

Manual de usuario

Rutas API:

- Rutas principales

```
const routes: Routes = [  
  { path: "comunidad", component: ComunidadComponent },  
  { path: "provincias", component: ProvinciasComponent },  
  { path: "municipios", component: MunicipiosComponent },  
  { path: "grafico", component: Grafico01Component },  
  { path: "graficos-circulares", component: Grafico02Component },  
  { path: "", redirectTo: "comunidad", pathMatch: "full" }  
];
```

Código de Servicios:

- Comunidad

```
export class ComunidadService {  
  
  // Url generada por heroku para nuestro servicio de comunidades  
  private url = "https://restapicovid.herokuapp.com/comunidades";  
  
  constructor(private http: HttpClient) { }  
  
  // Listado de comunidades API  
  getComunidadApi() {  
    return this.http.get(this.url);  
  }  
  
  // Insertar de comunidades API  
  insertarComunidad(doc: any) {  
    return this.http.post(this.url, doc);  
  }  
  
  // Eliminar de comunidades API  
  eliminarComunidad(comunidad: Comunidad) {  
    const urlId = `${this.url}/${comunidad.id}`;  
    return this.http.delete(urlId);  
  }  
}
```

- Provincias

```
export class ProvinciaService {

    // Url generada por heroku para nuestro servicio de
    provincias
    private url =
    "https://restapicovid.herokuapp.com/provincias";

    constructor(private http: HttpClient) { }

    // Listado de provincias API
    getProvinciaApi() {
        return this.http.get(this.url);
    }

    // Insertar de provincias API
    insertarProvincia(doc: any) {
        return this.http.post(this.url, doc);
    }

    // Eliminar provincia desde API
    eliminarProvincia(provincia: Provincia) {
        const urlId = `${this.url}/${provincia.id}`;
        return this.http.delete(urlId);
    }
}
```

- Municipios

```
export class MunicipioService {

    // Url generada por heroku para nuestro servicio de municipios
    private url = "https://restapicovid.herokuapp.com/municipios";
    constructor(private http: HttpClient) { }

    // Listado de municipios API
    getMunicipioApi() {
        return this.http.get(this.url);
    }
}
```

```

    // Insertar de municipios API
    insertarMunicipio(doc: any) {
        return this.http.post(this.url, doc);
    }

    // Eliminar de municipios API
    eliminarMunicipio(municipio: Municipio) {
        const urlId = `${this.url}/${municipio.id}`;
        return this.http.delete(urlId);
    }
}

```

Modelos:

- Comunidad

```

export class Comunidad {

    // Atributos de la clase Provincia
    id: number;
    nombre: string;

    // Constructor de la clase Provincia
    constructor(
        id: number,
        nombre: string,
    ) {
        this.id = id;
        this.nombre = nombre;
    }
}

```

- Provincias

```

export class Provincia {

    id: number;
    comunidad: number;
    nombre: string;
    habitantes: number;

    constructor(id: number,comunidad: number,nombre: string,
        habitantes: number) {
        this.id = id;
    }
}

```

```

        this.comunidad = comunidad;
        this.nombre = nombre;
        this.habitantes = habitantes;
    }
}

```

- Municipios

```

export class Municipio {

    // Atributos de la clase Provincia
    id: number;
    provincia: number;
    nombre: string;
    confirmados: number;
    recuperados: number;
    criticos: number;
    fallecidos: number;

    // Constructor de la clase Municipio
    constructor(
        id: number,
        provincia: number,
        nombre: string,
        confirmados: number,
        recuperados: number,
        criticos: number,
        fallecidos: number,
    ) {
        this.id = id;
        this.provincia = provincia;
        this.nombre = nombre;
        this.confirmados = confirmados;
        this.recuperados = recuperados;
        this.criticos = criticos;
        this.fallecidos = fallecidos;
    }
}

```

Colecciones (Capturas desde Mongo Atlas):

- Comunidades

comunidades

municipios

provincias

QUERY RESULTS 1-6 OF 6

```
_id: ObjectId("602e7de972731d0a7ceff52b")
id: 1
nombre: "Andalucia"
__v: 0
```

```
_id: ObjectId("602e7e2872731d0a7ceff52c")
id: 2
nombre: "Madrid"
__v: 0
```

```
_id: ObjectId("602e7e3972731d0a7ceff52d")
id: 3
nombre: "Cataluña"
__v: 0
```

```
_id: ObjectId("602e7e7172731d0a7ceff52e")
id: 4
nombre: "Valenciana"
__v: 0
```

```
_id: ObjectId("602f672a33ca762ab486e2f8")
id: 5
nombre: "Extremadura"
__v: 0
```

```
_id: ObjectId("603cf68d0bbb4b001546c252")
id: 6
nombre: "Asturias"
__v: 0
```

- Provincias

comunidades

municipios

provincias

QUERY RESULTS 1-5 OF 5

```
_id: ObjectId("603d55ba3888ae00155244ff")
id: 1
comunidad: 1
nombre: "Sevilla"
habitantes: 1960000
__v: 0
```

```
_id: ObjectId("603d55cc3888ae0015524500")
id: 2
comunidad: 1
nombre: "Malaga"
habitantes: 1640000
__v: 0
```

```
_id: ObjectId("603d55da3888ae0015524501")
id: 3
comunidad: 1
nombre: "Cadiz"
habitantes: 1230000
__v: 0
```

```
_id: ObjectId("603d560d3888ae0015524502")
id: 4
comunidad: 5
nombre: "Badajoz"
habitantes: 672493
__v: 0
```

- Municipios

comunidades

municipios

provincias

QUERY RESULTS 1-5 OF 5

```
_id: ObjectId("603d57203888ae0015524504")
id: 1
provincia: 1
nombre: "Guillena"
confirmados: 525
recuperados: 249
criticos: 26
fallecidos: 9
__v: 0
```

```
_id: ObjectId("603d576d3888ae0015524505")
id: 2
provincia: 1
nombre: "Gerena"
confirmados: 293
recuperados: 215
criticos: 18
fallecidos: 1
__v: 0
```

```
_id: ObjectId("603d57c53888ae0015524506")
id: 3
provincia: 1
nombre: "Garrobo"
confirmados: 59
recuperados: 36
criticos: 8
fallecidos: 1
__v: 0
```

Interfaces:







- Comunidad

PROYECTO COVID

Comunidad Provincias Municipios Gráficos de Barras Gráficos Circulares

Comunidades

Teclee el nombre de la comunidad

#	Nombre	Acciones
1	Andalucía	
2	Madrid	
3	Cataluña	
4	Valenciana	
5	Extremadura	
6	Asturias	





- Provincias

PROYECTO COVID

Comunidad Provincias Municipios Gráficos de Barras Gráficos Circulares

Provincias

Andalucía Teclee el nombre de la provincia Teclee el número de habitante

ID	Comunidad	Nombre	Habitantes	Acciones
1	1	Sevilla	1960000	
2	1	Malaga	1640000	
3	1	Cadiz	1230000	
4	5	Badajoz	672493	
5	5	Caceres	1230000	

- Municipios

PROYECTO COVID

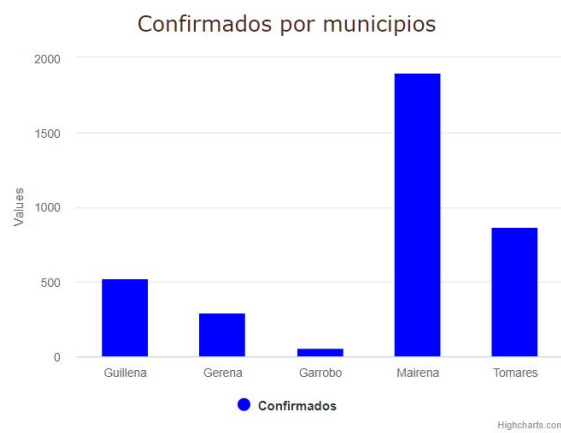
[Comunidad](#)[Provincias](#)[Municipios](#)[Gráficos de Barras](#)[Gráficos Circulares](#)

Municipios

ID	Provincia	Nombre	Confirmados	Recuperados	Críticos	Fallecidos	Total	Acciones
1	1	Guillena	525	249	26	9	809	
2	1	Gerena	293	215	18	1	527	
3	1	Garrobo	59	36	8	1	104	
4	1	Mairena	1900	1360	186	32	3478	
5	1	Tomares	864	586	102	10	1562	

- Gráficas de Barras:

PROYECTO COVID

[Comunidad](#)[Provincias](#)[Municipios](#)[Gráficos de Barras](#)[Gráficos Circulares](#)

- Gráficos Circulares:

PROYECTO COVID

Comunidad

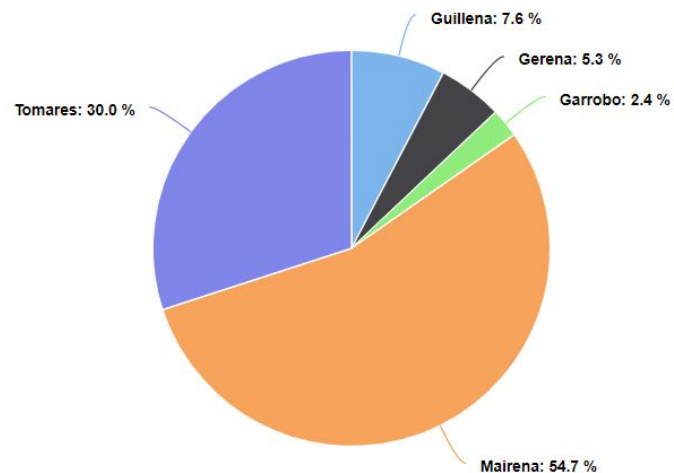
Provincias

Municipios

Gráficos de Barras

Gráficos Circulares

% de Casos Críticos por Municipios (340 casos críticos totales)



Conexión a BD:

```
class DataBase {  
  
  private _cadenaConexion: string = 'mongodb+srv://josanProyecto:proyectoFinal@cluster0.vrka3.mongodb.net/proyectoFinal?retryWrites=true&w=majority';  
  constructor(){  
  }  
  set cadenaConexion(_cadenaConexion: string){  
    this._cadenaConexion = _cadenaConexion  
  }  
  
  conectarBD = async () => {  
    const promise = new Promise<string>((resolve, reject) => {  
      await mongoose.connect(this._cadenaConexion, {  
        useNewUrlParser: true,  
        useUnifiedTopology: true,  
        useCreateIndex: true, // Para que cree el índice único asociado al campo unique  
        useFindAndModify: false // para usar findOneAndDelete y findAndModify  
      })  
      .then(() => resolve('Conectado a ${this._cadenaConexion}'))  
      .catch((error) => reject('Error conectando a ${this._cadenaConexion}: ${error}'))  
    })  
    return promise  
  }  
  
  desconectarBD = async () => {  
    const promise = new Promise<string>((resolve, reject) => {  
      await mongoose.disconnect()  
      .then(() => resolve('Desconectado de ${this._cadenaConexion}'))  
      .catch((error) => reject('Error desconectando de ${this._cadenaConexion}: ${error}'))  
    })  
    return promise  
  }  
}  
  
export const db = new DataBase()
```