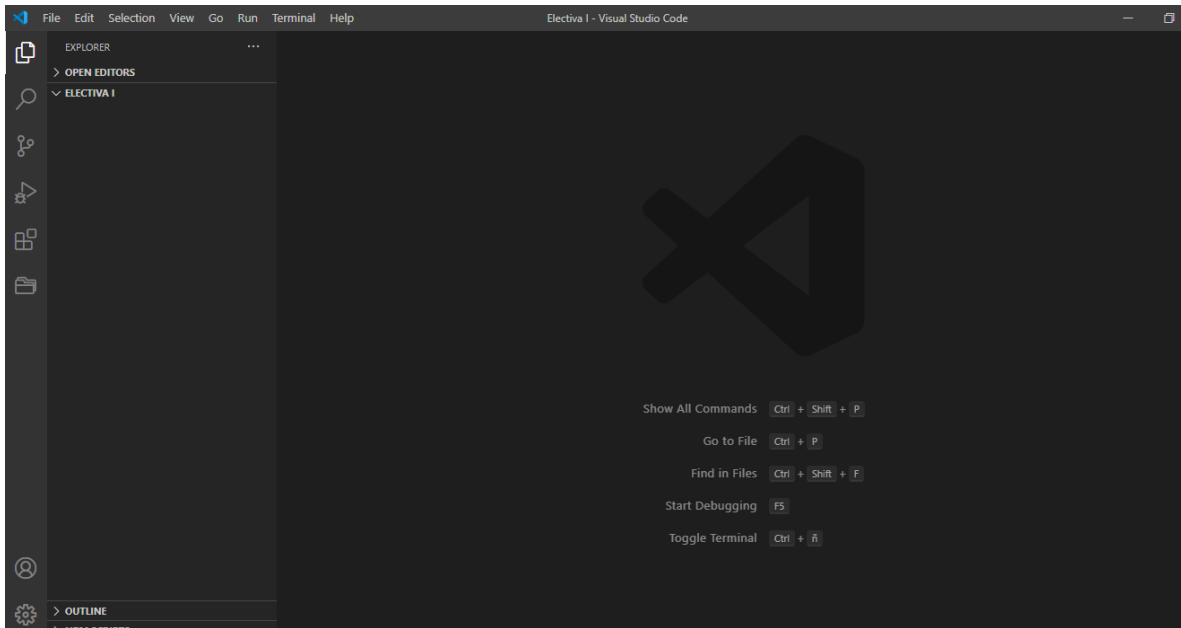
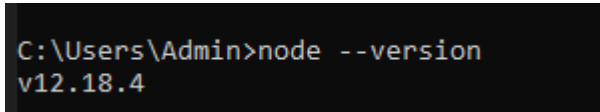


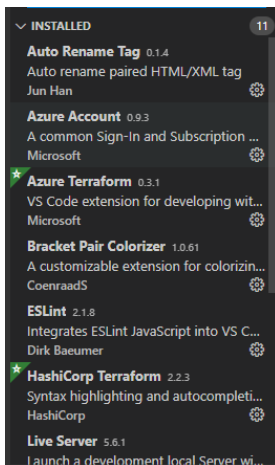
1. Visual Studio Code (o Atom si les parece, <https://code.visualstudio.com>)



2. Tener Instalado NodeJS (Version LTS <https://nodejs.org/es/>)



3. Opcional tener instalado los distintos plug ins



4. Crear una carpeta (ustedes deciden donde crean la carpeta, el único consejo es que sea de fácil recordación, que sea una ruta de máximo 256 caracteres).



5. En dicha carpeta abierta desde Visual Studio Code debe hacer lo siguiente:

A screenshot of the Visual Studio Code interface. The 'TERMINAL' tab is active, showing a bash shell. The prompt is 'PE-HO-262+Admin@PE-HO-262 MINGW64 ~/Desktop/Electiva I'. The command '\$ node --version' has been entered, and the output 'v12.18.4' is displayed below it. The terminal window has a dark theme and standard VS Code window controls at the top.

6. Después escribimos lo siguiente: npm init

```
$ 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.
```

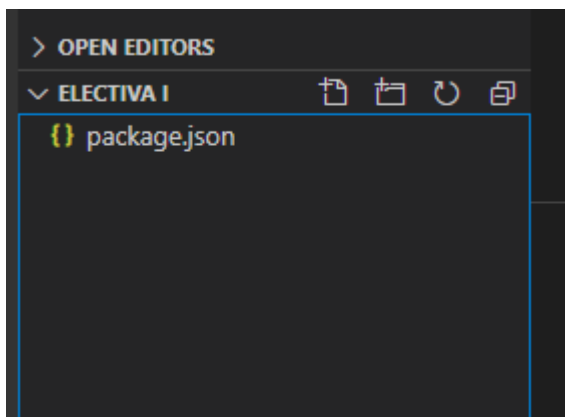
7. Nos aparecerá luego nuestro primer referencia con NPM lo siguiente

```
About to write to C:\Users\Admin\Desktop\Electiva 1\package.json:

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

Is this OK? (yes) yes
```

8. Se nos ha creado un archivo del tipo: package.json



9. Entramos en el tipo package.json que se nos despliega

```
{ } package.json ×
{ } package.json > ...
1  {
2    "name": "electiva-i",
3    "version": "1.0.0",
4    "description": "Electiva",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "Camilo",
10   "license": "ISC"
11 }
12
```

10. Remplazar en este punto donde aparece

```
    "scripts": {
      "start": "electron ."
    },
```

11. Luego después de esto darle install a electron

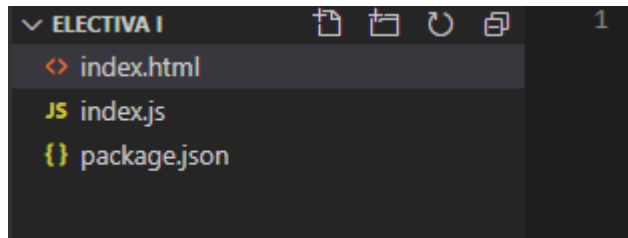
```
PE-HO-262+Admin@PE-HO-262 MINGW64 ~/Desktop/Electiva I
$ npm install --g electron
Debugger attached.
[.....] \ fetchMetadata: sill resolveWithNewModule lodash@4.17.20 checking installable status

Waiting for the debugger to disconnect...

> electron@10.1.2 postinstall C:\Users\Admin\AppData\Roaming\npm\node_modules\electron
> node install.js

Debugger attached.
Waiting for the debugger to disconnect...
+ electron@10.1.2
added 87 packages from 98 contributors in 53.399s
Waiting for the debugger to disconnect...
```

12. Luego de esto podríamos hacer lo siguiente: crear dos tipos de archivos, el primero de tipo javascript y el otro podríamos crear el código en html



13. En el archivo definido como index.js copiamos y pegamos lo siguiente, entendiendo como cada línea de código nos da el ejemplo

```
JS index.js > ...
1  const { app, BrowserWindow } = require('electron')
2
3  function createWindow () {
4    // Crea la ventana del navegador.
5    const win = new BrowserWindow({
6      width: 800,
7      height: 600,
8      webPreferences: {
9        nodeIntegration: true
10     }
11   })
12
13   // y carga el index.html de la aplicación.
14   win.loadFile('index.html')
15
16   const { app, BrowserWindow } = require('electron')
17
18   function createWindow () {
19     // Crea la ventana del navegador.
20     const win = new BrowserWindow({
21       width: 800,
22       height: 600,
23       webPreferences: {
24         nodeIntegration: true
25       }
26     })
27
```

14. Luego de esto, debemos copiar Y pegar conociendo como sería el archivo que necesitamos hacer.

```
index.html > html > body > h1
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>Hello World!</title>
6      <!-- https://electronjs.org/docs/tutorial/security#csp-meta-tag -->
7      <meta http-equiv="Content-Security-Policy" content="script-src 'self' 'unsafe-inline' 'unsafe-eval'">
8    </head>
9    <body>
10     <h1>Camilo</h1>
11     We are using node <script>document.write(process.versions.node)</script>,
12     Chrome <script>document.write(process.versions.chrome)</script>,
13     and Electron <script>document.write(process.versions.electron)</script>.
14   </body>
15 </html>
16
```

15. Luego de esto, debemos ver el control html

```
index.html > html > body > h1
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>Hello World!</title>
6      <!-- https://electronjs.org/docs/tutorial/security#csp-meta-tag -->
7      <meta http-equiv="Content-Security-Policy" content="script-src 'self' 'unsafe-inline' 'unsafe-eval'">
8    </head>
9    <body>
10     <h1>Camilo</h1>
11     We are using node <script>document.write(process.versions.node)</script>,
12     Chrome <script>document.write(process.versions.chrome)</script>,
13     and Electron <script>document.write(process.versions.electron)</script>.
14   </body>
15 </html>
16
```

16. Después de esto debemos darle a la aplicación: npm start

