



**Asignación—**

Investigación de Node.js

**Materia—**

Temas emergentes de aplicaciones web

**Maestro—**

Martin Guadalupe Bernal Lugo

**Alumno—**

Melanie Villarreal Madero - 199818

**Fecha—**

18 de enero 2024

En el entorno de Node.js, los "paquetes" se refieren a módulos o bibliotecas de código que se pueden utilizar para agregar funcionalidades específicas a una aplicación Node.js. Estos paquetes son distribuidos a través del sistema de gestión de paquetes de Node.js, conocido como npm (Node Package Manager).

- Express:

¿Qué es? Express es un marco de aplicación web para Node.js que simplifica el desarrollo de aplicaciones web y API. Proporciona una capa delgada sobre las características básicas de Node.js, facilitando la creación de servidores web y el manejo de rutas, middleware, y otras tareas comunes en el desarrollo web.

¿Para qué sirve? Express es utilizado para construir aplicaciones web y API de manera rápida y eficiente. Proporciona una estructura organizada para el código, facilita la gestión de rutas, middleware, y simplifica tareas comunes como el manejo de solicitudes y respuestas HTTP.

Cómo agregarlo a un proyecto:

- `npm install express`

Después de la instalación, puedes comenzar a usar Express en tu aplicación Node.js importándolo y configurándolo.

- Mongoose:

¿Qué es? Mongoose es una biblioteca de modelado de objetos para MongoDB y Node.js. Permite definir esquemas para los datos, proporciona una interfaz sencilla para interactuar con la base de datos MongoDB, y facilita la validación y manipulación de los datos.

¿Para qué sirve? Mongoose se utiliza para simplificar la interacción con bases de datos MongoDB en aplicaciones Node.js. Proporciona una capa de abstracción sobre las operaciones de la base de datos, facilita la definición de modelos y esquemas, y ofrece funcionalidades adicionales como la validación de datos y ganchos (hooks) para ejecutar código en eventos específicos.

Cómo agregarlo a un proyecto:

- `npm install mongoose`

Después de la instalación, puedes usar Mongoose en tu aplicación Node.js requiriéndolo y conectándose a tu instancia de MongoDB.

Para agregar cualquiera de estos paquetes a tu proyecto, simplemente ejecuta el comando `npm install` seguido del nombre del paquete en tu terminal, y luego podrás importarlo en tu código para comenzar a usarlo. Por ejemplo, para Express:

- `npm install express`

Y en tu código:

- javascript

```
const express = require('express');  
const app = express();  
// ... resto del código
```

Para Mongoose, el proceso es similar.

## Instalación y Uso de Express y Mongoose en Node.js

### Instalación de Express:

- Abrir la Terminal:
  - Abre la terminal en el directorio de tu proyecto Node.js.

```
C:\Users\melan>cd C:\Users\melan\OneDrive\Documents\ENE 2024\Emergentes web\asig01
```

- Ejecutar el Comando de Instalación:


```
C:\Users\melan\OneDrive\Documents\ENE 2024\Emergentes web\asig01>npm install express

added 62 packages, and audited 75 packages in 2s

11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 10.2.4 -> 10.3.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.3.0
npm notice Run `npm install -g npm@10.3.0` to update!
npm notice
C:\Users\melan\OneDrive\Documents\ENE 2024\Emergentes web\asig01>
```

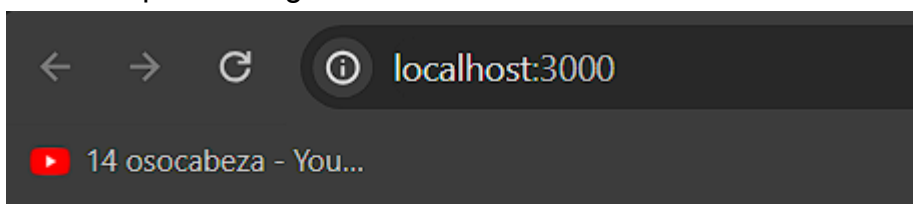
- Uso Básico en tu Código:



```
EXPLORER
ASIG01
  JS ejjs

JS ejjs > ...
1  const express = require('express');
2  const app = express();
3
4  app.get('/', (req, res) => {
5    | res.send('¡Hola, mundo!');
6  });
7
8  app.listen(3000, () => {
9    | console.log('Servidor escuchando en el puerto 3000');
10 });
11
```

Debería aparecer algo así:



¡Hola, mundo!

```
C:\Users\melan\OneDrive\Documents\ENE 2024\Emergentes web\asig01>node ej.js
Servidor escuchando en el puerto 3000
```

Instalación de Mongoose:

- Abrir la Terminal:

```
C:\Users\melan>cd C:\Users\melan\OneDrive\Documents\ENE 2024\Emergentes web\asig01
```

- Ejecutar el Comando de Instalación:

```
C:\Users\melan\OneDrive\Documents\ENE 2024\Emergentes web\asig01>npm install mongoose
added 20 packages, and audited 95 packages in 4s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\melan\OneDrive\Documents\ENE 2024\Emergentes web\asig01>
```

- Uso Básico en tu Código:

```
EXPLORER
└─ ASIG01
  └─ ej.js
  └─ ej2.js

JS ej.js > ...
1 const mongoose = require('mongoose');
2 mongoose.connect('mongodb://localhost/tu-base-de-datos', { useNewUrlParser: true, useUnifiedTopology: true });
3
4 // Define un esquema y un modelo
5 const Schema = mongoose.Schema;
6 const ejemploSchema = new Schema({
7   nombre: String,
8   edad: Number
9 });
10 const Ejemplo = mongoose.model('Ejemplo', ejemploSchema);
11
12 // Crea un nuevo documento y guárdalo en la base de datos
13 const nuevoEjemplo = new Ejemplo({ nombre: 'Juan', edad: 25 });
14 nuevoEjemplo.save((err, resultado) => {
15   if (err) return console.error(err);
16   console.log('Documento guardado:', resultado);
17 });
18
```

En este mini reporte, se demostró el proceso de instalación y uso básico de dos paquetes esenciales en el entorno de Node.js. Express se utilizó para la creación de una aplicación web simple, mientras que Mongoose se empleó para la conexión a MongoDB y la manipulación de datos. Estos paquetes proporcionan herramientas poderosas que facilitan el desarrollo de aplicaciones web robustas y la interacción con bases de datos NoSQL como MongoDB.