|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Carrera** | **Plan de estudios** | **Clave materia** | **Nombre de la materia** | **Semestre** | **Grupo** | **Periodo** |
| **ISC** | **ISIC-2010-** | DAM-1904 | Desarrollo de Aplicaciones para el  Comercio Electrónico | **9** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Practica No.** | **Laboratorio de:** | **Nombre de la practica** | **Duración (Hora)** |
| **3** | **CÓMPUTO** | APLICACIONES PARA EL LADO DEL SERVIDOR (conexión BD y creando servidor web) | **10HRS** |

|  |  |
| --- | --- |
| **Unidad temática: 1** | **Subtema:**  CONCEPTOS BÁSICOS DE LA HERRAMIENTA DE DESARROLLO DEL LADO DEL SERVIDOR |
| **Competencia especifica materia:**  Desarrolla aplicaciones para comercio electrónico en la nube utilizando software libre. | |
| **Competencia de la práctica:**  Crea y aplica funcionalidades especiales para aplicaciones de comercio electrónico  como proyecto final | |
| **Materiales:**  Visual Code, librerías API, Dependencias, REST | |
| **Desarrollo**  Vamos abrir mongodb y crear nuestra base de datos ejemplo  use proyecto  db.artista.save({nombre: 'Morrison', descripcion: 'musica Rock', imagen: 'null'});  Cuida el estilo de los apóstrofos, ya que si cambian te mandará error de sintaxis, muestra el registro con:  db. artista.find();  ahora, vamos a crear la conexión a la base de datos, vamos crear un fichero index.js  (en el ejemplo que se maneja en este manual usaré ejemplo.js)  Después de haber instalado mongose y node js (debería comprobar su funcionamiento aprendido en prácticas anteriores), abra visual code y escriba el siguiente código  var **mongoose** = **require**('mongoose');  **mongoose**.**connect**('mongodb://localhost:27017/proyecto', (err, res) => {      if (err) {          throw err;      } else {          console.**log**("conexión exitosa");      }  })  Modifique el package json para que al ejecutar npm start, automáticamente ejecute el programa con nodemon    El resultado de la ejecución se muestra en la siguiente imagen    Ahora, vamos a crear un servidor con node js, usando **Express,** va a ser el motor del backed, porque va a recibir las peticiones HTTP.  Creamos un nuevo fichero js llamado app.js e insertamos el siguiente código  'use strict'  var **express** = **require**('express');  var **bodyParser** = **require**('body-parser');  var app = **express**();  app.**use**(**bodyParser**.**urlencoded**({ extended: false }));  app.**use**(**bodyParser**.**json**());  *//configurar cabeceras http*  *// rutas base*  module.exports = app;  volvemos a index.js(ejemplo.js) y usamos el servicio express y el puerto como servicio web.  var **mongoose** = **require**('mongoose');  var app = **require**('./app');  var port = process.env.PORT || 3977;  **mongoose**.**connect**('mongodb://localhost:27017/proyecto', (err, res) => {      if (err) {          throw err;      } else {          console.**log**("conexion exitosa");          app.**listen**(port, function() {              console.**log**("Servidor de api rest de musica eschuando en http://localhost:" + port);          });      }  })  Vuelve a app.js y agrega el siguiente código para que se observe la respuesta del servidor web  app.**get**('/pruebas', function(req, res) {      res.**status**(200).**send**({ mesage: 'Bienvenido  al curso Ivan Azamar' });  });  Ahora, estaremos creando el Esquema, Clase o Estructura, en lo personal me gusta manejar el termino esquema, aunque en algunos manuales, libros o tutoriales usan clase o estructura, aquí haré la aclaración en clase y un poco de historia.  Entonces, creamos una carpeta llamada “modelo” de la filosofía mvc (modelo-vista-control) y creamos el esquema usuarios con el código que se muestra en la siguiente imagen    Usamos la función model de mongoose para exportar el Esquema cuando sea requerido en otros archivos js de nuestro proyecto (recuerde sus cursos sobre mvc)  Ahora, realizaremos los mismo para el Esquema Artistas de la siguiente forma    Ahora crearemos dos Estructuras adicionales, con la variante de que se tendrá que seguir el siguiente Diagrama de Clases    Como puedes observar existe, relación de tablas, es decir para poder crear una canción, necesita definir primero un album, y para definir un álbum, necesitas crear un artista. Existe relación de tablas con sus respectivos id, en Mongo podemos crear este tipos de relaciones sin ser una integridad referencial en el modelo relacional estricto.  Para lo anterior se necesita el siguiente código dentro del Esquema  Ejemplo:  artista: { type: **Schema**.ObjectId, ref: "Artista" }  Por lo que los dos esquemas siguientes quedarán como se muestran en las siguientes dos imágenes | |
| **Resultados** | |
| **Conclusiones:**  Realiza tus conclusiones una cuartilla completa | |
| **Referencias bibliográficas:**  Anota la referencias de tres libros que correspondan al tema | |
| **Anexos** | |

**FECHA DE ELABORACIÓN: 01-09-2021**