

# Creación de un nuevo proyecto TypeScript

Ismael García Vela 2ºASIR



- Primero instalamos **nodejs**.

```
zsh — Konsole
> sudo pacman -S nodejs
resolving dependencies...
looking for conflicting packages...

Packages (1) nodejs-16.9.1-1

Total Installed Size: 30,85 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
   nodejs-16.9.1-1-x86_64 is up to date
(1/1) checking keys in keyring [#####] 100%
(1/1) checking package integrity [#####] 100%
(1/1) loading package files [#####] 100%
(1/1) checking for file conflicts [#####] 100%
(1/1) checking available disk space [#####] 100%
:: Processing package changes...
(1/1) installing nodejs [#####] 100%
Optional dependencies for nodejs
  npm: nodejs package manager
:: Running post-transaction hooks...
(1/1) Arming ConditionNeedsUpdate...
```

- Comprobamos que esté instalado y que funcione correctamente.

```
zsh — Konsole
> node
Welcome to Node.js v16.9.1.
Type ".help" for more information.
> .help
.break      Sometimes you get stuck, this gets you out
.clear      Alias for .break
.editor     Enter editor mode
.exit       Exit the REPL
.help       Print this help message
.load       Load JS from a file into the REPL session
.save       Save all evaluated commands in this REPL session to a file

Press Ctrl+C to abort current expression, Ctrl+D to exit the REPL
>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
>
```

- Hacemos lo mismo con **npm**

```

~: zsh — Konsole
> sudo pacman -S npm
resolving dependencies...
looking for conflicting packages...

Packages (4) node-gyp-8.1.0-2  nodejs-nopt-5.0.0-2  semver-7.3.5-2  npm-7.23.0-1

Total Installed Size: 13,38 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
  npm-7.23.0-1-any is up to date
  node-gyp-8.1.0-2-any is up to date
  semver-7.3.5-2-any is up to date
  nodejs-nopt-5.0.0-2-any is up to date
(4/4) checking keys in keyring [#####] 100%
(4/4) checking package integrity [#####] 100%
(4/4) loading package files [#####] 100%
(4/4) checking for file conflicts [#####] 100%
(4/4) checking available disk space [#####] 100%
:: Processing package changes...
(1/4) installing nodejs-nopt [#####] 100%
(2/4) installing semver [#####] 100%
(3/4) installing node-gyp [#####] 100%
(4/4) installing npm [#####] 100%
:: Running post-transaction hooks...
(1/1) Arming ConditionNeedsUpdate...

```

- Y también comprobamos su funcionamiento.

```

~: zsh — Konsole
> npm
npm <command>

Usage:

npm install          install all the dependencies in your project
npm install <foo>    add the <foo> dependency to your project
npm test            run this project's tests
npm run <foo>       run the script named <foo>
npm <command> -h     quick help on <command>
npm -l              display usage info for all commands
npm help <term>     search for help on <term>
npm help npm        more involved overview

All commands:

access, adduser, audit, bin, bugs, cache, ci, completion,
config, dedupe, deprecate, diff, dist-tag, docs, doctor,
edit, exec, explain, explore, find-dupes, fund, get, help,
hook, init, install, install-ci-test, install-test, link,
ll, login, logout, ls, org, outdated, owner, pack, ping,
pkg, prefix, profile, prune, publish, rebuild, repo,
restart, root, run-script, search, set, set-script,
shrinkwrap, star, stars, start, stop, team, test, token,
uninstall, unpublish, unstar, update, version, view, whoami

Specify configs in the ini-formatted file:
  /home/ismael/.npmrc
or on the command line via: npm <command> --key=value

More configuration info: npm help config
Configuration fields: npm help 7 config

npm@7.23.0 /usr/lib/node_modules/npm

```

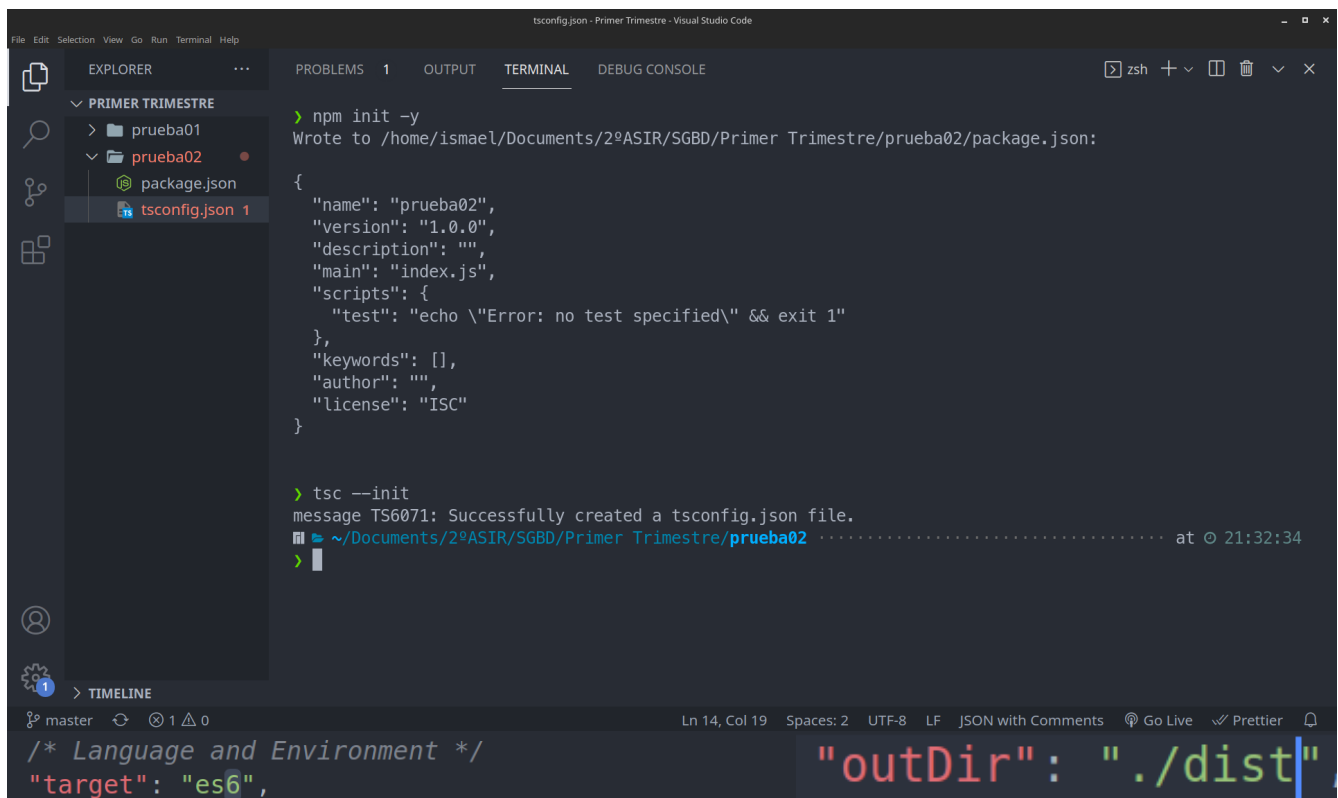
- Instalamos con **npm** (node package manager) typescript a nivel global

```
zsh — Konsole
> sudo npm install -g typescript
[sudo] password for ismael:

changed 1 package, and audited 2 packages in 3s

found 0 vulnerabilities
took 4s at 21:17:09
>
```

- Se crea la carpeta e inicia la carpeta del proyecto node (genera el archivo **package.json**) y typescript (genera el archivo **tsconfig.json**)
- Luego vamos al tsconfig y cambiamos el “**target**” a “**es6**”, también añadimos de directorio de salida **./dist**



The screenshot shows the Visual Studio Code interface with the Explorer, Problems, Output, and Terminal panels. The Explorer panel shows a project structure with folders 'prueba01' and 'prueba02'. The 'prueba02' folder contains 'package.json' and 'tsconfig.json'. The Terminal panel shows the following commands and output:

```
zsh — Visual Studio Code
> npm init -y
Wrote to /home/ismael/Documents/2ºASIR/SGBD/Primer Trimestre/prueba02/package.json:

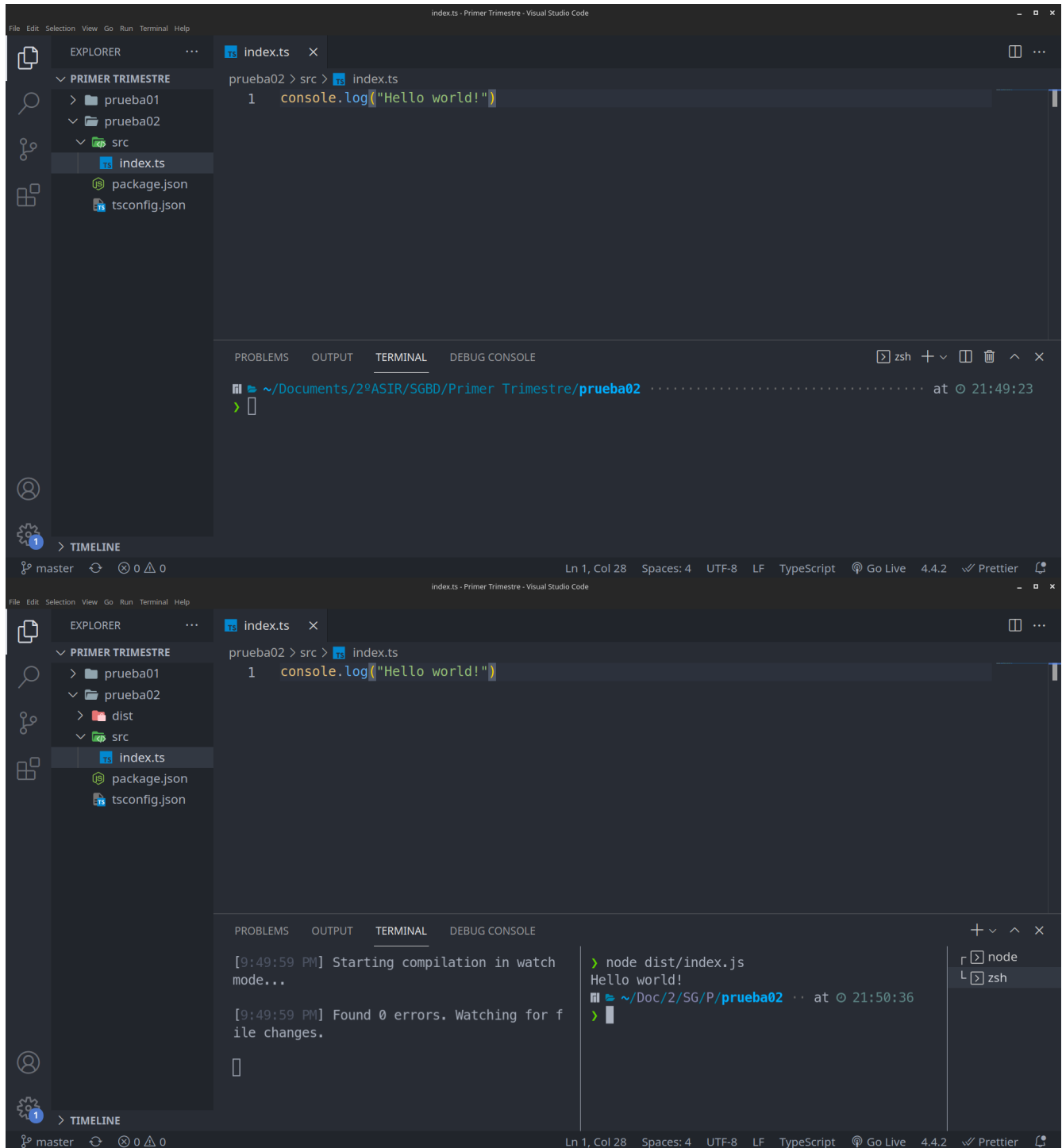
{
  "name": "prueba02",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

> tsc --init
message TS6071: Successfully created a tsconfig.json file.
~/Documents/2ºASIR/SGBD/Primer Trimestre/prueba02 at 21:32:34
>
```

The bottom of the image shows the 'tsconfig.json' file being edited. The 'target' is set to 'es6' and 'outDir' is set to './dist'.

```
/* Language and Environment */
"target": "es6",
"outDir": "./dist",
```

- Creamos una carpeta `src` y un primer archivo en su interior.
- Abrimos dos terminales, uno para llamar al compilador si detecta cambios con `tsc -w`, otro para llamar al ejecutable JS que se encontrará en la carpeta `dist` con `node dist/index.js`



- Así es cómo quedaría el directorio del proyecto al terminar. Con ambas carpetas, src y dist, y con sus respectivos archivos dentro. Por último los dos archivos de node y tsc.

