**Notas del curso**

**02. Workflow and debugging**

**1. Uso de npm**

npm init

npm install <package\_name>

npm install --save-dev nodemon

**2. Uso de nodemon**

npm start

npm run <script\_name> //Para custom scripts

**3. Debugging Node.js app con VSC**

- Config. extra:

#En el archivo '.vscode/launch.json' añadir:

#Permitirá reiniciar automáticamente el debugger

"restart": true,

"runtimeExecutable": "nodemon",

"console": "integratedTerminal"

**03. Express,js**

-**Middleware**

Se ejecuta de forma secuencial según el orden que exista en el código.Cualquier ruta irá a la '/' por lo que otra ruta debe ir antes del root. Para que la request continúe al sifg. middleware debe de usarse Next().

**- BodyParser**

**- Router de Express**

**- Filtering paths**

**- Servir contenido estáticamente**

app.use(express.static(path.join(\_\_dirname, "public")));

**04. Dynamic Content & Templating engines**

**pug**

**handlebars - hbs**

**ejs**

**Express.js app.set()**

Mantener variables de forma global en la aplicación.

En esta sección se vieron los anteriores *templating engines* y al final se limipió todo y sólo se dejó lo de *ejs*. Por lo que la carpeta *04 MVC* está limpia y sólo se usa *ejs*.

**05. The Model View Controller**

En esta sección el proyecto fue convertido al paradigma de Modelo Vista Controlador (MVC), se creó un modelo y controladores para las rutas existentes.

**06. Optional: Enhancing the app**

En esta sección se agregaron más características a la tienda online, como más atributos al modelo, más rutas, controladores y vistas para incrementar las funcionalidades de la tienda online.

**07. Dynamic routes & Advanced models**

**Passing route params**

**Using query params**

**?param=value&isActive=true**

Example:

/products/:productId?edit=true

/\* Access to the data \*/

//Params

req.params.productId

//Parameters

req.query.edit

**Enhance our models**

Se trabajó con métodos estáticos para no tener que crear varias instancias de un producto.

Se puede interactuar entre *models*.

Carpeta: 05

**08. SQL introduction**

Se aprendió a cómo conectar MySQL a la aplicación Node.js, se ejecutó el sig. comando:

npm install –-save mysql2

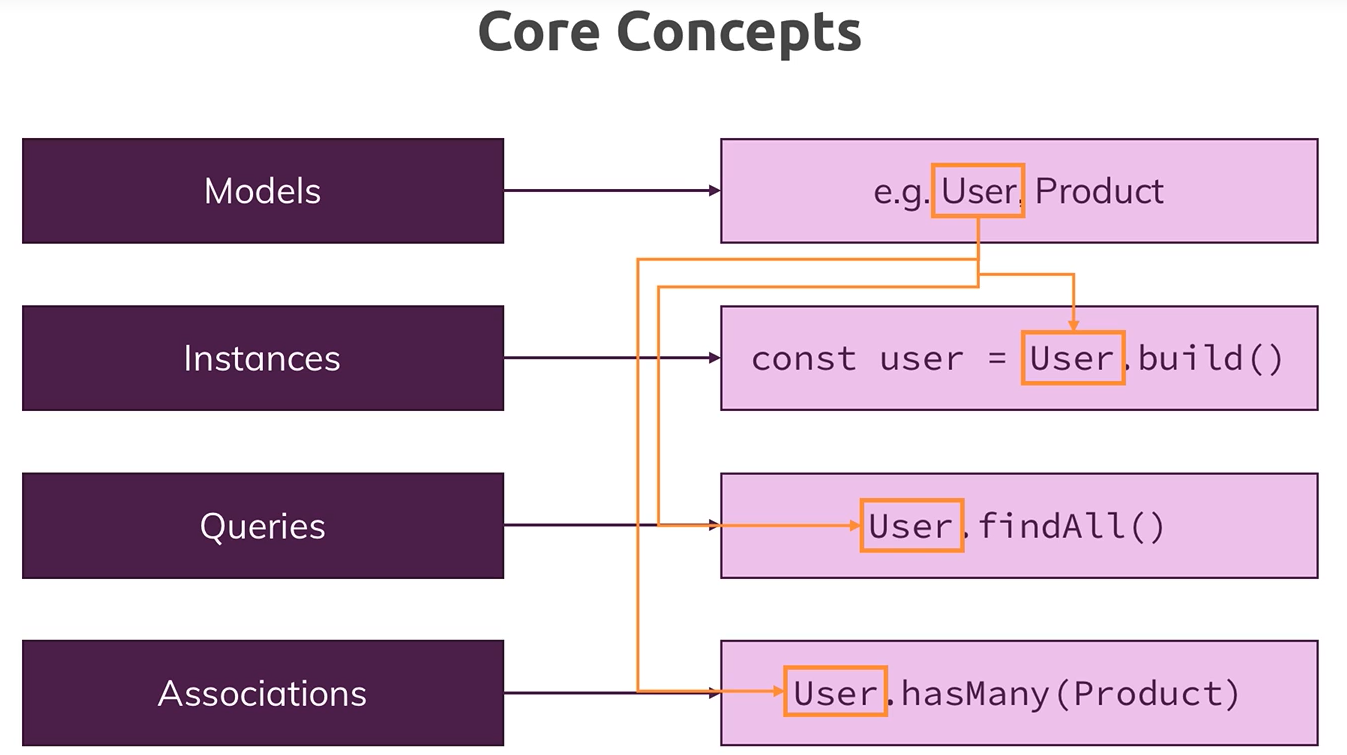
Se creó una base de datos llamada *nodejs*-*course* y una tabla llamada *products*, se hizo una conexión de tipo *pool* para conexiones persistentes sin necesidad de estar abriendo y cerrando la conexión. Se ejecutaron sentencias básicas para insertar y recuperar datos.

Se modificaron los modelos para trabajar con la BD en vez de archivos de texto.

Carpeta: 05, 06

**09. Sequelize**

Es un ORM – Object-Relational Mapping Library



**Instalación**

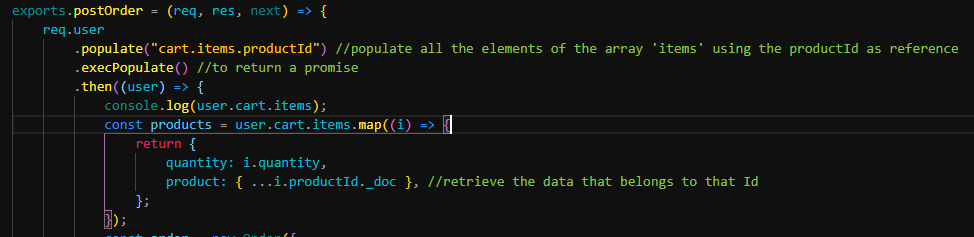
npm install –-save mysql2 //required

npm install --save sequelize

**10. NoSQL using MongoDB**

En esta sección se utilizó MongoDB como base de datos, fue más sencillo que utilizar sequelize, sin embargo, prefiero más mongoose porque es más sencillo y tiene métodos mejores para trabajar con la información.

Aún siento que la aplicación que se está creando no es bastante profesional, y aún se utilizan dummy values.

**11. Mongoose**

En esta sección se aprendió *mongoose* que realmente es bastante útil. Sería muy bueno leer la documentación, tiene bastantes cosas de utilidad.

**12. Sessions & Cookies**

