

Práctica: Angular ~ OniiExpress



Índice:

- 1. Consideraciones Previas:**
- 2. Explicación General de la práctica:**
- 3. Proyecto: OniiExpress**
 - a. API-Rest**
 - b. Servidor**

1. Consideraciones Previas:

Partiremos de la base que tenemos de los anteriores proyectos. No se explicará en este documento ningún conocimiento ya comentado en alguna práctica anterior.

Ni como utilizar GIT, NPM, fundamentos básicos sobre Mongo o incluso el cómo programar en Typescript e implementar una API-Rest serán explicados en la documentación de la práctica.

2. Explicación General de la práctica:

Se ha realizado una aplicación que cubra las necesidades de una tienda-web, tal y como pudiera ser aliexpress.

La aplicación viene dividida en dos, una orientada al cliente de la página y otra parte para los empleados y focalizada en la gestión de los productos.

En la primera parte del proyecto tendremos opciones para visualizar los productos, buscar y tendremos acceso también a 2 gráficas para analizar los distintos productos pudiendo filtrar por nota media o por el precio de los mismos productos.

En la segunda parte, tendremos opción a crear productos y eliminar los ya existentes.

3. Proyecto: OniiExpress:

a. API-Rest

En la API-Rest tendremos las siguientes rutas para gestionar el proyecto.

Estas rutas nos permitirán gestionar los datos de productos.

```

src > routes > routes.ts
package.json
DataRoutes
misRutas

560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576

misRutas(){
  this._router.get('/Productos/get', this.getProd)
  this._router.get('/Productos/precios', this.getPrecios)
  this._router.post('/Productos/add', this.addProd)
  this._router.post('/Productos/ropa/add', this.addRopa)
  this._router.post('/Productos/movil/add', this.addMovil)
  this._router.post('/Productos/procesador/add', this.addProcesador)
  this._router.get('/Productos/search/:id', this.searchProd)
  this._router.get('/Productos/prueba', this.prueba)
  this._router.delete('/Productos/delete/:id', this.deleteProd)

  this._router.post('/Users/add', this.addUsers)
  this._router.get('/Users/search/:id', this.addUsers)
  this._router.delete('/Users/delete', this.deleteUsers)
  this._router.get('/Users/list', this.listUsers)
}

```

Dentro de nuestro método `getPrecios` calcularemos mediante polimorfismo los distintos costes de cada producto para luego enviarlo todo a la aplicación servidor para que cree un high chart a partir de los datos.

```

92
93 private getPrecios = async (req: Request, res: Response) => {
94   await db.conectarBD()
95   .then( async (mensaje) => {
96     let dProducto: iProducto2
97     const query = await ProductoDB.find({})
98     let query2: Array<iProducto2> = []
99     for(dProducto of query) {
100       let miProducto
101       if(dProducto._CategoriaProducto == "Movil") {
102         miProducto = new Movil(dProducto._id, dProducto._NombreProducto, dProducto._CategoriaProducto, dProducto._PrecioBase)
103       } else if (dProducto._CategoriaProducto == "Procesador") {
104         miProducto = new procesador(dProducto._id, dProducto._NombreProducto, dProducto._CategoriaProducto, dProducto._PrecioBase)
105       } else if (dProducto._CategoriaProducto == "Ropa") {
106         miProducto = new Ropa(dProducto._id, dProducto._NombreProducto, dProducto._CategoriaProducto, dProducto._PrecioBase)
107       } else {
108         miProducto = new Producto(dProducto._id, dProducto._NombreProducto, dProducto._CategoriaProducto, dProducto._PrecioBase)
109       }
110       let coste = miProducto.calculoPrecio()
111       dProducto._PrecioBase=coste
112       query2.push(dProducto)
113       //Producto._PrecioBase= miProducto.calculoPrecio()
114       //console.log(Producto._PrecioBase)
115     }
116     res.json(query2)
117   })
118   .catch((mensaje) => {
119     res.send(mensaje)
120   })
121 }
122

```

b. Servidor

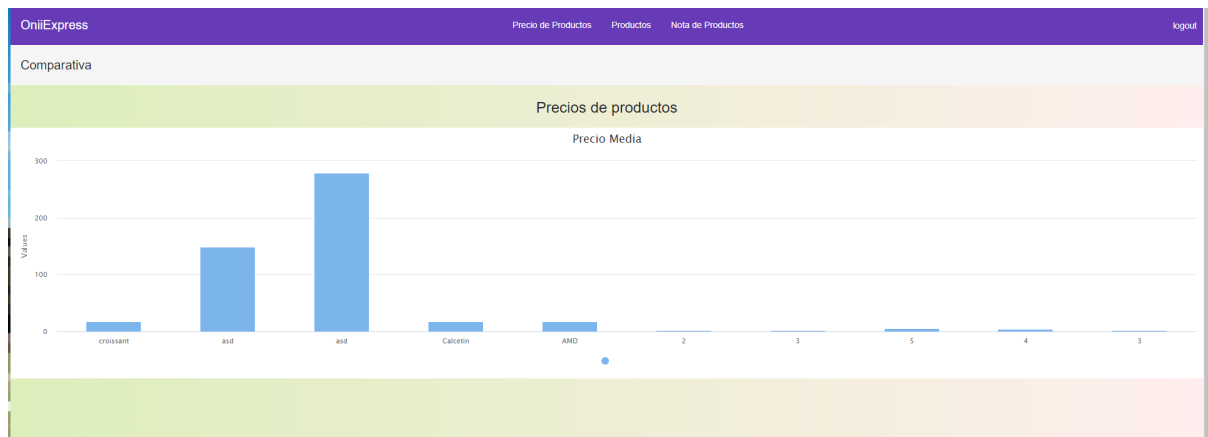
En primer lugar podremos listar los productos e incluso buscar.

OniiExpress					
Precio de Productos Productos Nota de Productos logout					
Listar Productos					
Filter					
ID	Nombre	Categoría	Precio	Nota	
8	croissant	Alimento	15	3	search
123	asd	varios	123	123	search
154132	asd	varios	231	23	search
211	Calcetin	Ropa	15	3	search
232	AMD	Procesador	15	3	search
2	2	2	2	2	search
3	3	Ropa	2	2	search
5	5	varios	5	5	search
4	4	varios	4	4	search
22	3	Ropa	2	2	search

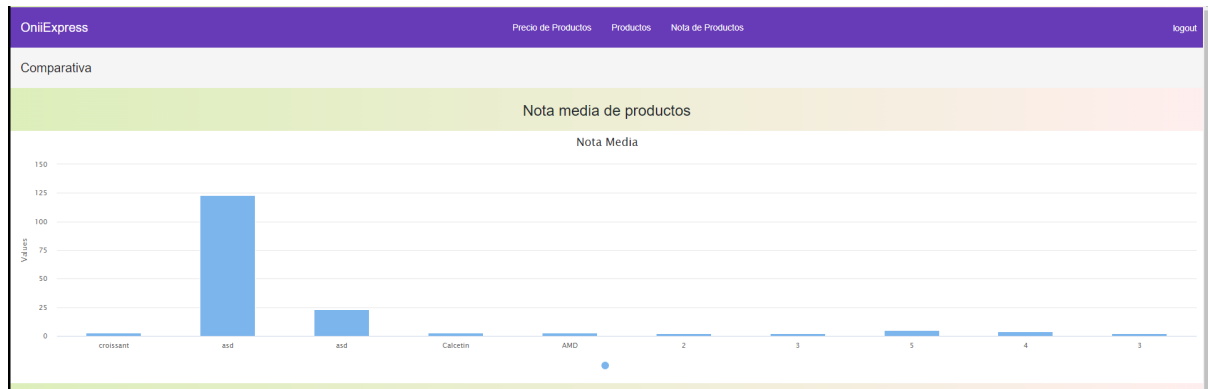
Podremos ver cada elemento de forma más concreta además.

OniiExpress	
Precio de Productos Productos Nota de Productos logout	
Información sobre croissant	
_Almacenamiento:	
_CategoríaProducto: Alimento	
_NombreProducto: croissant	
_NotaMedia: 3	
_PrecioBase: 15	
_Id: 8	
Precio Final: 18.15 €	

Tendremos acceso también a gráficas que nos mostrarán la nota media de nuestros productos



Nota media de los productos



En Administración tenemos la siguiente pantalla con un listado y la opción para eliminar cada elemento

OniiExpress Administración

Productos Clientes Random

Productos

Agregar Producto Genérico Agregar Móvil Agregar Procesador Agregar Ropa

Filter

ID	Nombre	Categoría	Precio	Nota	
8	croissant	Alimento	15	3	
123	asd	varios	123	123	
154132	asd	varios	231	23	
211	Calcetin	Ropa	15	3	
232	AMD	Procesador	15	3	
2	2	2	2	2	
3	3	Ropa	2	2	
5	5	varios	5	5	
4	4	varios	4	4	
22	3	Ropa	2	2	

Además de formularios para crear nuevos elementos

OniiExpress Administración

Productos Clientes Random

CREAR PRODUCTO

ID del producto

Nombre de producto

Nota media del producto

Precio de producto

Enviar