Referencias indirectas

indirection, indirect references

usar el valor de una variable para acceder al contenido de otra

```
# declaramos 2 variables
export xyzzy=plugh ; export plugh=cave

echo ${xyzzy}  # normal, xyzzy to plugh -> plugh

echo ${!xyzzy} # indirection, xyzzy to plugh to cave -> cave
```

existen 2 casos en el que este comportamiento se ve modificado:

- `${!N*}` : muestra las variables que empiecen por N
- `${!name[@]}`

```
export myVar="hi"
echo ${!my*} # myVar

export ${!my*}="bye"
echo $myVar # bye
```

/vía: https://stackoverflow.com/questions/8515411/what-is-indirect-expansion-what-does-var-mean

asignación a variable referenciada indirectamente

```
#!/bin/bash

AQUI_VALOR="GHJGHJGHJ"
VARIABLE=AQUI_VALOR

read -p "Valor: " VALOR
echo ${VALOR}

declare "${VARIABLE}=$VALOR"
# typeset equivalente a declare, soportado por otros shells (zsh)
# printf -v "${VARIABLE}" '%s' "${VALOR}"

echo $AQUI_VALOR        # $VALOR
echo "=="
echo $VARIABLE          # $VARIABLE
```

/via: https://stackoverflow.com/questions/9938649/indirect-variable-assignment-in-bash

+ info

- http://tldp.org/LDP/abs/html/ivr.html

- https://stackoverflow.com/questions/8515411/what-is-indirect-expansion-what-does-var-mean
- en arrays:
  https://stackoverflow.com/questions/11180714/how-to-iterate-over-an-array-using-indirect-reference

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/linux:bash:indirectreferences

Last update: **28/11/2018 18:46**