**PROYECTO 3: ECOMMERCE**

**WEB ECOMMERCE “SOFÍA KIDS”**

Mi hermana ha comenzado a vender accesorios de moda infantil en México a través de facebook y creo que sería una buena oportunidad que se abriera su propio espacio en el comercio electrónico.

La idea es desarrollar un proyecto tipo e-commerce donde el usuario pueda crear su cuenta e ingresar con su email y una contraseña, para poder ver los diferentes productos y comprar en línea.

Los productos se mostrarán por categorías y se dividirán en colecciones y temporadas.

Cada accesorio lo podrá conocer por:

* Foto-imagen
* Nombre
* Descripción
* Precio

**DIAGRAMA DE RELACIONES**

Modelos: Registro, usuario, productos, colecciones, temporadas

USER ADM

- Id

- Name

- Last name

- e-mail

- password

LOG IN

- Nickname

- e-mail

- password

- Facebook

SIGN IN

- Name

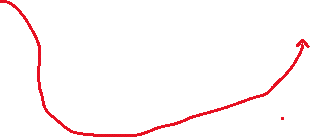
- Last name

- Nickname

- e-mail

- password

Relaciones:



PRODUCT

- Id

- Title

- Image

- Description

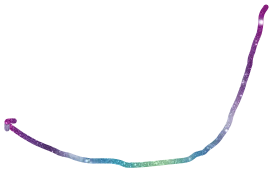
- Price

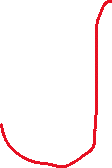
SEASONS

- Id

- Title

- Products [ ]





COLLECTIONS

- Id

- Title

- Products [ ]

- Price

USER

- Id

- Name

- Last name

- Nickname

- password

- e-mail

- Cart: [ ]



# Proyecto 3: Ecommerce

El proyecto tiene como objetivo desarrollar una página web tipo e-commerce donde el usuario pueda crear su cuenta e ingresar con su email y una contraseña, para poder ver los diferentes productos y comprar en línea.

Los productos se mostrarán por categorías y se dividirán en colecciones y temporadas.

Cada accesorio lo podrá conocer por:

• Foto-imagen

• Nombre

• Descripción

• Precio

• Id

* Crear un nuevo archivo: server.js

Abrir nueva terminal y comenzar con los siguientes comandos para inicializar…:

npm init

npm i express

npm i nodemon -D

* En el archivo server.js llamar a las funciones:

const express = require(“express”)

const app = express()

app.listen(5000, () => {

console.log(“Server is running on port 5000)

});

* Para ver si funcionas en la terminar poner el comando: **node server.js** ó **npm strat**
* En el archivo que ha creado: package.json en “scripts” crear el siguiente objeto

“dev”: “nodemon server.js”

* En la terminal con el comando: **npm run dev** arrancar la función del nodemon para ver en la terminal los cambios que se van realizando, como lo hace el “run server”. Al dar enter debería aparecer el console.log de “Server is running on port 5000”
* Para conectar con la Base de Datos (crear una cuenta en MongoDB y abrir un Clúster). En la terminal instalamos dos dependencias:

npm i dotenv mongoose

* Crear un nuevo archivo: **.env** que nos servirá para hacer cambios en nuestra Base de Datos, acceder a las variables del entorno y guardar nuestra clave del DB sin que puedan acceder a nuestra clave cuando subamos nuestro proyecto a Github.
* En el archivo **.env** creamos una variable:

MONGODB\_URL = url que copiaremos del clúster de MongoDB |Connect| 🡪 |Connect your app| copiarURL

(Cuando se crea y abre por primera vez el clúster: |Connect|🡪|Allow Access from anywhere| 🡪

|Add IP address|🡪|Choose a connection method| 🡪 |Connect your application|…)|… )

* Borramos 🡪 <password> y escribimos la contraseña que pusimos al crear el Clúster. Ej.:

MONGODB\_URL = mongodb+srv://Anais:007Montserrat777@cluster.mnmrrvw.mongodb.net/?retryWrites=true&w=majority

* Volvemos al documento server.js y agregamos el requerimiento de las dependencias que hemos agregado:

require(“dotenv”).config()

const mongoose = require(“mongoose”);

const URL = proccess.env.MONGODB\_URL

mongoose.connect(URL, {}).then(()=>{

console.log(“BBDD is now connected”)}).catch(error) => {

console.log(error);

});

* Para comprobar en la terminar donde está funcionando el **npm run dev** : al guardar el doc “server.js” debe aparecer primero el console.log(“Server is running on port 5000) y luego el de (“BBDD is now connected”). Si aparecen es que la Base de Datos ha sido conectada exitosamente, la contraseña es la correcta, la URL y el código están bien escritos.

HASTA EL MOMENTO NUESTRO CÓDIGO ESTÁ ASÍ:

const express = require("express")

const app = express()

require("dotenv").config();

const mongoose = require("mongoose");

const URL = process.env.MONGODB\_URL

mongoose.connect(URL, {}).then(() =>{

    console.log("BBDD is now connected")

}).catch((error) => {console.log(error);

});

app.listen(5000, ()=>{

    console.log("Server is running on port 5000")

})

Crear documento: server.js

Terminal:

npm init

npm i express

npm i nodemon -D

En server.js:

const express = require(“express”)

const app = express()

app.listen(5000, () =>{

console.log(“Server is running on port 5000);

}

* Guardar

En package.json:

En “scripts” agregar la siguiente key: value

“dev”: “nodemon server.js”

* Guardar

En la terminal:

npm run dev

--- Si es correcto debe aparecer el console.log del listen:

Server is running on port 5000

CONECTAR CON LA BASE DE DATOS

En una nueva terminal:

npm i dotenv

npm i mongoose

O

npm i dotenv mongoose

-- Verificar en package.json que estén agregados en “dependencies”, deben estar “dotenv”, “express” y “mongoose”

Crear un nuevo doc: **.env** y ahí crear la variable para la url de MongoDB

MONGODB\_URL = mongodb+srv://Pepe:Pepe007@peliculas.adllu5a.mongodb.net/?retryWrites=true&w=majority

* Guardar

En server.js:

require(“dotenv”).config();

const mongoose = require(“mongoose”);

const URL = proccess.env.MONGODV\_URL;

mongoose.connect(URL, {} ).then(()=>{

console.log(“DDBB is now connected”)

}).catch(error) => {

Console.log(error);

})

* Guardar

En la terminal donde está arrancado el dev debe aparecer:

Server is running on port 5000

DDBB is now connected

137424103191454225

XsEDuN6v