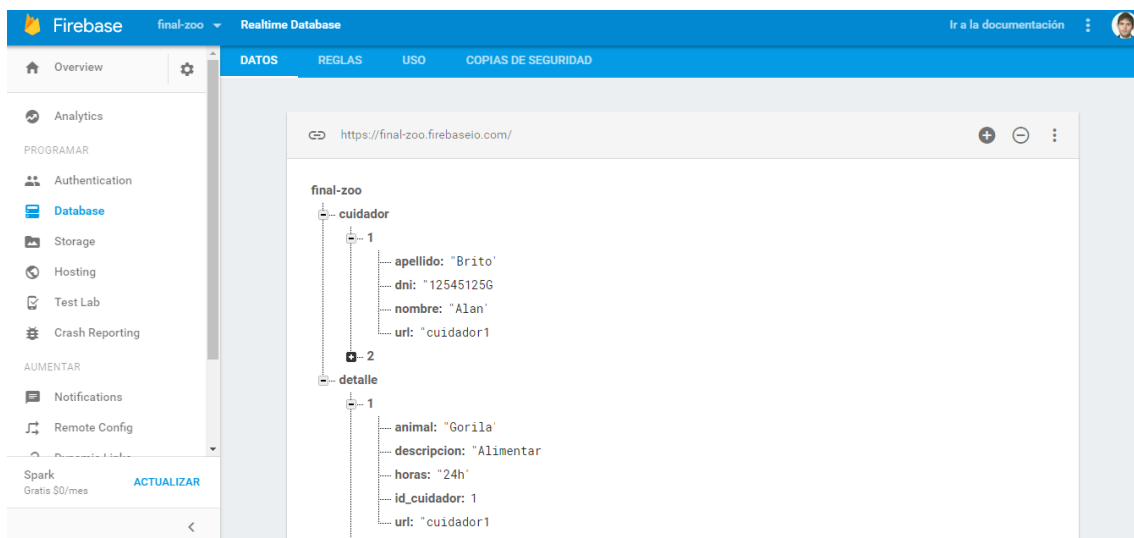


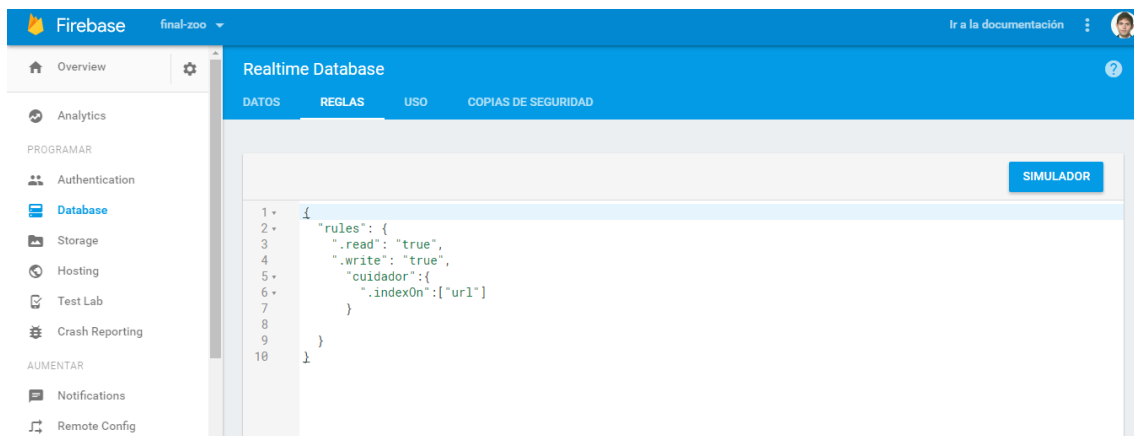
Descargamos los archivos para la aplicación que están en el github de angular



Vamos a Firebase a crear la base de datos



Cambiamos las reglas



Conectamos con nuestro firebase

```
import {AuthMethods, AuthProviders} from "angularfire2";

export const firebaseConfig = {
  apiKey: "AIzaSyAX1IBni0tUwVl1DWghbP_JpxroK_uwagI",
  authDomain: "final-zoo.firebaseio.com",
  databaseURL: "https://final-zoo.firebaseio.com",
  storageBucket: "final-zoo.appspot.com",
  messagingSenderId: "896858142315"
};

export const authConfig = {
  provider: AuthProviders.Password,
  method: AuthMethods.Password
};
```

Crearemos los métodos para hacer consultas la base de datos para listar objetos

```
import { Component } from '@angular/core';
import {AngularFire, FirebaseListObservable, FirebaseObjectObservable} from "angularfire2";
import {Observable} from 'rxjs/Observable';
import 'rxjs/add/operator/map';
import 'rxjs/add/operator/do';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'app works!';

  cuidador$: FirebaseListObservable<any>;
  detalle$: FirebaseObjectObservable<any>;
  firstCuidador: any;
  constructor(private af: AngularFire) {
    this.cuidador$ = af.database.list('cuidador');

    this.detalle$ = af.database.object('detalle/1');

    this.cuidador$.map(cuidador => cuidador[1])
      .subscribe(
        cuidador => this.firstCuidador = cuidador
      );
  }
}
```

Después se crea los métodos para agregar, eliminar y modificar.

```
listPush() {  
  this.cuidador$.push({ description: 'TEST NEW CUIDADOR' })  
  .then(  
    () => console.log('List Push Done'),  
    console.error  
  );  
}  
  
listRemove() {  
  this.cuidador$.remove(this.firstCuidador);  
}  
  
listUpdate() {  
  this.cuidador$.update(this.firstCuidador, { description: 'Modificado' })  
}  
  
objUpdate() {  
  this.detalle$.update({ descripcion: 'nueva descripcion' })  
}  
  
objSet() {  
  this.detalle$.update({ descripcion: 'nueva descripcion' })  
}
```

Una vez hecho, hay que crear un nuevo componente, el **home**.

ng generate component home

En el fichero home.component.html se creará el html con los estilos

```
<h1>Listado Cuidadores</h1>  
  
<h3>Busqueda por nombre del cuidador: {{cuidador?.length}}</h3>  
<input class="search-bar" placeholder="Buscar" (keyup)="search(input.value)" #input>  
  
<cuidador-list [cuidador]="filtro"></cuidador-list>
```

Hay que crear un nuevo servicio con el comando `ng g service`

Después hay que crear una clase para detalle y otra para el cuidador

```
import {Cuidador} from "../Cuidador";
import {Observable} from "rxjs/Rx";

export class Detalle {

    constructor(
        public $key:string,
        public descripcion:string,
        public animal:string,
        public horas: string,
        public id_cuidador:string,
        public url:string) {

    }

    static fromJson({$key, descripcion, animal, horas, id_cuidador, url}) {
        return new Detalle($key, descripcion, animal, horas, id_cuidador, url);
    }

    static fromJsonArray(json : any[]) : Detalle[] {
        return json.map(Detalle.fromJson);
    }

}

import {Observable} from "rxjs/Rx";
import 'rxjs/add/operator/map';
export class Cuidador {

    constructor(
        public $key:string,
        public apellido: string,
        public dni: string,
        public nombre: string,
        public url:string) {

    }

    static fromJson({$key, apellido, dni, nombre, url}) {
        return new Cuidador($key, apellido, dni, nombre, url);
    }

    static fromJsonArray(json : any[]) : Cuidador[] {
        return json.map(Cuidador.fromJson);
    }

}
```

En detalles.service.ts y cuidador.service.ts creamos la función que obtendrá los datos de la base de datos

```
import {Injectable, Inject} from '@angular/core';
import {AngularFire, FirebaseRef} from "angularfire2";
import {Observable, Subject} from "rxjs";
import {Detalle} from "../detalle";
import {AngularFireDatabase} from "angularfire2";
@Injectable()
export class DetallesService {

    sdkDb:any;

    constructor(private db: AngularFireDatabase, @Inject(FirebaseRef) fb) {
        this.sdkDb= fb.database().ref();
    }

    findAllDetalles(): Observable<Detalle[]> {
        return this.db.list('detalle');
    }

import { Injectable } from '@angular/core';
import {Observable} from "rxjs/Rx";
import {AngularFireDatabase} from "angularfire2";
import {Cuidador} from "../Cuidador";
import {Detalle} from "../detalle";
import {FirebaseListFactoryOpts} from "angularfire2/interfaces";
@Injectable()
export class CuidadorService {

    constructor(private db: AngularFireDatabase) { }

    findAllCuidador(): Observable<Cuidador[]> {
        return this.db.list('cuidador');
    }

    findAllDetallesForCuidador(cuidadorUrl: string): Observable<Detalle[]> {

        return this.db.list('detalle', {
            query: {
                orderByChild: 'url',
                equalTo: cuidadorUrl
            }
        })
    }
}
```

llamamos la función para guardar los datos en detalles

```
import { Component, OnInit } from '@angular/core';
import { DetallesService } from '../shared/model/detalles.service';
import { Observable } from "rxjs/Rx";
import { Detalle } from '../shared/model/detalle';

@Component({
  selector: 'app-detalles',
  templateUrl: './detalles.component.html',
  styleUrls: ['./detalles.component.css']
})
export class DetallesComponent implements OnInit {

  detalles$: Observable<Detalle[]>;
  constructor(private detallesService : DetallesService) { }

  ngOnInit() {

    this.detalles$=this.detallesService.findAllDetalles();

  }
}
```

```

import [{ Component, OnInit } from '@angular/core';
import { DetallesService } from '../shared/model/detalles.service';
import { Observable } from "rxjs/Rx";
import { Detalle } from '../shared/model/detalle';
import { ActivatedRoute, Router } from "@angular/router";
@Component({
  selector: 'app-detalle-cuidador',
  templateUrl: 'detalle-cuidador.component.html',
  styleUrls: ['detalle-cuidador.component.css']
})
export class DetalleCuidadorComponent implements OnInit {

  detalles: Detalle[];
  constructor(
    private route:ActivatedRoute,
    private router:Router,
    private detalleService : DetallesService) {

  }

  ngOnInit() {
    const cuidadorUrl = this.route.snapshot.params['id'];

    this.detalleService.findAllDetallesForCuidador(cuidadorUrl)
      .do(console.log)
      .subscribe(
        detalles => this.detalles = detalles
      );
  }
}

```

En detalle-cuidador.component.html creamos un bucle para poder recorrer los datos y mostrarlos.

```
<h1> Detalles </h1>
<table class="table lessons-list card card-strong">
  <tbody>
    <tr style="background-color: black; color:white ">
      <td>Descripcion</td>
      <td> Horas </td>
      <td> Animal </td>
    </tr>
    <tr *ngFor="let detallado of detalles">
      <td class="lessons-title">{{detallado.descripcion}}</td>
      <td class="duration">
        {{detallado.horas}}
        <span><i class="md-icon duration-icon">access_time</i> </span>
      </td>
      <td>
        {{detallado.animal}}
      </td>
    </tr>
  </tbody>
</table>
```

Buscador

```
import { Component, OnInit, Input } from '@angular/core';
import { CuidadorService } from '../shared/model/cuidador.service';
import { Cuidador } from '../shared/model/Cuidador';
@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent implements OnInit {
  allcuidador: Cuidador[];
  filtro: Cuidador[];
  constructor(private cuidadorService: CuidadorService) { }

  ngOnInit() {
    this.cuidadorService.findAllCuidador()//ESTO ES UNA FUNCION QUE RETURN UNA LISTA CON Cuidador
    .do(console.log)
    //hacemos el filtrado
    .subscribe(
      cuidador => this.allcuidador = this.filtro = cuidador
    );
  }
  // ESTA funcion se activa cada vez que escriben en el buscador
  search(search:string) {
    this.filtro= this.allcuidador.filter(cuidador => cuidador.nombre.includes(search));
  }
}
```


Creamos y configuramos el archivo **router.config.ts**

```
import {Route} from "@angular/router";
import {HomeComponent} from "../home/home.component";
import {DetallesComponent} from "../detalles/detalles.component";
import {DetalleCuidadorComponent} from "../detalle-cuidador/detalle-cuidador.component";
import {NuevoDetalleComponent} from "../nuevo-detalle/nuevo-detalle.component";

export const routerConfig : Route[] = [
  {
    path:'home',
    children: [
      {
        path:'id',
        component:DetalleCuidadorComponent
      },
      {
        path:'',
        component:HomeComponent
      }
    ]
  },
  {
    path:'detalles',
    children: [
      {
        path: '',
        component: DetallesComponent
      },
      {
        path:'new',
        component:NuevoDetalleComponent
      }
    ]
  }
]
```

Creamos la barra de navegación

```
<header class="l-header v-center-parent" style="...">
  

  <ul class="top-menu disable-link-styles" style="...">
    <li>
      <a routerLink="home"
        routerLinkActive="menu-active">Inicio</a>
    </li>
    <li>
      <a routerLink="detalles"
        routerLinkActive="menu-active" >Detalle</a>
    </li>
  </ul>
</header>
```

Por ultimo creamos los formularios y sus validaciones para añadir los Detalles.

```
<form [formGroup]="form" autocomplete="off" novalidate class="lesson-form">

  <table class="table lessons-list card card-strong">
    <tr style="Background:Black; color:white;">
      <td colspan="2"> Formulario</td>
    </tr>
    <tr>
      <td> Id_cuidador </td>
      <td>
        <input name="title" formControlName="id_cuidador" placeholder="Id_cuidador">
        <div class="field-error-message"
          *ngIf="isErrorVisible('id_cuidador','required')">Este campo es necesario</div>
      </td>
    </tr>
    <tr>
      <td> Descripcion </td>
      <td>
        <input name="title" formControlName="descripcion" placeholder="Descripción">
        <div class="field-error-message"
          *ngIf="isErrorVisible('descripcion','required')">Este campo es necesario</div>
      </td>
    </tr>
    <tr>
      <td> Horas </td>
      <td>
        <input name="title" formControlName="horas" placeholder="Horas">
        <div class="field-error-message"
          *ngIf="isErrorVisible('horas','required')">Este campo es necesario</div>
      </td>
    </tr>
  </table>

  <div class="form-group">
    <input type="text" formControlName="nombre" placeholder="Nombre">
    <div class="field-error-message"
      *ngIf="isErrorVisible('nombre','required')">Este campo es necesario</div>
  </div>
  <div class="form-group">
    <input type="text" formControlName="apellido" placeholder="Apellido">
    <div class="field-error-message"
      *ngIf="isErrorVisible('apellido','required')">Este campo es necesario</div>
  </div>
  <div class="form-group">
    <input type="text" formControlName="telefono" placeholder="Teléfono">
    <div class="field-error-message"
      *ngIf="isErrorVisible('telefono','required')">Este campo es necesario</div>
  </div>
  <div class="form-group">
    <input type="text" formControlName="email" placeholder="Email">
    <div class="field-error-message"
      *ngIf="isErrorVisible('email','required')">Este campo es necesario</div>
  </div>
  <div class="form-group">
    <input type="text" formControlName="password" placeholder="Contraseña">
    <div class="field-error-message"
      *ngIf="isErrorVisible('password','required')">Este campo es necesario</div>
  </div>
  <div class="form-group">
    <input type="text" formControlName="confirm_password" placeholder="Confirmar Contraseña">
    <div class="field-error-message"
      *ngIf="isErrorVisible('confirm_password','required')">Este campo es necesario</div>
  </div>
  <div class="form-group">
    <input type="button" value="Guardar" (click)="guardar()">
  </div>
  <div class="form-group">
    <input type="button" value="Cancelar" (click)="cancelar()">
  </div>
</form>

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from "@angular/router";
import { DetallesService } from "../shared/model/detalles.service";

@Component({
  selector: 'app-detalle',
  templateUrl: './nuevo-detalle.component.html',
  styleUrls: ['./nuevo-detalle.component.css']
})
export class NuevoDetalleComponent implements OnInit {

  constructor(private route:ActivatedRoute, private detallesService: DetallesService) { }

  ngOnInit() {
  }

  save(form){
    this.detallesService.createNewDetalle(form.value)
    .subscribe(
      () => {
        alert("Se ha creado correctamente");
        form.reset();
      },
      err => alert(`error al crear el detalle ${err}`)
    );
  }
}
```

Constructor

```
import ...

@Component({
  selector: 'app-formulario-nuevo',
  templateUrl: './formulario-nuevo.component.html',
  styleUrls: ['./formulario-nuevo.component.css']
})
export class FormularioNuevoComponent implements OnInit {

  form: FormGroup;

  constructor(private fb: FormBuilder) { }

  ngOnInit() {
    this.form = this.fb.group({
      descripcion: ['', Validators.required],
      horas: ['', Validators.required],
      id_cuidador: ['', Validators.required],
      animal: ['', Validators.required],
      url: ['', Validators.required]
    });
  }

  isErrorVisible(field: string, error: string) {

    return this.form.controls[field].dirty
      && this.form.controls[field].errors &&
      this.form.controls[field].errors[error];
  }

  reset() {
    this.form.reset();
  }
}
```