

## Trabajo Practico N°3

### Explica qué es npm y su función en un proyecto de Node.js.

**npm (Node Package Manager)** es el gestor de paquetes de Node.js. Su función principal es permitir la instalación, actualización y administración de bibliotecas o módulos externos en un proyecto de Node.js.

#### Funciones principales:

- Instalación de paquetes (`npm install nombredelpaquete`).
- Gestión de dependencias (`package.json`).
- Ejecución de scripts definidos en el proyecto.
- Publicación y distribución de paquetes en el registro de npm.

### Describe la estructura básica del archivo package.json y menciona tres campos importantes.

```
{
  "name": "mi-proyecto",
  "version": "1.0.0",
  "description": "Proyecto de Node.js",
  "main": "index.js",
  "scripts": {
    "start": "node index.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "dependencies": {
    "express": "^4.18.2"
  },
  "devDependencies": {
    "nodemon": "^2.0.22"
  }
}
```

#### \_ Tres campos importantes:

**dependencies:** Lista de paquetes necesarios para ejecutar la aplicación.

**devDependencies:** Lista de paquetes necesarios solo para el desarrollo (no en producción).

**scripts:** Comandos personalizados que pueden ejecutarse con `npm run`.

## ¿Cuál es la diferencia entre dependencias normales y dependencias de desarrollo en npm?

**dependencies:** Son paquetes esenciales para que la aplicación funcione en producción. Se instalan con:

```
npm install nombredelpaquete
```

**devDependencies:** Son paquetes necesarios solo para el entorno de desarrollo (por ejemplo, herramientas de testing o linters). Se instalan con:

```
npm install paquete --save-dev
```

### Ejemplo:

```
json
CopiarEditar
"dependencies": {
  "express": "^4.18.2"
},
"devDependencies": {
  "nodemon": "^2.0.22"
}
```

## ¿Qué comando utilizamos para actualizar un paquete instalado en nuestro proyecto?

Para actualizar un paquete específico:

```
npm update nombre-del-paquete
```

Para actualizar todos los paquetes:

```
npm update
```

Para actualizar un paquete a la última versión mayor disponible:

```
npm install nombre-del-paquete@latest
```

## ¿Para que se utiliza el comando npm link?

El comando `npm link` se usa para crear enlaces simbólicos entre paquetes locales y un proyecto en desarrollo.

Uso:

Desarrollo de un paquete de Node.js y prueba en otro proyecto sin publicarlo en npm.

Facilita el desarrollo de paquetes locales sin necesidad de reinstalarlos manualmente.

Ejemplo de uso:

Con una biblioteca en desarrollo como por ejemplo: `mi-paquete`, y queremos enlazarla en otro proyecto:

```
cd mi-paquete  
npm link
```

Luego, en el proyecto donde se desee usarlo:

```
cd mi-proyecto  
npm link mi-paquete
```

Esto permite importar `mi-paquete` en `mi-proyecto` sin necesidad de publicarlo.

## Ver el módulo de inicio de un proyecto

El módulo de inicio de un proyecto en Node.js está definido en el campo `"main"` de `package.json`.

Ejemplo en `package.json`:

```
"main": "index.js"
```

Esto significa que el archivo principal es `index.js` y para ejecutarlo, usamos:

```
node index.js
```