

# Ayudantia 1 PNG 2021

Alex Villarroel Carrasco

Universidad de Concepción

*avillarroe2019@udec.cl*

22 de julio de 2021



# Bash



¿Qué es?

Es un lenguaje de órdenes muy potente de Unix que hasta la fecha sigue sirviendo y brindando muchas herramientas útiles que usaremos.

I ♥ #!/bin/bash



# Como lo ocupamos?

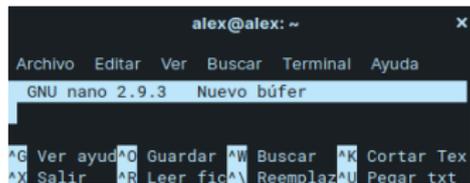
## Scripts

La forma mas utilizada es almacenar las instrucciones en un script, el script actúa como guión para el shell

## Editor de texto

El primer paso para crear un script es elegir un buen editor de texto

Aqui aparece el buen amigo GNU NANO



# Antes que todo(¿o antes que nada?)

## A Considerar

- Siempre en la primera línea:  
`#!/bin/bash`
- Para ejecutarlo se abre la terminal en la ubicación del archivo, y se escribe:  
`./script.sh` , pero antes se deben dar permisos, escribiendo: `chmod u+x script.sh`
- Los comentarios se escriben precediendo el símbolo `#`



## Trucazo

Si el nombre de un archivo es muy largo, apretando la tecla TAB se autocompleta el nombre

# Instrucciones

## Fijar una variable

- nombrevariable='Mi variable'
- echo \$nombrevariable  
Esto es similar al 'Escribir' de PseInt

## IMPORTANTE

NOTAR QUE NO HAY ESPACIO ENTRE EL = cuando se fija una variable, y la variable se llama con un \$ antes



# Condicionales

IF

ELIF

ELSE

Estos condicionales se explican de una manera muy sencilla: "Si esto se cumple haz aquello y si no, pues haz esto, y si nada de lo anterior cumple, entonces haz esto otro".

```
#!/bin/bash
if [[  $\$(\$1 \% 2)$  == 0 ]]
then
    echo Par
else
    echo Impar
fi
```

## Importante

el \$1 es un argumento, este va luego de escribir el nombre del script en la terminal.



# IF

## Sintaxis

Luego de dar las ordenes en el if dentro de un doble corchetes, `[[instrucciones]]`, debe seguir con: `; then. ;` actúa como un salto de linea.

```
alex@alex:~/Escritorio/Ayudantia PNG$ chmod u+x ejemplo.sh
alex@alex:~/Escritorio/Ayudantia PNG$ ./ejemplo.sh 123
Impar
alex@alex:~/Escritorio/Ayudantia PNG$
```

## Argumento

En este caso el argumento \$1 es 123. Es importante mencionar que si un script pide 2 argumentos, estos se separan por un espacio, y si hay menos o más argumentos escritos, arrojará error.



# CASE

## VENTAJAS

- En muchos casos es más cómodo que ir concatenando con If-then
- Es más fácil de leer para la persona que revisa el código.
- Útil cuando tienes muchas opciones o casos.

## DESVENTAJAS

- Se lleva mal con los números :(



# Sintaxis

```
#!/bin/bash
```

```
case <expresion> in
    <patron 1>)
        comandos
        ;;
    <patron 2>)
        comandos
        ;;
    *)
        comandos
        ;;
esac
```

- En la expresión puedes colocar una variable que se haya pedido o un argumento.
- en los patrones puedes colocar nombres, rangos de valores, condiciones, etc.



# Loops

## WHILE

"Mientras esta condición se mantenga, haré esto, si deja de mantenerse, dejo de hacer lo que me pides"

## FOR

El ciclo for básicamente es una estructura iterativa para ejecutar un mismo segmento de código una cantidad de veces deseada, esta cantidad de veces deseada es controlada por una condición que evaluará si sigue o no ejecutando las líneas de código internas.



## Ejercicio 3 Guía 2 Shell

Haga un script en shell que imprima los números del 1 al 10, usando distintas funciones (for, while).

```
#!/bin/bash
contador=1
while [ $contador -lt 11 ]; do
    echo $contador
    let contador=contador+1
done
```

**let** le indica a bash que la variable es de tipo aritmética

## Ejercicio 3 Guía 2 Shell

```
#!/bin/bash
for i in {1..10}; do
    echo $i;
done
```



## Ejercicio 7 Guia 2 Shell

Haga un script en shell que pregunte la edad y, según esta, diga donde debería estar estudiando.

### Analizemos el ejercicio

Nos piden hacer un script que diga donde debería estar estudiando la persona, antes que eso, ¿Qué hay que pedirle al programa? ¿Donde guardamos esa variable? ¿Qué hacemos con ella? ¿Y luego qué? ¿Cómo abordamos el problema?

### IMPORTANTE

Cada persona puede abordar el ejercicio de manera distinta, lo importante es llegar a la respuesta y saber como llegaron a ella (y saber explicarla).



# 1era forma de abordar el ejercicio

```
#!/bin/bash
echo "Ingrese su edad"
read e
if [ 4 -le $e ]; then
    if [ $e -le 5 ]; then
        echo "Deberia cursar preescolar"
    fi
fi
if [ 6 -le $e ]; then
    if [ $e -le 13 ]; then
        echo "Deberia cursar basica"
    fi
fi
if [ 13 -le $e ]; then
    if [ $e -le 17 ]; then
        echo "Deberia cursar media"
    fi
fi
if [ 18 -le $e ]; then
    if [ $e -le 26 ]; then
        echo "Deberia cursar superior"
    fi
fi
else
    echo "Se desconoce que deberia cursar"
fi
```



# Con Case

```
#!/bin/bash
echo 'Ingrese su edad'
read a
case 1 in
  $((a <= 6))
    echo "Prebasica"
    ;;
  $((a <= 13))
    echo "Basica"
    ;;
  $((a <= 17))
    echo "Media"
    ;;
  $((a <= 25))
    echo "Superior"
    ;;
  *)
    echo "Puede o no estar estudiando"
    ;;
esac
```

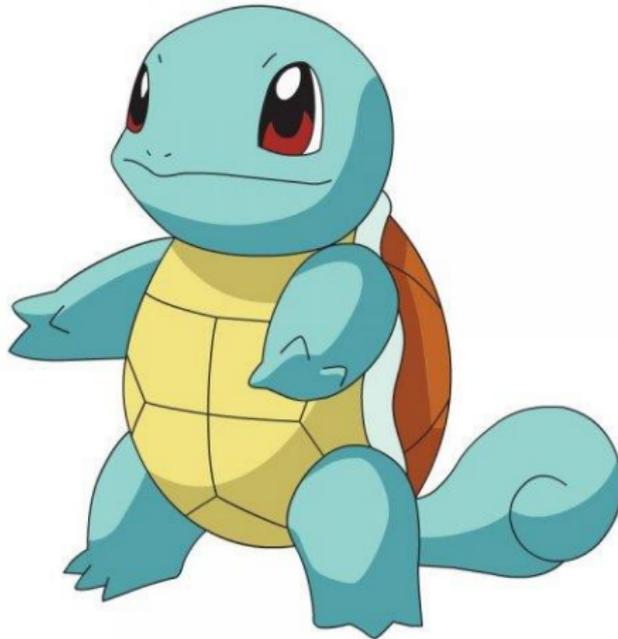


# Con Case

```
#!/bin/bash
echo 'Ingrese su edad'
read a
case $a in
  4|5|6)
    echo "Prebasica"
    ;;
  7|8|9|10|11|12|13)
    echo "Basica"
    ;;
  14|15|16|17|18)
    echo "Media"
    ;;
  19|20|21|22|23|24|25)
    echo "Superior"
    ;;
  *)
    echo "No se que deberia estar cursando"
    ;;
esac
```



VAMO A



PRACTICAR

Imagen creada en [GeneradorMemes.com](https://www.GeneradorMemes.com)



# Fuentes

- `https://www.atareao.es/tutorial/scripts-en-bash/`
- `https://www.mttmlr.com/GMT/apuntes/gmt_guia_2016.pdf`

