



SEGUNDA ENTREGA DEL PROYECTO FINAL

Deberás entregar el avance de tu aplicación eCommerce Backend correspondiente a la segunda entrega de tu proyecto final.

SEGUNDA ENTREGA DEL PROYECTO FINAL

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Proyecto
Final



>>Consigna: Basándose en los contenedores ya desarrollados (memoria, archivos) desarrollar dos contenedores más (que cumplan con la misma interfaz) que permitan realizar las operaciones básicas de CRUD en MongoDB (ya sea local o remoto) y en Firebase. Luego, para cada contenedor, crear dos clases derivadas, una para trabajar con Productos, y otra para trabajar con Carritos.

SEGUNDA ENTREGA DEL PROYECTO FINAL

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Proyecto
Final



>>Aspectos a incluir en el entregable:

- A las clases derivadas de los contenedores se las conoce como DAOs (Data Access Objects), y pueden ir todas incluidas en una misma carpeta de 'daos'.
- En la carpeta de daos, incluir un archivo que importe todas las clases y exporte una instancia de dao de productos y una de dao de carritos, según corresponda. Esta decisión se tomará en base al valor de una variable de entorno cargada al momento de ejecutar el servidor (opcional: investigar el uso de imports dinámicos).
- Incluir un archivo de configuración (config) que contenga los datos correspondientes para conectarse a las bases de datos o medio de persistencia que corresponda.

>>Opcional:

- Hacer lo mismo para bases de datos relacionales: MariaDB/SQLite3.

SEGUNDA ENTREGA DEL PROYECTO FINAL

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Proyecto
Final



Ejemplos

```
import { promises as fs } from 'fs'

class ContenedorArchivo {

  constructor(ruta) {
    this.ruta = ruta;
  }

  async listar(id) {
    const objs = await this.listarAll()
    const buscado = objs.find(o => o.id === id)
    return buscado
  }

  async listarAll() {
    try {
      const objs = await fs.readFile(this.ruta)
      return JSON.parse(objs)
    } catch (error) {
      return []
    }
  }
}
```

```
import ContenedorArchivo from '../.. /contenedores/C

class ProductosDaoArchivo extends ContenedorArchivo {

  constructor() {
    super('DB/productos.json')
  }

  async desconectar() {
  }
}

export default ProductosDaoArchivo
```

```
▼ src
  ▼ contenedores
    JS ContenedorArchivo.js
    JS ContenedorFirebase.js
    JS ContenedorMemoria.js
    JS ContenedorMongoDb.js
  ▼ daos
    ▼ carritos
      JS CarritosDaoArchivo.js
      JS CarritosDaoFirebase.js
      JS CarritosDaoMem.js
      JS CarritosDaoMongoDb.js
    ▼ productos
      JS ProductosDaoArchivo.js
      JS ProductosDaoFirebase.js
      JS ProductosDaoMem.js
      JS ProductosDaoMongoDb.js
  JS index.js
  > utils
  JS config.js
  JS main.js
  JS server.js
```