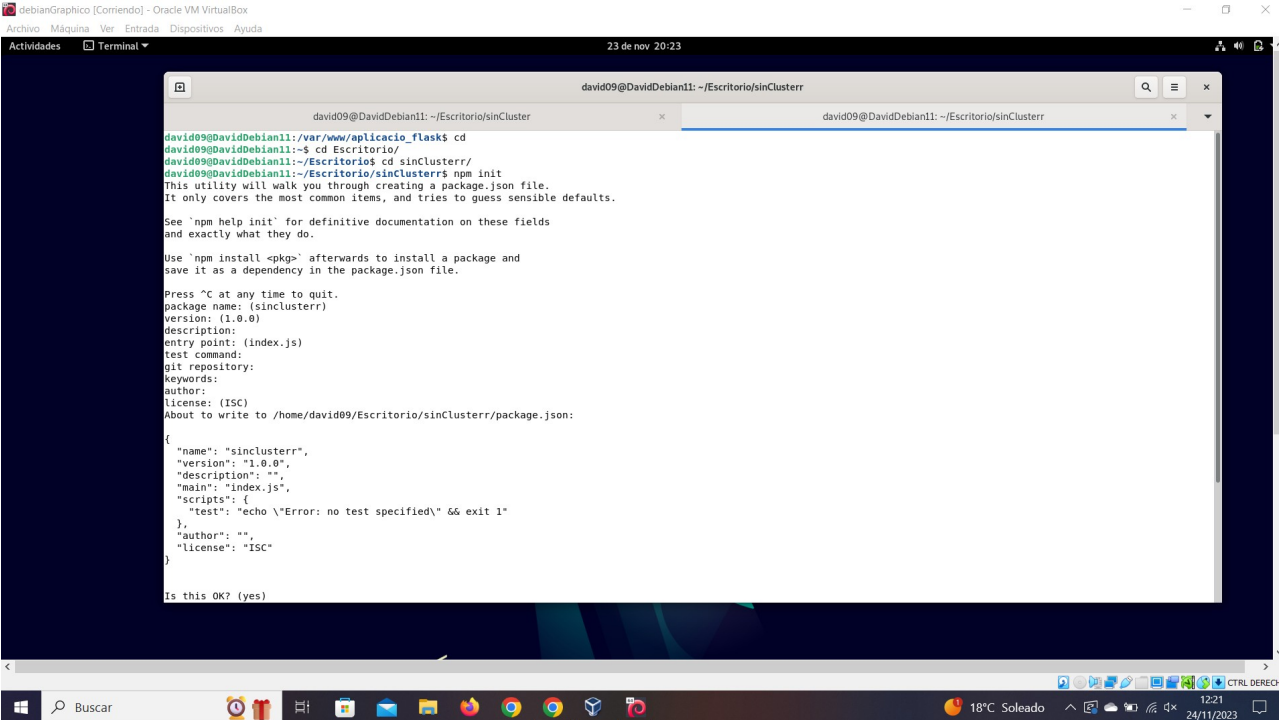


SinCluster

Primero creamos un directorio(en mi caso sera en el escritorio) y lo llamaremos sinClusterr usando el comando `mkdir Escritorio/sinClusterr`. Accedemos dentro del directorio.

Utilizamos el comando en consola `npm init` que nos creara todos los archivos necesarios



```
debiangrphico [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 23 de nov 20:23

david09@DavidDebian11: ~/Escritorio/sinCluster
david09@DavidDebian11:~/Escritorio/sinCluster$ cd /var/www/aplicacio_flask$ cd
david09@DavidDebian11:~/Escritorio$ cd Escritorio/
david09@DavidDebian11:~/Escritorio$ cd sinClusterr/
david09@DavidDebian11:~/Escritorio/sinClusterr$ 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: (sinclusterr)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/david09/Escritorio/sinClusterr/package.json:

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

Is this OK? (yes)
```

Después tenemos que meterle express al para que funcione el proyecto usando el comando `npm install express`.

```
debianGraphico [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 23 de nov 20:24

david09@davidDebian11: ~/Escritorio/sinClusterr

Is this OK? (yes)
david09@davidDebian11:~/Escritorio/sinClusterr$ ls
package.json
david09@davidDebian11:~/Escritorio/sinClusterr$ npm install express
added 62 packages, and audited 63 packages in 3s
11 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
david09@davidDebian11:~/Escritorio/sinClusterr$ ls
node_modules package.json package-lock.json
david09@davidDebian11:~/Escritorio/sinClusterr$
```

como ves nos ha instalado 1 carpeta que es node mudeles y 2 archivos json.

Ahora vamos hacer un archivo js que sera el archivo que a traves de node vamos a lanzar.

Podle el nombre que quieras (sinCluster.js por ejemplo).

Dentro de ese archivo tiene que contener lo siguiente;

```
debianGraphico [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 23 de nov 20:33

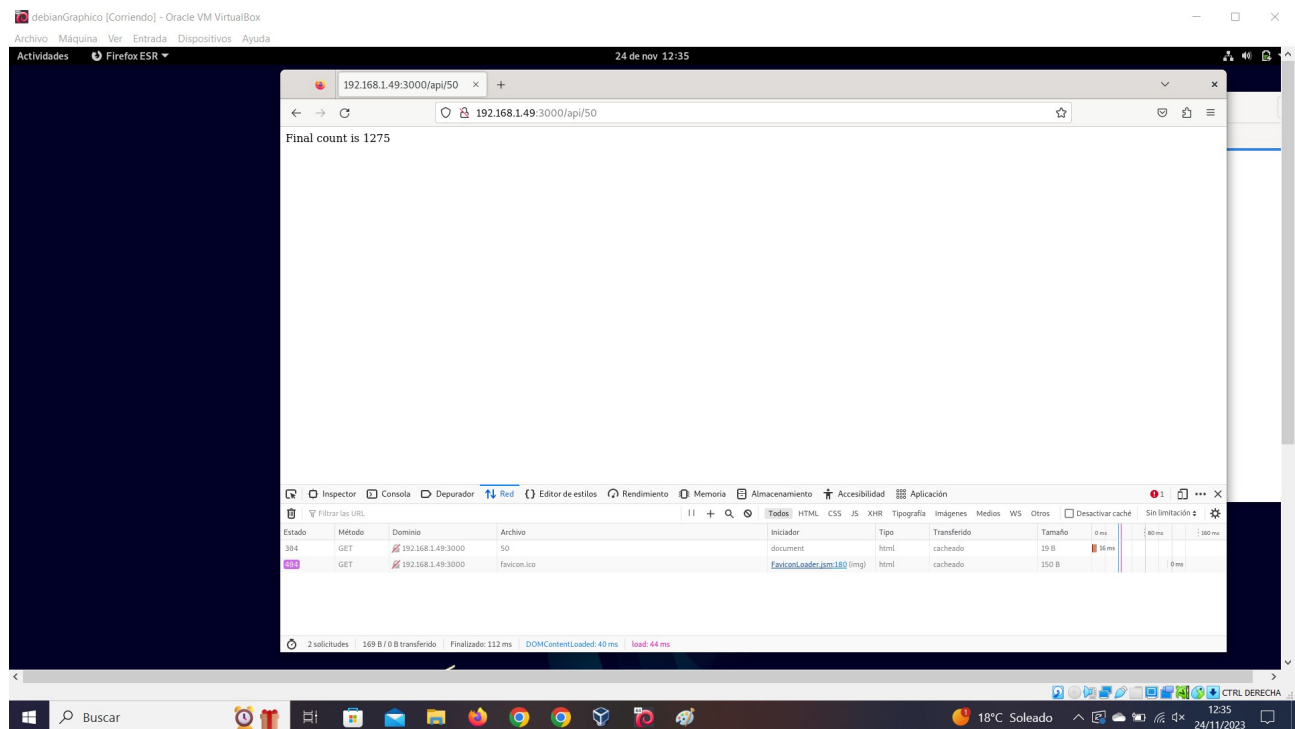
david09@davidDebian11: ~/Escritorio/sinClusterr

david09@davidDebian11:~/Escritorio/sinClusterr$ ls
node_modules package.json package-lock.json
david09@davidDebian11:~/Escritorio/sinClusterr$ sudo nano sinCluster.js
[sudo] password for david09:
david09@davidDebian11:~/Escritorio/sinClusterr$ cat sinCluster.js
const express = require("express");
const app = express();
const port = 3000;
app.get("/", (req,res) => {
  res.send("Hello world!");
});
app.get("/api/:n", function(req,res) {
  let n = parseInt(req.params.n);
  let count = 0;
  if( n > 5000000000) n = 5000000000;
  for(let i = 0; i <= n;i++){
    count += i;
  }
  res.send('Final count is ${count}');
});
app.listen(port,()=> {
  console.log('App listening on port ${port}');
});
david09@davidDebian11:~/Escritorio/sinClusterr$
```

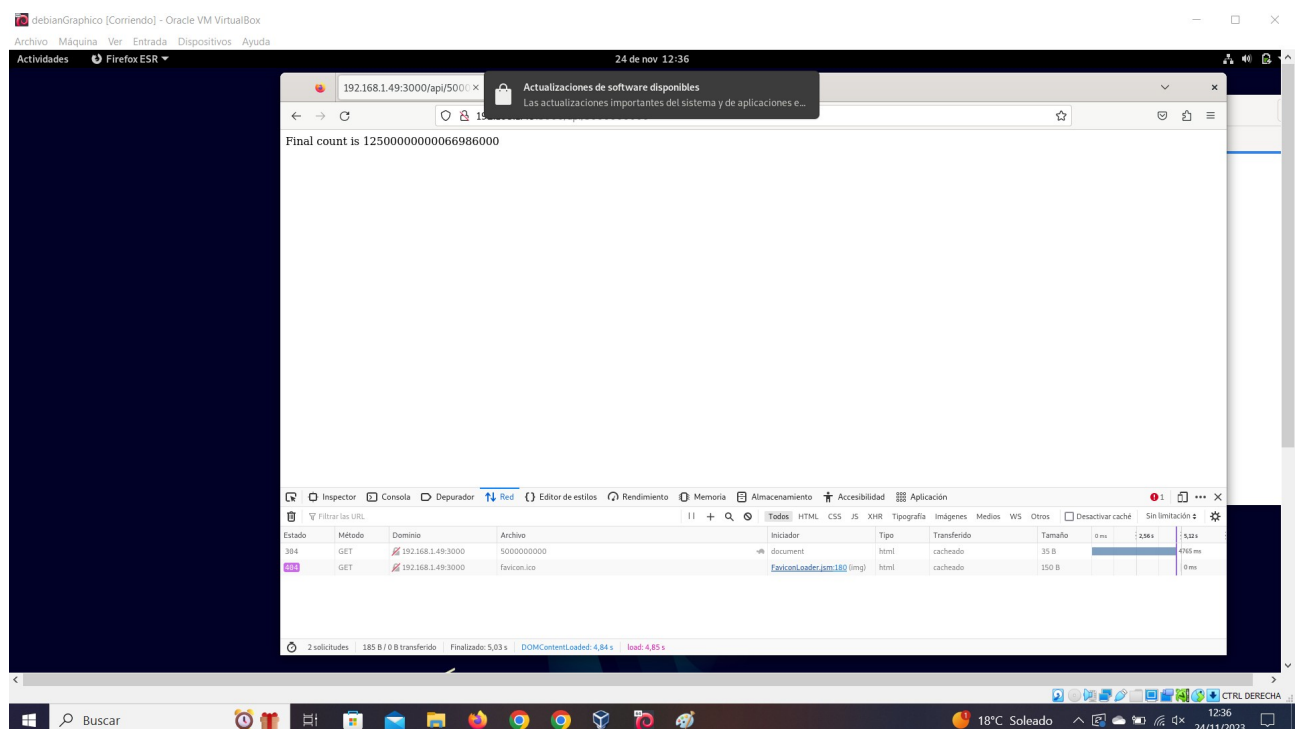
tiene que usa la librería express, el puerto que tiene que escuchar(en este caso 3000) si entra por ejemplo en localhost:3000 saldra hello world.

Si ponemos localhost:3000/api/(un numero entre 1 y 5000000000) hara un count de lo que tardara.

Si lo hacemos por separado vemos que poniendo un numero bajo(50) lo hace enseguida.

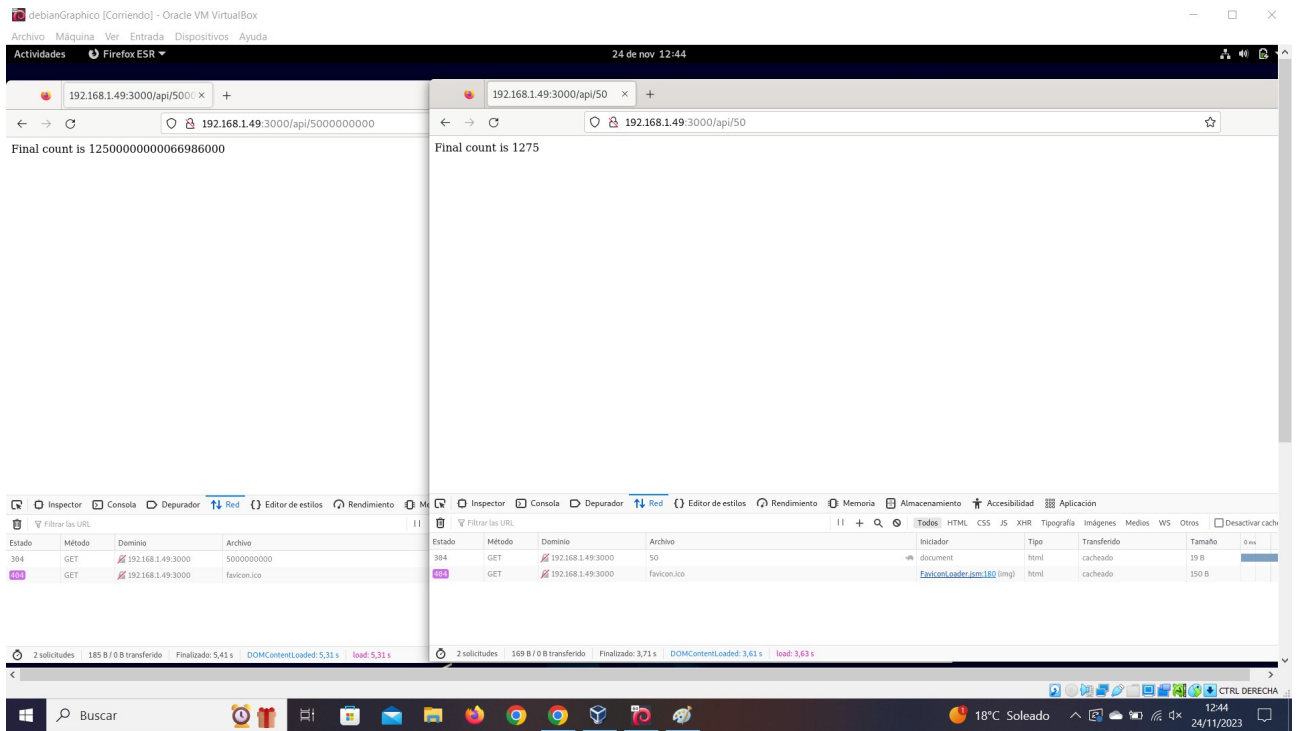


Vemos como tarda 44 ms en hacerlo y en cambio si ponemos 5000000000 tardara mas en hacerlo.



Como vemos en la siguiente imagen tarda 4,65 segundos en hacerlo.

Ahora vamos hacer los 2 a la vez y vemos que uno tarda mucho mas que el otro.



El de 5000000000 tarda 5,31s y el de 50 3,63s y esto se debe a que sin cluster al usar solo 1 CPU hasta que un proceso no termina no empieza la otra.