



## Tarea 3: Base de Datos para desarrollo

En esta tareas crearemos una BD local con los datos de la tarea anterior. Usaremos el [ORM](#), una capa de software encima de la BD que uniformiza su manejo, y nos va a permitir tener distintas BDs en la fase de desarrollo y en la de producción, sin cambiar el código.

Para la fase de desarrollo usaremos [SQLite](#), que no necesita instalarse ni consume recursos. En la fase de producción ya usariamos una BD de mayor capacidad como [PostgreSQL](#) o alguna [BD gestionada](#).

Usaremos el ORM [Prisma](#) que [maneja estas BDs](#) e incluso alguna [NO-SQL](#), como [MongoDB](#). Todo el software que manejaremos será open source.

Seguimos los pasos de la documentación de [Prisma](#) en [Quickstart](#). Instalamos [prisma](#) como dependencia de desarrollo y generamos el cliente de [prisma](#)

```
> pnpm i -D prisma
> pnpm exec prisma init --datasource-provider sqlite
```

## Esquema de la BD

El archivo `./prisma/schema.prisma`, con la información de la BD y su esquema queda:

```
generator client {  
  provider = "prisma-client-js"  
}  
  
datasource db {  
  provider = "sqlite"  
  url      = "file:obras.sqlite"  
}  
  
model Obra {  
  id          Int @id @default(autoincrement())  
  título      String  
  imagen      String  
  descripción String  
  procedencia String  
  comentario  String  
}
```

Y

```
> pnpm exec prisma migrate dev --name esquema_inicial
```

Para generar las tablas. Este proceso habrá que hacerlo cada vez que se cambie el esquema para crear tablas nuevas y copiar el contenido de las antiguas.

## El script **seed.mjs**:

```
// seed.mjs
import { readFileSync } from "fs"
import { PrismaClient } from '@prisma/client'
const prisma = new PrismaClient()

const obras = leer_desde('./obras.json')

for (const obra of obras) {
  // bajarse las imágenes y guardarlas en ./public/images/
  // el servidor de las imágenes es https://www.museosdeandalucia.es
  // no https://museosdeandalucia.es
  // p.e.
  // https://www.museosdeandalucia.es/image/journal/article?img_id=682

  // guardar en la BD los datos
}
```

## Prisma Studio

Prisma incluye un cliente gráfico [prisma studio](#), que se puede usar para inspeccionar y tener acceso a cada tabla, se accede

```
> pnpm exec prisma studio
```

## Referencias:

- [PRISMA 101 : A beginner's guide to understand prisma](#)

- [SQLite vs PostgreSQL: 8 Critical Differences](#)
- [Prisma Schema](#)
- [Prisma SQLite](#)
- [Prisma Client CRUD](#)
- [Prisma and Express](#)
- [Download file with fetch in Node.js](#)