

Manual de programador

El proyecto está desarrollado pensando en la arquitectura MEAN

- Mongo
- Express
- Angular
- Node

Por lo que se utilizaron todas las tecnologías para poder crear un sistema gestor de productos bastante básico pero que permite crecer si así lo desea el cliente.

El proyecto se parte en dos grandes sectores

- Cliente
- Servidor

Comenzando por el primero (Cliente) el cual contiene:

- **Estructura de Angular**

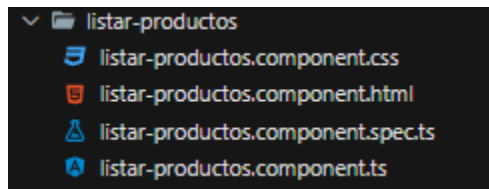
División en componentes.

Se decidió trabajar por componentes por la facilidad de poder integrarlos y utilizarlos en caso de ser necesario, los formularios pueden ser genéricos y hasta bastante repetitivos por lo que se puede partir de la base que se tiene para integrar más.

Para crear un nuevo componente en el proyecto basta con ubicarnos en la carpeta del cliente y ejecutar el siguiente comando

```
ng generate component [name]
```

A continuación se creará una carpeta dentro de components que tendrá los siguientes archivos:



Los cuales podrán ser modificados para insertar código en los formatos especificados, podrán agregarse más archivos si se desea pero tendrán que ser incluidos en el archivo typescript.

- **Estructura de Node**

Paqueterías

Para hacer uso de los paquetes para utilizar Angular y Node se recurre al archivo package.json que contiene todos los índices necesarios para que en cualquier máquina el proyecto funcione tras ejecutar el comando.

```
npm install
```

Para el proyecto no se instalaron demasiadas paqueterías, solamente las básicas para estilos e iconos

- **Api**

Conexión a consulta y modificación de datos almacenados en Mongo.

Para ejecutar el lado del cliente se tendrá que ejecutar el siguiente comando

```
ng serve --open
```

ó

```
ng serve --o
```

Para el Servidor se tiene una estructura más sencilla, la cual contiene:

- **Express**

Con el cual podremos hacer uso de peticiones de cualquier tipo mediante una conexión a una base de datos.

Para poder modificar o crear nuevas peticiones mediante rutas definidas en el Router de Express se tendrán que hacer los siguientes pasos.

Especificar en el archivo raíz de que archivo de rutas se definirán las consultas.

Archivo index.js

```
app.use('/api/productos', require('./routes/producto'));
```

El ejemplo anterior utiliza una ruta a partir de un archivo externo que contiene las subrutas en las cuales se define el tipo de petición

Archivo routes/producto.js

```
const express = require('express');
const router = express.Router();
const productoController = require('../controllers/productoController');

//api/producto
router.post('/', productoController.crearProducto);

router.get('/', productoController.obtenerProductos);
router.put('/:id', productoController.actualizarProducto);
router.get('/:id', productoController.obtenerProducto);
router.delete('/:id', productoController.eliminarProducto);

module.exports = router;
```

Dentro del archivo se especifica el tipo de consulta a realizar según la subruta, para que las consultas puedan ejecutarse se necesita de una función que indique que se necesita extraer, consultar o modificar.

Como mejor práctica se utilizó un controlador, el cual es recomendable por cada archivo de rutas que se utilice en el proyecto para especificar las funciones con las instrucciones.

Archivo controllers/productController

```
const Producto = require("../models/Producto");

exports.crearProducto = async (req, res) => {
  try {
    let producto = new Producto(req.body);
    await producto.save();
    res.send(producto);
  } catch (error) {
    console.log(error);
    res.status(500).send('Hubo un error');
  }
}

exports.obtenerProductos = async (req, res) => {
  try {
    const productos = await Producto.find();
    res.json({ productos });
  } catch (error) {
    console.log(error);
    res.status(500).send('Hubo un error');
  }
}
```

Cada función retorna una respuesta de tipo JSON si fue exitosa o un error diferenciador.

- **Mongo**

Para la conexión al cluster de la base de datos basta con especificar una variable global en el archivo .env que contenga la liga para establecer una comunicación

Archivo .env

```
DB_MONGO=mongodb+srv://josue:admin123@cluster0.qn4oafr.mongodb.net/meanproductos
```

Para ejecutar el lado del servidor se tendrá que ejecutar el siguiente comando

```
npm run dev
```