

Pokédex Bash: toma de decisiones y comprobaciones en scripts

Instala la herramienta jq

- sudo apt install jq

```
li@li:~$ sudo apt install jq
```

Crea un directorio y entra, a continuación, crea un archivo JSON.

- sudo nano datos.json

```
li@li:~$ mkdir EjJSON
li@li:~$ cd EjJSON
li@li:~/EjJSON$ sudo nano datos.json
```

Introduzca el siguiente contenido

```
{
  "nombre": "Ana",
  "edad": 30,
  "skills": ["bash", "python", "jq"],
  "direccion": { "ciudad": "Madrid", "cp": "28001" },
  "usuarios": [
    { "nombre": "Ana", "edad": 30 },
    { "nombre": "Luis", "edad": 22 }
  ]
}
```

```
GNU nano 7.2                                     datos.json
{
  "nombre": "Ana",
  "edad": 30,
  "skills": ["bash", "python", "jq"],
  "direccion": { "ciudad": "Madrid", "cp": "28001" },
  "usuarios": [
    { "nombre": "Ana", "edad": 30 },
    { "nombre": "Luis", "edad": 22 }
  ]
}
```

Mostrar todo formateado

- jq . datos.json

```
li@li:~/EjJSON$ jq . datos.json
{
  "nombre": "Ana",
  "edad": 30,
  "skills": [
    "bash",
    "python",
    "jq"
  ],
  "direccion": {
    "ciudad": "Madrid",
    "cp": "28001"
  },
  "usuarios": [
    {
      "nombre": "Ana",
      "edad": 30
    },
    {
      "nombre": "Luis",
      "edad": 22
    }
  ]
}
```

Obtener campo como texto sin comillas

- jq -r '.nombre' datos.json

```
li@li:~/EjJSON$ jq -r '.nombre' datos.json
Ana
```

Obtener número

- jq '.edad' datos.json

```
li@li:~/EjJSON$ jq '.edad' datos.json
30
```

Campo anidado

- jq -r '.direccion.ciudad' datos.json

```
li@li:~/EjJSON$ jq -r '.direccion.ciudad' datos.json
Madrid
```

Primer elemento de un array

- jq -r '.skills[0]' datos.json

```
li@li:~/EjJSON$ jq -r '.skills[0]' datos.json
bash
```

Filtrar dentro de un array de objetos

- jq -r '.usuarios[] | select(.edad > 25) | .nombre' datos.json

```
li@li:~/EjJSON$ jq -r '.usuarios[] | select(.edad > 25) | .nombre' datos.json
Ana
```

Iterar elementos (por ejemplo, imprimir cada skill)

- jq -r '.skills[]' datos.json | while read -r skill; do
echo "Skill: \$skill"
done

```
li@li:~/EjJSON$ jq -r '.skills[]' datos.json | while read -r skill; do
  echo "Skill: $skill"
done
Skill: bash
Skill: python
Skill: jq
```

Crear un script: [superScript.sh](#)

- sudo nano superScript.sh

```
li@li:~/EjJSON$ sudo nano superScript.sh
```

Introduzca el siguiente contenido

```
#!/usr/bin/env bash
# =====
# Script: leer_json.sh
# Descripción: Comprueba si jq está instalado.
#           Si no lo está, lo instala (Ubuntu/Raspbian)
#           y luego lee datos desde un archivo JSON.
# =====

# Colores para hacerlo más legible
verde="\e[32m"
rojo="\e[31m"
amarillo="\e[33m"
reset="\e[0m"

# --- 1) Comprobar si jq está instalado ---
echo -e "${amarillo}Comprobando si jq está instalado...${reset}"
if ! command -v jq &> /dev/null; then
  echo -e "${rojo}jq no está instalado.${reset}"

# Detectar si es Ubuntu o Raspbian
if grep -qi "ubuntu" /etc/os-release || grep -qi "raspbian" /etc/os-release; then
  echo -e "${amarillo}Sistema Ubuntu o Raspbian detectado.${reset}"
  echo "Instalando jq..."
  sudo apt update && sudo apt install -y jq
else
  echo -e "${rojo}Este script solo instala jq automáticamente en Ubuntu o
Raspbian.${reset}"
  echo "Instálalo manualmente con: sudo apt install jq"
```

```

    exit 1
fi

# Verificar que jq se haya instalado correctamente
if ! command -v jq &> /dev/null; then
    echo -e "${rojo}Error: jq no se pudo instalar correctamente.${reset}"
    exit 1
fi
else
    echo -e "${verde}jq ya está instalado.${reset}"
fi

# --- 2) Leer datos de un archivo JSON ---
json_file="datos.json"

if [ ! -f "$json_file" ]; then
    echo -e "${rojo}No se encuentra el archivo ${json_file}.${reset}"
    echo "Crea un archivo con contenido como este:"
    echo '{
"nombre": "Ana",
"edad": 30,
"skills": ["bash", "python", "jq"],
"direccion": { "ciudad": "Madrid", "cp": "28001" }
}'
    exit 1
fi

# --- 3) Extraer información usando jq ---
nombre=$(jq -r '.nombre' "$json_file")
edad=$(jq -r '.edad' "$json_file")
ciudad=$(jq -r '.direccion.ciudad' "$json_file")

# Leer array como lista
mapfile -t skills < <(jq -r '.skills[]' "$json_file")

# --- 4) Mostrar resultados ---
echo -e "\n${verde}== Datos del archivo JSON ==${reset}"
echo "Nombre: $nombre"
echo "Edad: $edad"
echo "Ciudad: $ciudad"
echo "Skills:"
for s in "${skills[@]}"; do
    echo " - $s"
done

echo -e "\n${verde}Lectura completada correctamente.${reset}"

```

```

GNU nano 7.2                                         superScript.sh
#!/usr/bin/env bash
# =====
# Script: leer_json.sh
# Descripción: Comprueba si jq está instalado.
#           Si no lo está, lo instala (Ubuntu/Raspbian)
#           y luego lee datos desde un archivo JSON.
# =====

# Colores para hacerlo más legible
verde="\e[32m"
rojo="\e[31m"
amarillo="\e[33m"
reset="\e[0m"

# --- 1) Comprobar si jq está instalado ---
echo -e "${amarillo}Comprobando si jq está instalado...${reset}"

if ! command -v jq &> /dev/null; then
    echo -e "${rojo}jq no está instalado.${reset}"

    # Detectar si es Ubuntu o Raspbian
    if grep -qi "ubuntu" /etc/os-release || grep -qi "raspbian" /etc/os-release; then
        echo -e "${amarillo}Sistema Ubuntu o Raspbian detectado.${reset}"
        echo "Instalando jq..."

```

Conceder permisos de script

- sudo chmod 777 ./superScript.sh

```

li@li:~/EjJSON$ sudo chmod 777 ./superScript.sh

```

Si el script se ejecuta correctamente, aparecerá la siguiente información.

- ./superScript.sh

```

li@li:~/EjJSON$ ./superScript.sh
Comprobando si jq está instalado...
jq ya está instalado.

```

==== Datos del archivo JSON ===

Nombre: Ana

Edad: 30

Ciudad: Madrid

Skills:

- bash
- python
- jq

Crear un archivo JSON (pokedex.json)

- sudo nano pokedex.json

```
li@li:~/EjJSON$ sudo nano pokedex.json
```

Introduzca el siguiente contenido

```
{  
    "pokemons": [  
        { "numero": 1, "nombre": "Bulbasaur", "tipo": "Planta/Veneno", "nivel": 5 },  
        { "numero": 4, "nombre": "Charmander", "tipo": "Fuego", "nivel": 5 },  
        { "numero": 7, "nombre": "Squirtle", "tipo": "Agua", "nivel": 5 },  
        { "numero": 25, "nombre": "Pikachu", "tipo": "Eléctrico", "nivel": 8 }  
    ]  
}
```

```
GNU nano 7.2                                     pokedex.json *  
{  
    "pokemons": [  
        { "numero": 1, "nombre": "Bulbasaur", "tipo": "Planta/Veneno", "nivel": 5 },  
        { "numero": 4, "nombre": "Charmander", "tipo": "Fuego", "nivel": 5 },  
        { "numero": 7, "nombre": "Squirtle", "tipo": "Agua", "nivel": 5 },  
        { "numero": 25, "nombre": "Pikachu", "tipo": "Eléctrico", "nivel": 8 }  
    ]  
}
```

Crear un script pokedex.sh

```
li@li:~/EjJSON$ sudo nano pokedex.sh
```

Introduzca el siguiente contenido

```
#!/usr/bin/env bash  
# ======  
# Script:pokedex.sh  
# Descripción:Buscar información sobre Pokémon utilizando jq  
# ======  
  
# Colores para hacerlo más legible  
verde="\e[32m"  
rojo="\e[31m"  
amarillo="\e[33m"  
reset="\e[0m"  
  
# --- 1) Comprobar si jq está instalado ---  
echo -e "${amarillo}Comprobando si jq está instalado...${reset}"
```

```

if ! command -v jq &> /dev/null; then
    echo -e "${rojo}jq no está instalado.${reset}"

# Detectar si es Ubuntu o Raspbian
if grep -qi "ubuntu" /etc/os-release || grep -qi "raspbian" /etc/os-release; then
    echo -e "${amarillo}Sistema Ubuntu o Raspbian detectado.${reset}"
    echo "Instalando jq..."
    sudo apt update && sudo apt install -y jq
else
    echo -e "${rojo}Este script solo instala jq automáticamente en Ubuntu o
Raspbian.${reset}"
    echo "Instálalo manualmente con: sudo apt install jq"
    exit 1
fi

# Verificar que jq se haya instalado correctamente
if ! command -v jq &> /dev/null; then
    echo -e "${rojo}Error: jq no se pudo instalar correctamente.${reset}"
    exit 1
fi
else
    echo -e "${verde}jq ya está instalado.${reset}"
fi

# ---2)Comprueba si existe el archivo pokedex.json. ---
json_file="pokedex.json"
if [ ! -f "$json_file" ]; then
    echo -e "${rojo}No se encuentra el archivo ${json_file}. ${reset}"
    echo "Crea el archivo, porfavor."
    exit 1
fi

# --- 3)Obtener el nombre o id del Pokémon ---
read -p "Introduce el nombre o número del Pokémon: " pokemon_input

# Si no se ha introducido nada
if [ -z "$pokemon_input" ]; then
    echo -e "${rojo}El campo no puede estar vacío. Vuelva a ejecutar el script e
introduzca el contenido.${reset}"
    exit 1
fi

# --- 4)Buscar Pokémon ---
# Coincidir simultáneamente tanto el número como el nombre

```

```

result=$(jq -r --arg input "$pokemon_input" '
.pokemons[]
| select(.numero == ($input | tonumber?) or .nombre == $input)
| "Número: \(.numero)\nNombre: \(.nombre)\nTipo: \(.tipo)\nNivel: \(.nivel)"
' "$json_file")

# --- Resultado ---
if [ -z "$result" ]; then
    echo -e "${rojo}No se ha encontrado este Pokémon:${pokemon_input}${reset}"
else
    echo -e "\n${verde}==> Pokémon encontrado ==>${reset}"
    echo "$result"
fi

#!/usr/bin/env bash
# =====
# Script: pokedex.sh
# Descripción: Buscar información sobre Pokémon utilizando jq
# =====

# Colores para hacerlo más legible
verde="\e[32m"
rojo="\e[31m"
amarillo="\e[33m"
reset="\e[0m"

# --- 1) Comprobar si jq está instalado ---
echo -e "${amarillo}Comprobando si jq está instalado...${reset}"

if ! command -v jq &> /dev/null; then
    echo -e "${rojo}jq no está instalado.${reset}"

    # Detectar si es Ubuntu o Raspbian
    if grep -qi "ubuntu" /etc/os-release || grep -qi "raspbian" /etc/os-release; then
        echo -e "${amarillo}Sistema Ubuntu o Raspbian detectado.${reset}"
        echo "Instalando jq..."
        sudo apt update && sudo apt install -y jq
    else

```

Conceder permisos de script

- sudo chmod 777 ./pokedex.sh

```
li@li:~/EjJSON$ sudo chmod 777 ./pokedex.sh
```

Una vez completado, ejecuta el script e introduce el ID de un Pokémon, como por ejemplo 25

```

li@li:~/EjJSON$ ./pokedex.sh
Comprobando si jq está instalado...
jq ya está instalado.
Introduce el nombre o número del Pokémon: 25

==> Pokémon encontrado ==
Número: 25
Nombre: Pikachu
Tipo: Eléctrico
Nivel: 8

```