Node.js

Índice

Inicializar un proyecto

Para inicializar un projecto deberemos ejecutar el siguiente comando:

```
npm init
```

Este comando nos creará un archivo `package.json` con la información del proyecto, como el nombre, la versión, la descripción, etc.

Instalar dependencias

Para instalar dependencias deberemos ejecutar el siguiente comando:

```
npm install <nombre-de-la-dependencia>
```

Si corremos este comando sin el nombre de ninguna dependencia, nos instalará todas las dependencias que tengamos en el archivo `package.json` en la sección `dependencies`.

Carpeta `node_modules`

La carpeta `node_modules` es donde se instalan todas las dependencias de nuestro proyecto. Esta carpeta no se sube al repositorio, ya que es muy pesada y no es necesario que esté en el mismo. Para evitar esto, deberemos crear un archivo `.gitignore` en la raíz del proyecto y añadir la siguiente línea:

```
node_modules/
```

Instalar dependencias de desarrollo

Para instalar dependencias de desarrollo deberemos ejecutar el siguiente comando:

```
npm install <nombre-de-la-dependencia> --save-dev
```

Instalar dependencias de forma global

Para instalar dependencias de forma global deberemos ejecutar el siguiente comando:

```
npm install <nombre-de-la-dependencia> -g
```

Sección `scripts`

Esta sección del archivo `package.json` es donde podemos añadir comandos que queramos ejecutar desde la terminal. Por ejemplo, si queremos utilizar `nodemon` para que se reinicie el servidor cada vez que guardemos un archivo, podemos añadir el siguiente comando:

```
{
    "name": "node",
    "version": "1.0.0",
    "description": "",
    "main": "index.js",
    "scripts": {
        "dev": "npx nodemon index.js"
    },
    "author": "",
    "license": "ISC"
}
```

Y luego ejecutarlo con el siguiente comando:

```
npm run dev
```

Express

Para instalar Express deberemos ejecutar el siguiente comando:

```
npm install express
```

Crear un servidor

Para crear un servidor con Express deberemos crear un archivo `index.js` y añadir el siguiente código:

```
const express = require("express");
const app = express();

app.get("/", (req, res) => {
    res.send("Hello World!");
});

app.listen(3000, () => {
    console.log("Example app listening on port 3000!");
});
```

Y luego ejecutarlo con el siguiente comando:

```
node index.js
```

MySQL

Para instalar MySQL deberemos ejecutar el siguiente comando:

```
npm install mysql2
```

Conexión

Para conectarnos a la base de datos deberemos añadir el siguiente código:

```
const mysql = require("mysql2");

const connection = mysql.createConnection({
    host: "localhost",
    user: "usuario",
    password: "contraseña",
    database: "nombre_de_la_db",
});

connection.connect((err) => {
    if (err) {
        console.error("Error conectándose: " + err);
        return;
    }

    console.log("Base de datos conectada");
});
```

Consultas

Consultas sin parámetros

Para hacer consultas a la base de datos deberemos añadir el siguiente código:

```
connection.query("SELECT * FROM tabla", (err, rows) => {
    if (err) {
        console.error("Error consultando: " + err);
        return;
    }

    console.log(rows);
});
```

Consultas con parámetros

Para hacer consultas a la base de datos con parámetros deberemos añadir el siguiente código:

```
connection.query("SELECT * FROM tabla WHERE id = ?", [id], (err, rows) => {
    if (err) {
        console.error("Error consultando: " + err);
        return;
    }

    console.log(rows);
});
```

En este caso, incluimos en la consulta un `?` en cada lugar donde queramos incluir un parámetro. Luego, como segundo parámetro de la función `query` incluimos un array con los parámetros que queremos incluir en la consulta, en el mismo orden en el que queremos que aparezcan en la consulta.