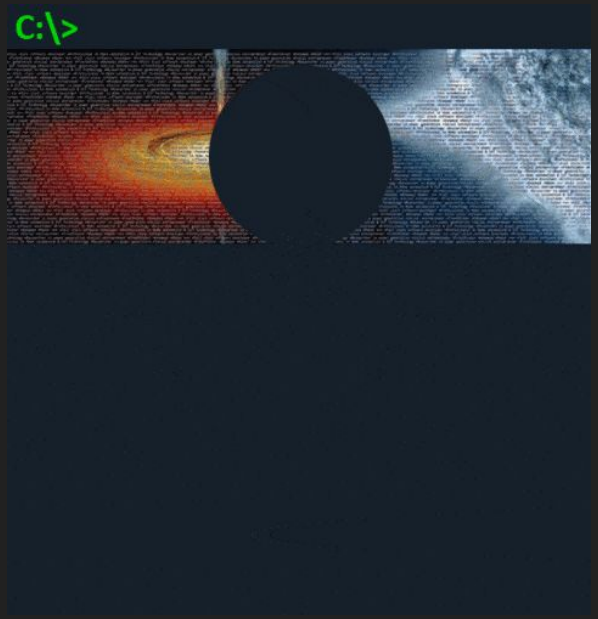
The Power of CLI

"Lo sofisticado es enemigo de lo simple"

Víctor A. Higuita C. <vickodev@gmail.com> <@vickodev>

¡Hola a tod@s!



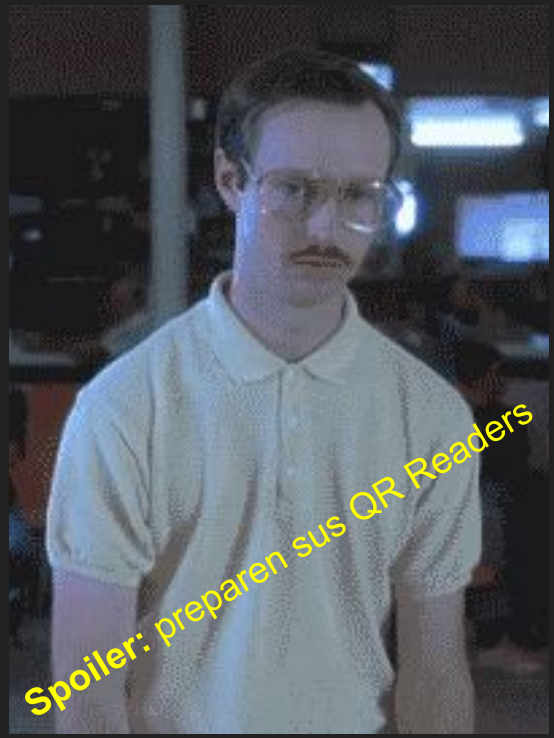
¡Curiosidades! 🧠

- En el 2017 consumimos un aproximado de 157,133 TW de energía (≈85% Combustible fósil).
- Reemplazar la energía por combustible fósil, requiere ≈1080 veces el PIB Mundial estimado para 2019.
- La contaminación por plástico es tan alarmante que cada semana ingerimos el equivalente a una TC.
- El recalentamiento global es un problema de retroalimentación positiva.

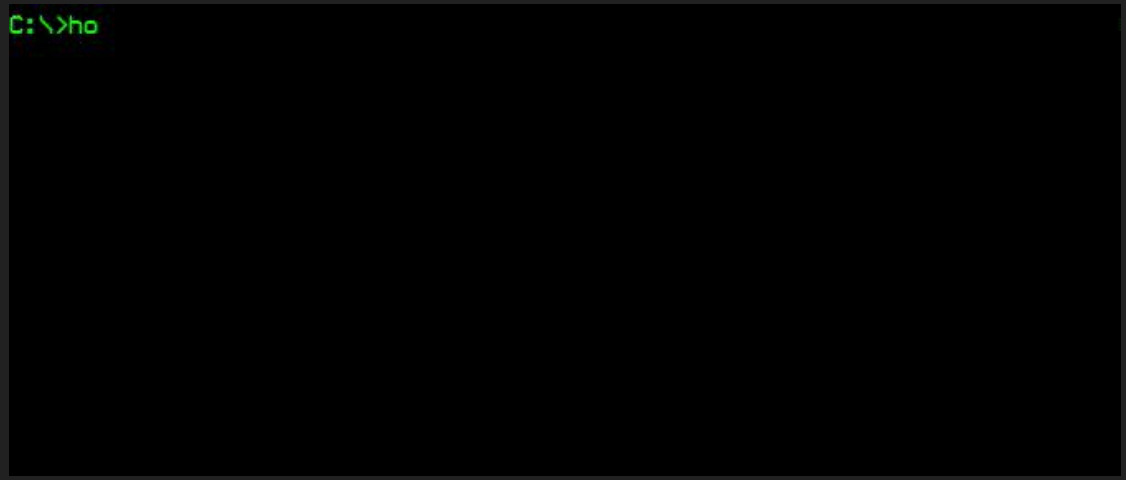
¿Qué veremos?

- Breve historia de los CLIs
- La solución, ¿a qué?😱
- Qué podemos hacer con un CLI
- Conceptos claves para crear CLIs en NodeJS.
- Características de los CLI con UX, “Amigables”🤨
- Herramientas (paquetes) que tenemos para crear CLIs
- Crear la base de un CLI (Coding)* 🤓
- Demo* 👽

*Si nos da el tiempo



Una breve historia



Al principio todo era oscuridad👀, se crearon los sistemas de cómputo (se hizo la “luz”), entonces llegaron los CLI y gobernaron el mundo desde que empezó esta revolución (todavía), pero inicialmente fueron un dolor de cabeza🤒, ¿por qué?

A ¡Y el KMDFK Problema! 😵 B

SYNTAX AS A COMMAND:

```
asp search_list search_path1 {-control_args} ...
    search_pathN {-control_args}
```

FUNCTION: adds one or more search paths to the specified search list.

EXAMPLES:

Multics OS (GE, LabBell, MIT) 1964
Multiplexed Information and Computing Sevice

The command line:

```
! asp info info_files -after >doc>info
```

adds the absolute pathname represented by the relative pathname info_files as a search path to the info search list. This new search path is positioned in the info search list after the >doc>info search path.

SYNTAX AS AN ACTIVE FUNCTION:

```
[be strA strB]
```

FUNCTION: returns the string preceding the first occurrence of strB in strA. If strB does not occur in strA, the entire string strA is returned.

EXAMPLES:

```
! string [before abcdef123defabc def]
  abc
! string [before abcdef g]
  abcdef
! string [before abcdef123 abc]

! string
  [format_line XY^aZZ [before 1.4596e+17 7]]
  XY1.4596e+1ZZ
```

Utility ROM Commands

Control Program for MicroComputers
CP/M (Digital Research Inc) 8/16/32 (Intel 8080/85)

CONFIG
PX-4 configuration tool.

FILINK
Transfer files through the serial port using the FILINK protocol.

PIP [d:][newfile.ext]=[d:][filespec][<t>
Copy files between peripherals whilst performing optional conversion *<t>*.

Copy file from drive to drive: `C> PIP H:=A:INFO.DAT`
Copy file to new name: `C> PIP A:NEW.DAT=A:INFO.DAT`
Copy file to printer: `C> PIP LST:=A:LETTER.TXT`

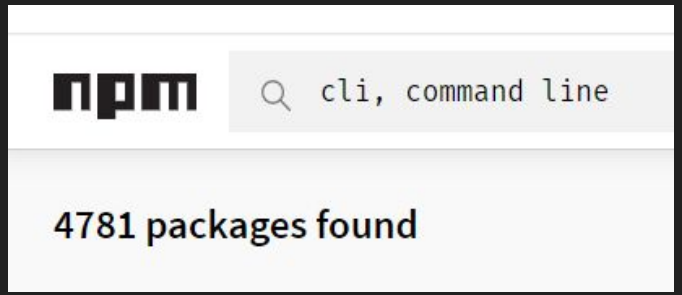
In addition, a range of parameters can be specified in square brackets [] immediately after the command to perform various translations during the copy.

[B]	Block Mode Transfer - Copy data in blocks, used with devices such as paper tape readers.
[E]	Echo to screen - copy to screen as well as destination
[P*n*]	Form Feed - Insert a form feed (ASCII code 12, 0x0C) every *n* lines.
[F]	Supress form feeds
[H]	Hex format - only allow copying of intel HEX format files, terminate if match fails
[I]	Ignore NULL records - An alternative/extension to [H], ignores NULL (hex 00) records and ensures content is in Intel HEX format.
[L]	Lower case conversion - converts all upper case characters to lower case.
[N]	Number lines - Move text to start at column 9, and insert right justified line numbering in the space
[N2]	As [N] but zero pad the number to 6 digits
[O]	Object file transfer - would otherwise break on non ascii/hex files
[R]	Read System Files - read and copy system, sets [W]
[Q*string*^Z]	Stop copying at specified *string*, ^Z refers to **CTRL-Z**

El estándar llegó (80’s by AT&T)🤕

- En 1985 AT&T entrega al dominio público GETOPT en la conferencia UNIFORUM en Dallas y se especifica en el estándar POSIX.2 por la IEEE (-).
- Se crearon varias derivaciones de GETOPT, especialmente una para los sistemas GNU llamada GETOPT_LONG con esteroides (-,--,=,multi)

¿Qué podemos hacer con CLI en NodeJS?🤔



Los usas con Vue, React, Angular, Now, Webpack, Expo, NodeJS y cientos más, hasta podría estar el tuyo, ¿por qué no?

Y entonces, ¿qué necesitamos para hacerlo?🤓

Conocer un concepto básico y otros detalles, así que ¡Vamos a ver! 🤩

Conozcamos a “process.argv”

A

index.js × | package.json | launch.json

index.js ▸ processArgv

```
1  'use strict';
2
3  const processArgv = process.argv;
4
5  console.log(processArgv);
6  console.log('I finished');
7
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

B

index.js ▸ ...

```
1   'use strict';
2
3   const processArgv = process.argv;
4
5   if (!processArgv[2]){
6     console.log('No entry');
7     process.exit(1);
8   } else {
9     console.log('System params: ', processArgv.slice(0,2));
10    console.log('User params: ', processArgv.slice(2));
11  }
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

```
MINGW64 /e/Repos/node-cli (master)
$ node index myCliApp action --param1 value1 --param2 value2 --param3 value3
System params:  [ 'C:\\Program Files\\nodejs\\node.exe',
  'E:\\Repos\\node-cli\\index' ]
User params:  [ 'myCliApp',
  'action',
  '--param1',
  'value1',
  '--param2',
  'value2',
  '--param3',
  'value3' ]
```

process.argv es un parámetro (array) que contiene la información del comando utilizado para la lanzar el script en ejecución.

A => script executed by debugger
B => script executed by command

Un CLI con UX🤭debe:

Tener ejecución directa

Shebang: (#!) secuencia usada en UNIX para indicar que un archivo de texto es un ejecutable.
Formato: #!interpreter [optional-arg]

Entonces debemos crear una instancia de Shebang line al inicio de nuestro script de la siguiente manera: `#!/usr/bin/env node`

Posteriormente agregar a nuestro package.json una rama de nombre “bin”, así:

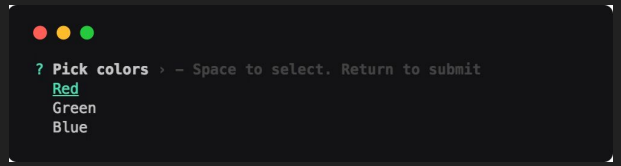
```
"bin": {
  "nodewk": "./index.js"
}
```

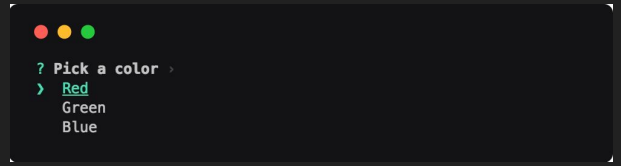
*Debemos ejecutar “npm link” para no usar “node index.js” sino “nodewk [optional-args]”

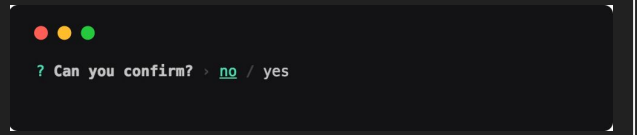
E Interactividad (Esteroides)

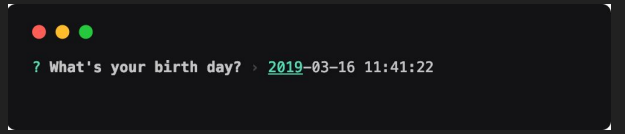
- Parsear entradas
- Autocompletar
- Feedback de errores
- Poder responder preguntas
- Validar respuestas
- Ojo con los password
- Selección múltiple
- Progreso de una actividad
- Entre otros...

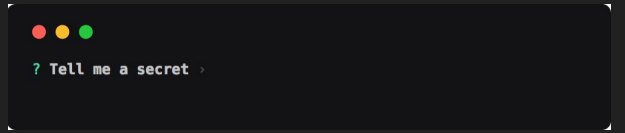
Herramientas de UX para CLIs 🤯











args, yargs, commander, minimist ⇔ prompts, prompt, enquirer, inquirer, prompt-base

Antes, vamos a dar Créditos a:

Eso es todo, ahora al Coding y al Demo

Repositorio

@vickodev

Víctor A. Higuita C. <vickodev@gmail.com> <@vickodev>