

TÉCNICO PROCESAMIENTO DE PRUEBAS DE SOFTWARE

INSTRUCTIVO API CRUD CON NODE JS



INSTRUCTIVO API CRUD CON NODE JS

A continuación, se relacionan los pasos para instalar Api CRUD con Node JS.

Crear un directorio.

`mkdir crud`

Ubicarse dentro del directorio

`cd crud`

Imagen 1. Crear directorio y acceder a él

```
C:\Users\Peter>mkdir crud  
C:\Users\Peter>cd crud
```

Iniciar en el directorio un proyecto JavaScript, para facilitar el proceso se dejan los datos por default dando enter a cada pregunta del script de parametrización del proyecto como se muestra en la siguiente imagen.



Imagen 2. Iniciar un proyecto JavaScript

```

C:\Users\Peter\crud>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (crud)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author: Peter
license: (ISC)
About to write to C:\Users\Peter\crud\package.json:

{
  "name": "crud",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Peter",
  "license": "ISC"
}

Is this OK? (yes) yes
```

A continuación, se puede instalar la librería express con el comando:

npm install express -save

donde -save guarda express como una dependencia en package.json. Luego con el comando:

npm install se podrán recuperar todas las dependencias.



INSTRUCTIVO API CRUD CON NODE JS

Imagen 3. Instalar Express

```
C:\Users\Peter\crud>npm install express --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN crud@1.0.0 No description
npm WARN crud@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 3.69s
found 0 vulnerabilities

C:\Users\Peter\crud>
```

Es momento de usar el IDE vs-code, instalado antes para lo cual se debe abrir entorno y abrir el directorio creado.

Imagen 4. Abrir el proyecto.

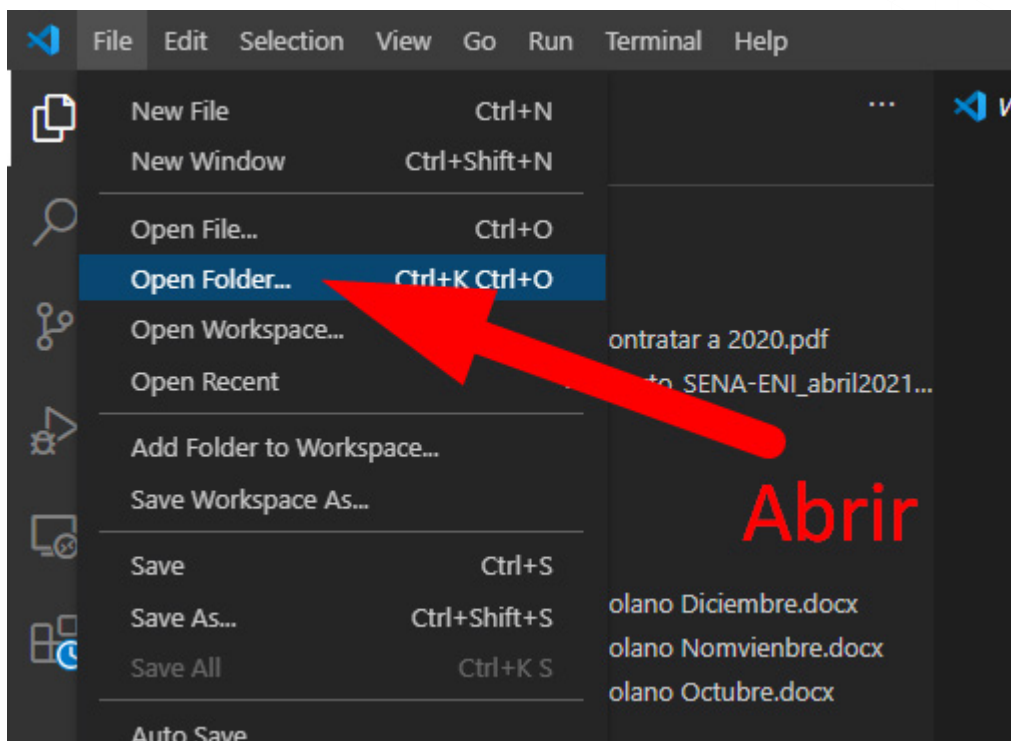
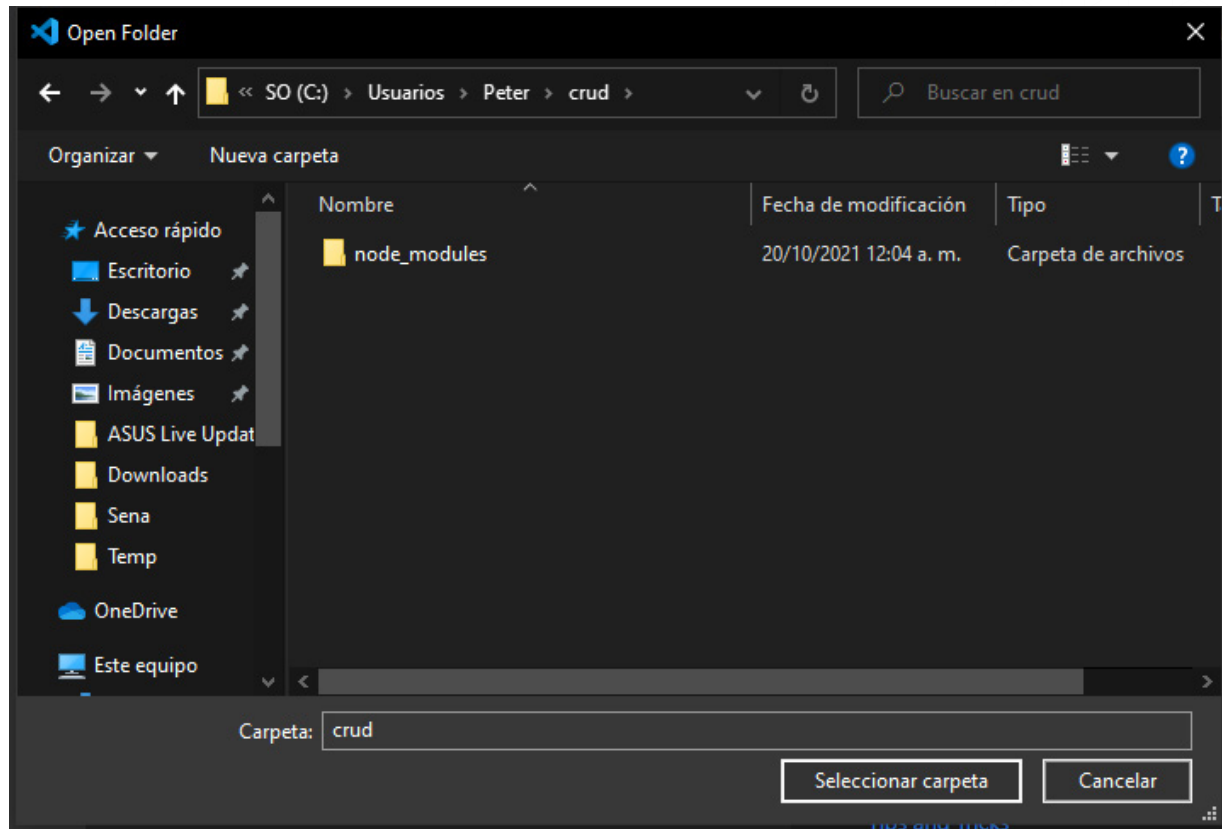




Imagen 5. Seleccionar directorio del proyecto



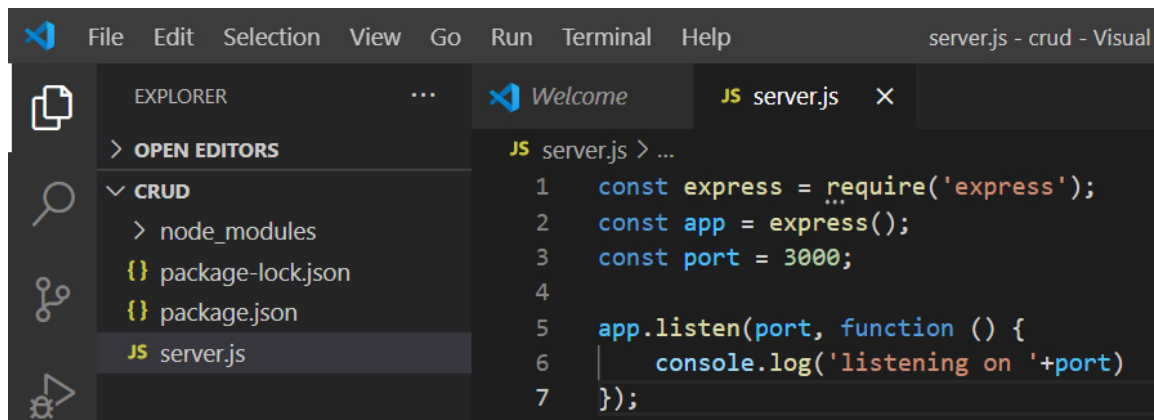
En nuestro archivo server.js vamos a agregar el siguiente código para iniciar nuestro servidor Express lo que vamos a hacer es importar los módulos necesarios para ejecutarlo.

```
const express = require('express');
const app = express();
const port = 3000;

app.listen(port, function () {
  console.log('listening on '+port)
});
```

INSTRUCTIVO API CRUD CON NODE JS

Imagen 6. Configurar servidor web

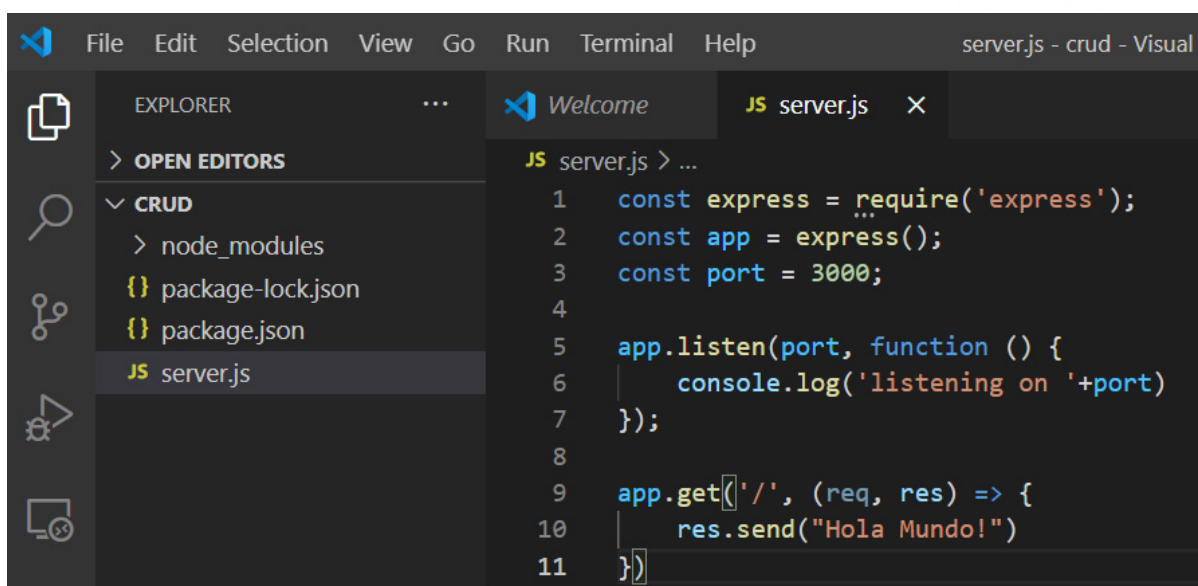


En el código anterior se agrega las librerías express (línea 1) y se inician los recursos que necesitara el servicio web (línea 2) y se define puerto que tendrá el servidor web (línea 3). Finalmente, se inicia el servicio web en el puerto definido (línea 5).

Se debe agregar el método GET para comprobar el funcionamiento de nuestro servidor.

```
app.get('/', (req, res) => {
  res.send("Hola Mundo!")
})
```

Imagen 7. Probar el servidor web.

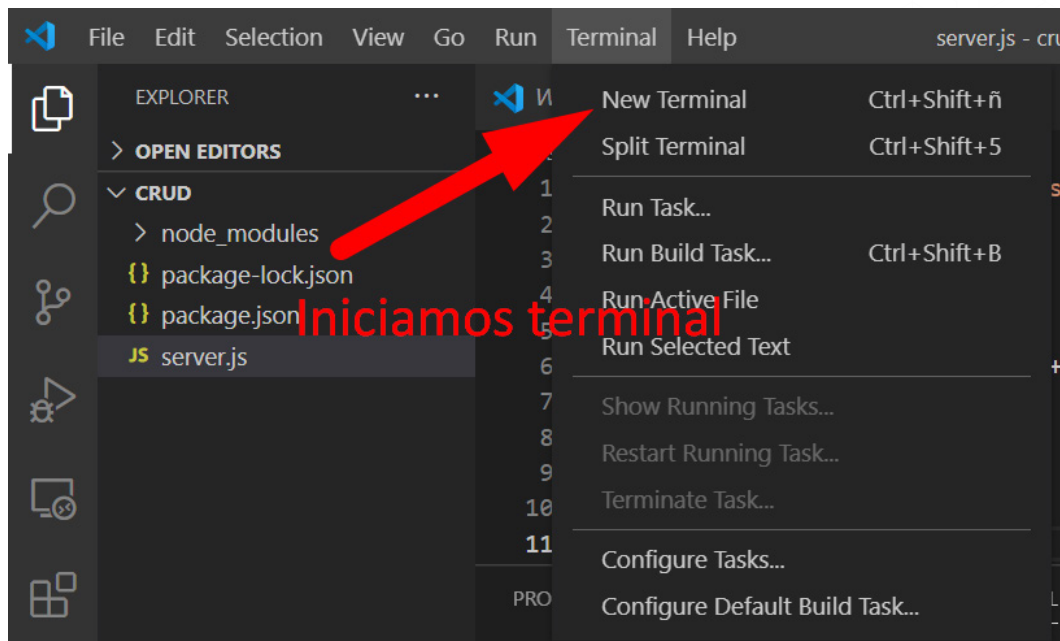




INSTRUCTIVO API CRUD CON NODE JS

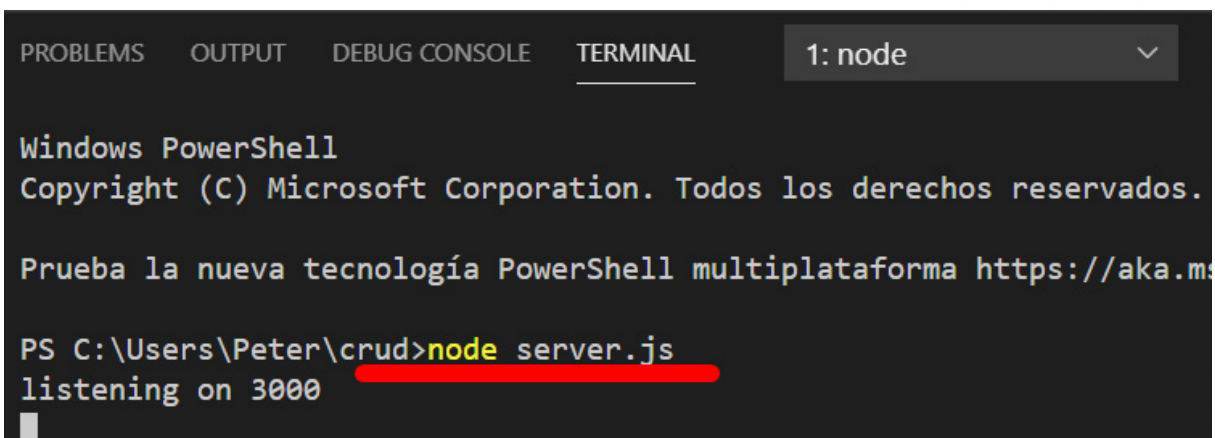
Es preciso abrir una terminal de consola para ejecutar o dar arranque al servidor web que se ha configurado.

Imagen 8. Abrir una terminal vs-code.



Una vez abierta la terminal, se procede a ejecutar el comando `node server.js`, para que arranque el servidor web.

Imagen 9. Iniciar servidor web





INSTRUCTIVO API CRUD CON NODE JS

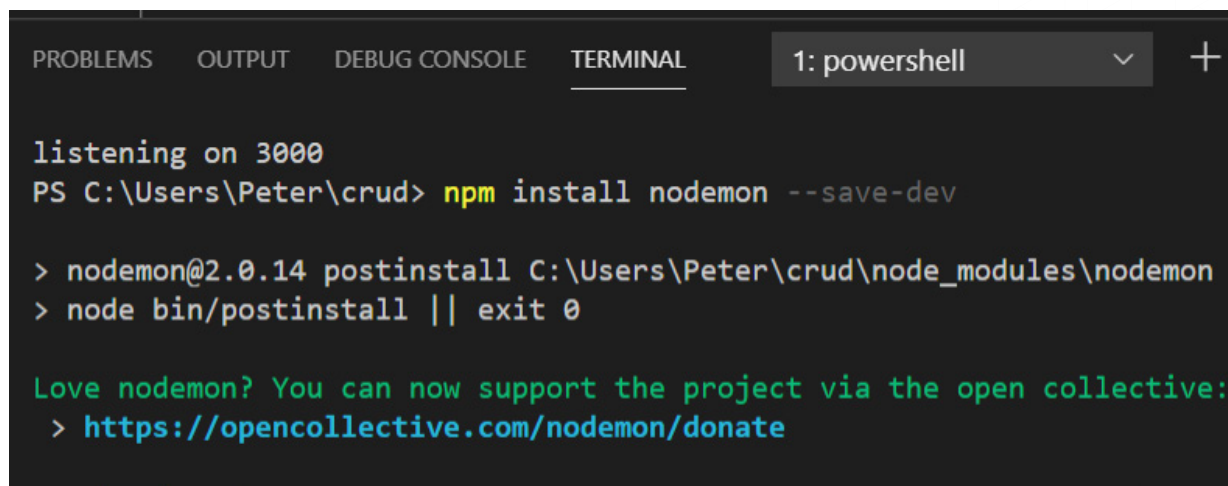


A continuación, probar en un navegador escribiendo la ruta `http://localhost:3000` para ver el mensaje de Hola Mundo ! y comprobar que el servidor está funcionando correctamente. Una vez realizada esta prueba se puede detener el servidor web con le comando de teclado Ctrl+D.

Para el desarrollo, se instalará **nodemon**, esta dependencia reinicia automáticamente el servidor web cada vez que exista un cambio en los archivos del proyecto. Para este caso, solo se necesitara **nodemon**, en el proceso de desarrollo por ello lo tanto se usará con el atributo bandera **--save-dev** en la instalación para que lo guarde como dependencia de desarrollo.

```
npm install nodemon --save-dev
```

Imagen 10. Instalar nodemon



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  1: powershell  +
listening on 3000
PS C:\Users\Peter\crud> npm install nodemon --save-dev

> nodemon@2.0.14 postinstall C:\Users\Peter\crud\node_modules\nodemon
> node bin/postinstall || exit 0

Love nodemon? You can now support the project via the open collective:
> https://opencollective.com/nodemon/donate
```

Una vez instalado, se agrega una clave de script en nuestro el archivo package.json para que nodemon sea ejecutado con un comando npm.

En la sección de scripts se deba gregar una línea lo siguiente:

```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "dev": "nodemon server.js"
},
```




INSTRUCTIVO API CRUD CON NODE JS

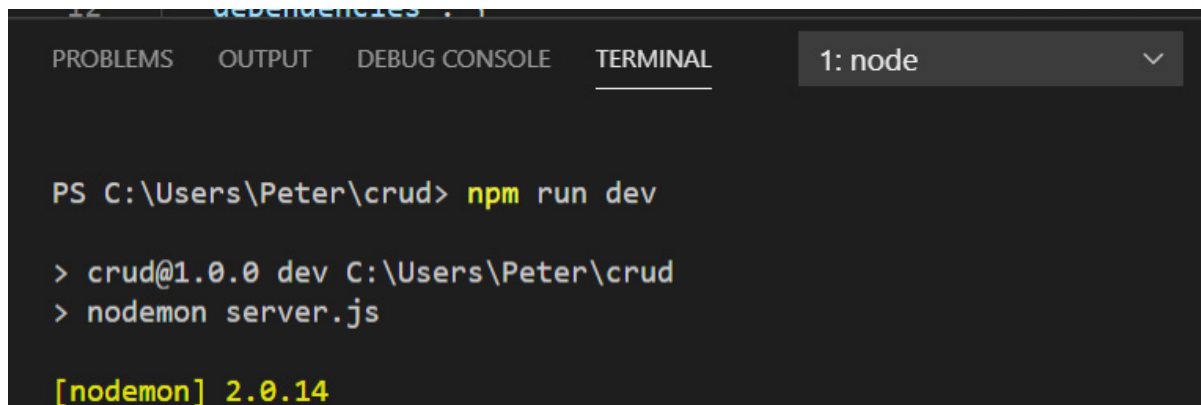
Imagen 11. Configurar nodemon

```
6   "scripts": {  
7     "test": "echo \"Error: no test specified\" && exit 1",  
8     "dev": "nodemon server.js"  
9   },
```

Ahora en nuestra consola se debe ejecutar el comando para llamar este script:

```
npm run dev
```

Imagen 12. Ejecutar nodemon



```
12  dependencies: . . .  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: node  
  
PS C:\Users\Peter\crud> npm run dev  
  
> crud@1.0.0 dev C:\Users\Peter\crud  
> nodemon server.js  
  
[nodemon] 2.0.14
```

Hasta este punto se ha logrado configurar las herramientas necesarias para implementar un conjunto de funciones o API con el intérprete de JavaScript NodeJS, ahora resta contar el a la base de datos e implantar las funciones CRUD que usara nuestra página web.