**UPTASK**

## NOTAS DEL PROYECTO:

Primer proyecto del curso de **Node.js** de **Udemy**:

<https://www.udemy.com/nodejs-bootcamp-desarrollo-web-mvc-y-rest-apis/learn/v4/content>

Primeros pasos:

* Crear carpeta con el nombre del proyecto.
* Consola: npm init
  + Crear el *package.json*
* Instalar dependencias:
  + npm install --save express
  + npm install --save-dev nodemon
* Crear *index.js*
  + Habilitar *express*:

const express = require(‘express’);

* + Configurar la aplicación:

const app = express();

* + Ruta para el inicio:

app.use(‘/’, (req, res) => {

res.send(‘Hola’);

});

* + Determinar puerto:

app.listen(3000);

* *Package.json*:
  + Formular “*scripts”*:

“start”: “nodemon ./index.js”;

* Crear carpeta + archivo *routes/index.js*
  + Ruta de enrutamiento en *index.js*

Const routes = require(‘./routes’);

app.use(‘/’, routes());;

* + *Routes/index.js*

const express = require(‘express’);

const router = express.Router();

module.exports = function() {

router.get(‘/’, (req, res) => {

res.send(‘Index’);

});

return router;

}

El sistema MODEL-VIEW-CONTROLLER (MVC):

* **Model**:

Encargado de los datos (desde una base de datos) y de la lógica destinada a mostrar esos datos.

*Ejemplo: un usuario quiere ver la sección de productos. El modelo se encarga de realizar la consulta pertinente a la base de datos con el fin de obtener los productos.*

* **View**:

Se encarga de lo que se imprime en la pantalla (html).

*Ejemplo: si el modelo hace la consulta a la base de datos para obtener los productos, la vista es la que se ocupa de mostrar los resultados obtenidos de la consulta a la BD.*

* **Controller**:

Intercede entre el modelo y la vista, antes de que el modelo realice la consulta a la base de datos, se encarga de llamar a la misma y, una vez obtenidos los resultados de la consulta a la base de datos, se ocupa de pasarlos o enviarlos a la vista.

* **Rputer**:

Encargado de soportar todas las URL’s o *endpoints* que gestiona la aplicación. Si el usuario accede a “*/productos*”, el Router llama a un controlador, que se encarga de comunicarse con el modelo (base de datos), para obtener los resultados de la consulta realizada y gestionar su envío a la vista que se ocupará de mostrar dicho resultado.