

CIP FP **cedh**este

Desarrollo Web en Entorno Cliente

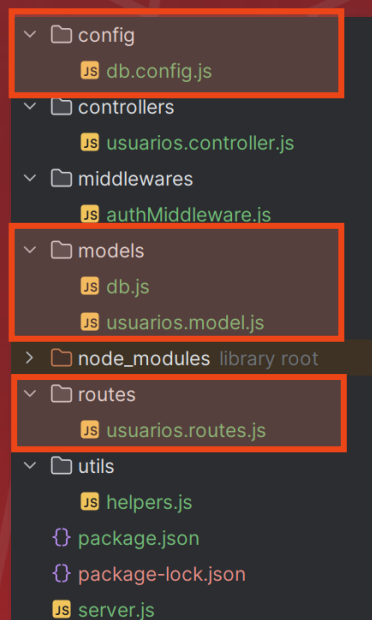
Javascript

API REST

NodeJS

En un principio utilizaremos 3 carpetas

- **config**: donde guardaremos los archivos con la información de acceso a la BBDD.
- **models**: donde estarán las funciones de acceso a la BBDD (queries).
- **routes**: donde encontraremos las rutas para invocar las diferentes funciones de acceso a la BBDD (models). ...



El resto de carpetas no serán utilizadas de momento:

- **controllers**: lógica de la aplicación.
- **middleware**: Funciones intermedias empleadas en el módulo express, como puede ser la autenticación de usuarios.
- **utils**: utilidades y funciones auxiliares.

NodeJS

server.js

```
const express = require('express');
const bodyParser = require('body-parser');
const mysql = require('mysql2');

const port = 3000;

//Declaramos la aplicación
const app = express()

require("./routes/usuarios.routes.js")(app);

// Configurar body-parser
app.use(bodyParser.json());

//Arrancamos el servidor
app.listen(port, () => {
  console.log(`Example app listening at http://localhost:${port}`)
})
```

NodeJS

db.config.js

```
module.exports = {  
  HOST: "localhost",  
  USER: "root",  
  PASSWORD: "",  
  DB: "testnode"  
};
```

NodeJS

usuarios.routes.js

```
module.exports = app => {  
  
  // Importamos el módulo donde se encuentran las Queries que nos devolverán los datos de la BBDD  
  const usuarios = require("../models/usuarios.model.js");  
  
  // Creamos las diferentes rutas y métodos para poder realizar las diferentes acciones.  
  /***** GET *****/  
  
  //Devuelve todos los usuarios  
  app.get("/usuarios", usuarios.buscarTodos);  
  
  //Devuelve un usuario por ID  
  app.get("/usuarios/:usuarioId", usuarios.buscarPorID);  
  
  /***** POST *****/  
  
  /***** PUT *****/  
  
  /***** DELETE *****/  
  
};
```

NodeJS

db.js

```
const mysql = require("mysql2");
const dbConfig = require("../config/db.config.js");

// Create a connection to the database
const connection = mysql.createConnection({
  host: dbConfig.HOST,
  user: dbConfig.USER,
  password: dbConfig.PASSWORD,
  database: dbConfig.DB
});

// open the MySQL connection
connection.connect(error => {
  if (error) throw error;
  console.log("Conectado correctamente a la bbdd: " + dbConfig.DB);
});

module.exports = connection;
```

NodeJS

usuarios.models.js

```
const sql = require("../db.js");

// Construimos una función llamada Usuario (a modo de objeto)
function Usuario(usuario) {
  this.email = usuario.email;
  this.name = usuario.name;
  this.password = usuario.password;
};

// Definimos las funciones que forman parte de Usuario, a las que accederemos desde la ruta correspondiente en función de la acción que queramos realizar: INSERT, SELECT, UPDATE o DELETE.

/***** GET *****/
Usuario.buscarTodos = (request, result) => {
  sql.query("SELECT * FROM usuarios", (err, res) => {
    console.log("Usuarios: ", res); // Información que mostramos en la consola donde estamos ejecutando NodeJS (Servidor)
    result.json(res); // Información que enviamos al cliente.
  });
};

// Donde en la función anterior teníamos una petición (request) sin datos de entrada, ahora tenemos un usuarioID definido en la ruta.
Usuario.buscarPorID = (request, result) => {
  sql.query(`SELECT * FROM usuarios WHERE id_usuario = ${request.params.usuarioId}`, (err, res) => {
    // Si la respuesta de la query devuelve una longitud de 1 o más valores, mostramos el usuario encontrado.
    console.log(res);
    if (res.length) {
      console.log("Usuario encontrado: ", res[0]); // Información que mostramos en la consola donde estamos ejecutando NodeJS (Servidor)
      result.json(res[0]); // Información que enviamos al cliente.
    } else result({ kind: "not_found" }, null); // No existe un usuario con ese ID
  });
};

/***** POST *****/
/***** PUT *****/
/***** DELETE *****/

module.exports = Usuario;
```