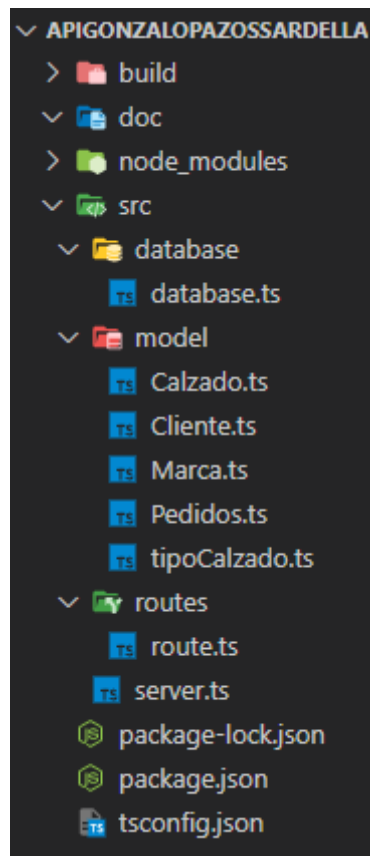


REST API en Heroku

El primer paso que hemos de hacer es crear nuestro proyecto. En este incluiremos tanto el apartado del código como de esta presentación.

En el apartado del código lo separo por database, donde se encuentra la conexión a la base de datos. Model, donde están los esquemas de lo*s objetos los cuales servirán para nuestra API. Y por último las rutas donde estableceremos las funciones de nuestra API y sus rutas para el posterior funcionamiento en PostMan.



Antes de todo esto debemos instalar todas las cosas necesarias para que nuestra API funcione correctamente. Con este comando podremos instalar todas las funciones necesarias: `npm install @types/node @types/cors @types/express @types/morgan nodemon -D`

Una vez instalado todo lo que necesitemos, continuaremos con el funcionamiento de nuestra API.

Por ejemplo, en mi API, una de las funciones que encontramos es la búsqueda de un pedido específico. Esto lo haremos primero haciendo una búsqueda del pedido a través de su número identificador. Y después de la búsqueda dentro de todos los pedidos existentes, especificando el que queramos.

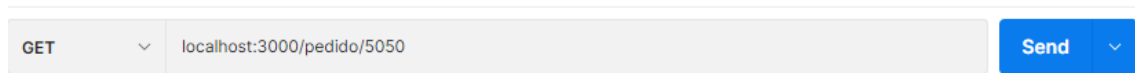
```
private getPedido = async (req: Request, res: Response) => {
  const nPedido = req.params.nPedido
  await db.conectarBD()
  .then(async (mensaje) => {
    const query = await Pedidos.findOne({_nPedido: nPedido})
    res.json(query)
  })
  .catch((mensaje) => {
    res.send(mensaje)
  })
  await db.desconectarBD()
}
```

Para poder hacer que esto funcione, en mi caso, usare PostMan. Una herramienta que nos permite hacer HTTP requests. Es decir, podemos a través de unos parámetros recoger información por ejemplo en mi caso de una colección de una base de datos.

Para recibir esta información primero deberemos establecer una ruta en la cual escribiremos los parámetros que queramos recoger.

```
this._router.get('/pedido/:nPedido', this.getPedido),
```

Una vez establecida la ruta podemos comprobar si esta funciona correctamente. Primero deberemos establecer el método que usaremos, en mi caso será el GET. Una vez hecho esto estableceremos la conexión con la base de datos. Por último, los parámetros que serian de los pedidos el número de pedido 5050.



A screenshot of a REST client interface. On the left, a dropdown menu shows 'GET'. In the center, the URL 'localhost:3000/pedido/5050' is entered. On the right, there is a blue 'Send' button with a small downward arrow next to it.

Como vemos este es un pedido existente y nos devuelve toda la información que le hayamos pedido, en este caso todo el pedido completo.



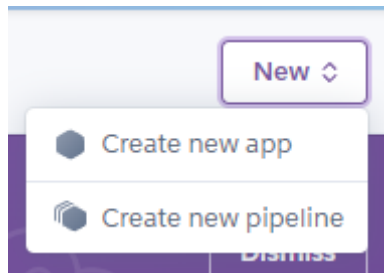
```
{
  "_id": "61d4acd92ca6ad547672967e",
  "_nPedido": 5050,
  "_objetos": [
    {
      "objectID": "61d4ac652ca6ad5476729670",
      "Marca": "Adidas",
      "Tipo": "Chancla",
      "Talla": 3,
      "Color": "Rojo",
      "Precio": 31,
      "Cantidad": 1
    }
  ],
  "_fecha": "2022-01-04T20:23:52.743Z",
  "_importeTotal": 31,
  "__v": 0
}
```

A screenshot of a JSON response displayed in a REST client. The JSON is formatted with syntax highlighting and a vertical line on the left for folding. It represents a single order with one item.

Al igual que existe el método GET, existen otros cuyas funcionalidades son el subir datos, recibirlos, modificarlos o borrarlos.


Una vez terminada nuestra API, deberemos subirla primero a GitHub y después a Heroku.

Primero crearemos una nueva APP donde pondremos nuestra API.





Estableceremos la región y un nombre.

App name



apigonzalopazossardella is available

Choose a region

 Europe 

Add to pipeline...

Create app

Por último usaremos GitHub para poder desplegar nuestra API.

Deployment method

