

# API REST - configuración

Guión

# Node

## Crear repositorio BitBucket para el Back-End

- Ir a <https://bitbucket.org/>
- Ir a la parte izquierda al botón (+) y elegir Repository
- Indicar Repository name (api)
- Marcar Access level como público → desmarcar la opción private
- Elegir en Version control system → Git

# Node

## Configurar Node

- Ejecutar en la Terminal de Ubuntu los siguientes comandos:
  - › **node --version** (está instalado en la máquina virtual la versión 6.0.0)
  - › **npm --version** (está instalado en la máquina virtual la versión 3.8.6)

## Crear proyecto Node

- Crear una carpeta en /home/alumno (por ejemplo practitioner) y posicionarnos en ella
- Ejecutar el comando **npm init** para que nos cree el fichero **package.json**
- Indicar los siguientes detalles: nombre del proyecto (apicol), versión (0.0.1), podemos dejar el resto de valores con los predeterminados
- Abrir Atom en esa carpeta ejecutando el comando **atom** .
- Chequear el fichero package.json

# BitBucket

## Vincular repositorio BitBucket

- En la Terminal ejecutar los siguientes comandos:
  - › `git remote add miapirepo https://<usuario>@bitbucket.org/<usuario>/api.git`
- Con esto habremos añadido un **alias** al repositorio de la API

# Node

## Instalar express

- Ejecutar en la Terminal el siguiente comando:
  - › **npm install express** → con esto se crea la carpeta node\_modules y dentro está express y los paquetes de los que depende
  - › **npm install express --save** → con esto se crea la carpeta node\_modules y dentro está express y los paquetes de los que depende. Además añade una entrada en **dependencies** del fichero package.json

## Instalar nodemon

- Ejecutar en la Terminal el siguiente comando:
  - › **npm install nodemon --save** → con esto se crea la carpeta node\_modules y dentro está express y los paquetes de los que depende. Además añade una entrada en **devDependencies** del fichero package.json

# Node

## Preparar la ejecución

- Crear un fichero llamado **server.js** en la raíz del proyecto
- Escribir la siguiente línea de código de ejemplo: `console.log("Hola desde API Node")`
- En la Terminal ejecutar el siguiente comando:
  - › **node server.js** → La aplicación se ejecuta y se cierra mostrando el mensaje "Hola desde API Node"
- En la sección **scripts** del `package.json` añadir la siguiente entrada:  
**"start": "nodemon server.js"**
- En la Terminal ejecutar el siguiente comando:
  - › **npm start** → La aplicación se ejecuta y se cierra mostrando el mensaje "Hola desde API Node". El script `start` es un alias de la ejecución completa (`node index.js`) y permite facilitar comandos con muchos parámetros.

# Node – Creación de la API

## Pasos a seguir

- En el fichero server.js añadir las siguientes líneas:  

```
var express = require('express')  
var app = express()  
app.get('/', function(req, res) {  
  res.send('Hola API')  
})
```

  - ← Con esto referenciamos a la biblioteca express
  - ← Creamos nuestra app que escuchará en un puerto
  - ← Con esto escuchamos peticiones GET a la raíz
- Al final del fichero añadimos lo siguiente:  

```
app.listen(3000)  
console.log("Escuchando en puerto 3000")
```

  - ← Nuestra app escuchará en el puerto 3000
  - ← Mensaje en la consola
- Ejecutamos la aplicación con **npm start**
- Accedemos con el navegador o con Postman a <http://localhost:3000>
- Debería de contestarnos con el mensaje **Hola API**