



Certified Tech Developer

The Ultimate Degree

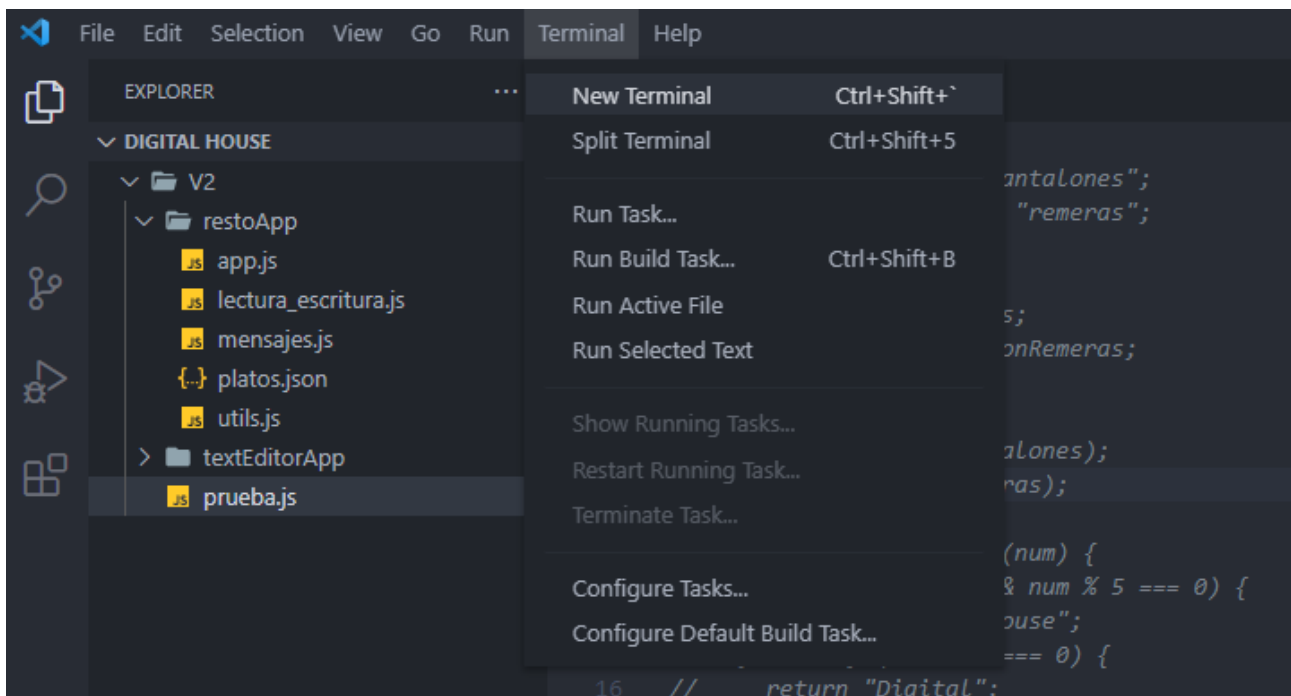
Troubleshooting de Node y VScode

Vamos a realizar una serie de pasos y pruebas para verificar que todo funcione correctamente, o solucionarlo en caso de que no lo haga.

1. Asegurarnos de que estén las herramientas correctamente instaladas.

Abrir una terminal en nuestro VScode:

- a. En la parte superior izquierda del programa, hacer clic en "Terminal"
- b. Luego clic en "New Terminal" o "Nueva Terminal" dependiendo el idioma.



- c. En la Consola o Terminal que se abrió escribir el siguiente comando:
node -v o **node --version** (son exactamente el mismo comando, verificar



la versión de Node que tengamos, uno es versión larga y otro version corta, pero hacen lo mismo).

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\leand\Desktop\Digital House\V2> node -v
v14.17.1
PS C:\Users\leand\Desktop\Digital House\V2> |
```

2. Una vez asegurados que tenemos una versión de Node instalada (no tiene por que ser la misma que la imagen, puede ser una posterior, o incluso anterior), asegurarnos de que nuestro archivo esté correctamente guardado.
 - a. Si vemos el siguiente círculo blanco, significa que no hemos guardado nuestro archivo. Lo guardamos entonces con *ctrl* + *s*

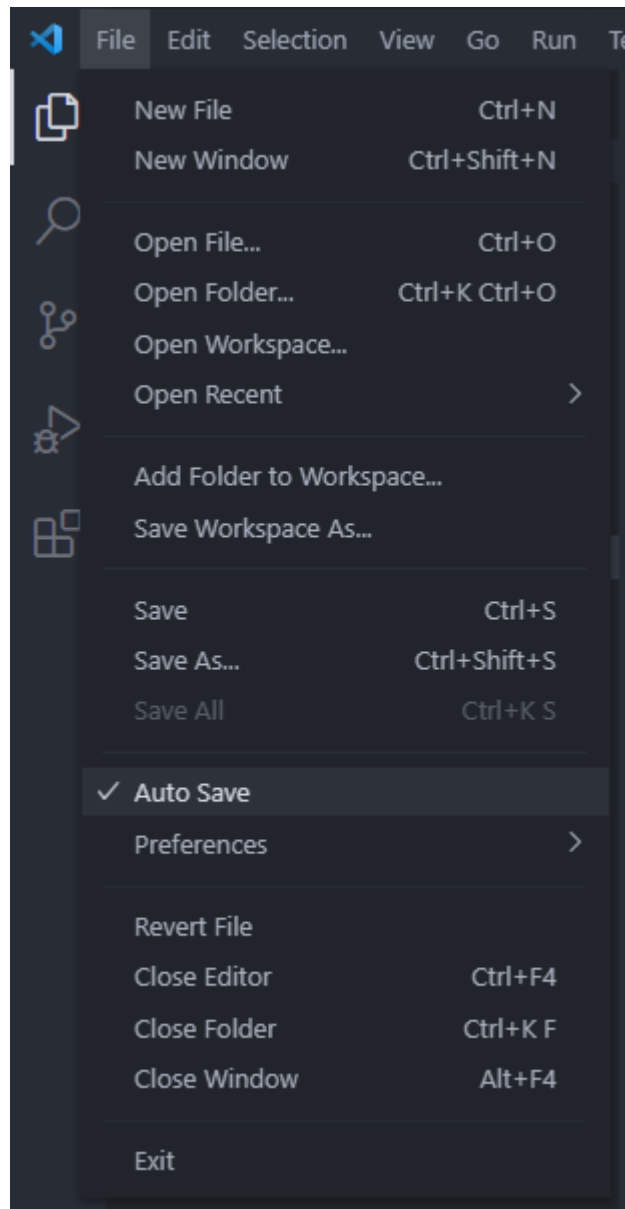
```
.js prueba.js  ●  .js utils.js

V2 > .js prueba.js > ...
1  let cajonRemeras = "pantalones";
2  let cajonPantalones = "remeras";
3  let cama;
4  |
```

- b. Lo recomendable siempre es activar la función "Auto Save" o "Guardar Automáticamente" la cual se activa haciendo clic en "File" o "Archivo" en



la parte superior izquierda de VScode.



3. Asegurémonos de estar correctamente “parados” en la carpeta donde tenemos guardado nuestro archivo de trabajo.



- a. Con la terminal abierta veremos un mensaje al comenzar, que nos indica dónde estamos actualmente.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  
PS C:\Users\leand\Desktop\Digital House\V2> |
```

- b. Después de esto escribimos el siguiente comando que nos mostrará todos los archivos y carpetas que contenga la carpeta en la que estamos parados (en este ejemplo la carpeta en cuestión es la V2).

Comando: **ls**

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  
PS C:\Users\leand\Desktop\Digital House\V2> ls  
  
Directorio: C:\Users\leand\Desktop\Digital House\V2  
  
Mode                LastWriteTime         Length Name  
----                -  
d-----           27/06/2021 12:14 a. m.         restoApp  
d-----           27/06/2021 12:14 a. m.        textEditorApp  
-a----           02/07/2021 12:40 p. m.         3941 prueba.js  
  
PS C:\Users\leand\Desktop\Digital House\V2> |
```

- c. Como vemos, la carpeta V2 contiene dos carpetas —cuando son carpetas no tienen ninguna terminación, ej: .txt, .js, .html, etc.— y un archivo **prueba.js**. Nosotros queremos acceder a la carpeta **textEditorApp**, por lo tanto, en la consola escribiremos el comando **cd** seguido del nombre de la carpeta **exactamente como está en la imagen** —recordá que la programación, y la computación en general, son *case sensitive*, por lo que



se distinguen las mayúsculas de las minúsculas—.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\leand\Desktop\Digital House\V2> cd textEditorApp
PS C:\Users\leand\Desktop\Digital House\V2\textEditorApp> |
```

- d. Luego de presionar Enter vemos que nuestra línea principal de texto agregó “\textEditorApp” a lo que ya tenía escrito, esto quiere decir que nos “movimos” hacia la carpeta deseada. Podemos verificar esto escribiendo nuevamente el comando **ls** y verificando los archivos contenedores.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\leand\Desktop\Digital House\V2\textEditorApp> ls

    Directorio: C:\Users\leand\Desktop\Digital House\V2\textEditorApp

Mode                LastWriteTime         Length Name
----                -
d-----         27/06/2021 12:14 a. m.             terminado
d-----         27/06/2021 12:14 a. m.             tutorial

PS C:\Users\leand\Desktop\Digital House\V2\textEditorApp> |
```

- e. Ahora bien, si queremos volver para atrás, dado que nos confundimos de carpeta, lo haremos escribiendo el comando **cd** seguido de dos puntos ..

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\leand\Desktop\Digital House\V2\textEditorApp> cd ..
PS C:\Users\leand\Desktop\Digital House\V2> |
```



- f. Por último, vamos a acceder a la carpeta correcta, que es “prueba” y ver qué archivos contiene.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\leand\Desktop\Digital House\V2> cd prueba
PS C:\Users\leand\Desktop\Digital House\V2\prueba> ls

    Directorio: C:\Users\leand\Desktop\Digital House\V2\prueba

Mode                LastWriteTime         Length Name
----                -
-a----           02/07/2021 12:50 p. m.           3941 prueba.js

PS C:\Users\leand\Desktop\Digital House\V2\prueba> |
```

- g. Y para ejecutar nuestro archivo .js lo haremos con el comando **node** seguido del nombre del archivo que queremos ejecutar.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\leand\Desktop\Digital House\V2\prueba> node prueba.js |
```

Si realizamos los pasos correctamente deberíamos visualizar en la consola lo que sea que hayamos pedido en nuestro archivo que se muestre por consola:



```
prueba.js x
V2 > prueba > prueba.js > ...
1
2 console.log("Bienvenidos");
3
4
5
6

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\leand\Desktop\Digital House\V2\prueba> node prueba.js
Bienvenidos
PS C:\Users\leand\Desktop\Digital House\V2\prueba> |
```

4. Podemos encontrarnos con los siguientes errores comunes:

- a. Que estemos queriendo acceder a una carpeta que no está dentro de la carpeta en la que estamos trabajando.

```
DIGITAL HOUSE
└─ V2
   └─ prueba
      └─ prueba.js
   └─ restoApp
   └─ textEditorApp
      └─ terminado
      └─ tutorial

V2 > prueba > prueba.js > ...
9
10
11
12
13
14
15

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\leand\Desktop\Digital House\V2\prueba> cd terminado
cd : No se encuentra la ruta de acceso 'C:\Users\leand\Desktop\Digital House\V2\prueba\terminado' porque no existe.
En línea: 1 Carácter: 1
+ cd terminado
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\leand\...rueba\terminado:String) [Set-Location], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.SetLocationCommand

PS C:\Users\leand\Desktop\Digital House\V2\prueba> |
```

En este caso, vemos cómo estamos trabajando en la carpeta prueba, pero queremos acceder a la carpeta “terminado” que se encuentra en textEditorApp, y no en “prueba”, por lo tanto el error nos dice que no se encuentra la ruta de acceso, ya que no existe.



- b. Que queramos ejecutar un archivo que no se encuentra en la carpeta donde estamos trabajando.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\leand\Desktop\Digital House\V2\prueba> node app.js
internal/modules/cjs/loader.js:905
  throw err;
  ^

Error: Cannot find module 'C:\Users\leand\Desktop\Digital House\V2\prueba\app.js'
    at Function.Module._resolveFilename (internal/modules/cjs/loader.js:902:15)
    at Function.Module._load (internal/modules/cjs/loader.js:746:27)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:76:12)
    at internal/main/run_main_module.js:17:47 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}
PS C:\Users\leand\Desktop\Digital House\V2\prueba> 
```

Quizás el error más común es el que sucede cuando ejecutamos nuestro código, pero el nombre del archivo no se encuentra. Si nos encontramos con este error verifiquemos que estemos en la carpeta correcta y que nuestro archivo se llame de la manera en que lo escribimos.

Estos son los errores más comunes a los que nos enfrentaremos, por supuesto siempre es una buena práctica leer el error, ya que en la mayoría de los casos, nos va a indicar en su mensaje cuál es la equivocación en cuestión, y así poder solucionarla o buscar la manera de hacerlo —ya sea mediante esta herramienta, algún colega, o incluso Google; quizás nuestra mejor herramienta y amigo—.