

# Lab. de Programación



Escuela de Educación Secundaria Técnica N°1  
"Juan Bautista Alberdi"  
Conesa - San Nicolás

## 6to Año Informática

### Ejercicios

#### 1. Funcionalidades varias

##### a. Ejercicio 1: comandos internos y externos

¿Son los comandos `umask` y `chmod` comandos internos?

##### b. Ejercicio 2: generación de nombres de archivo

Sea la siguiente lista de archivos:

```
$ ls
bd.class.php    header.inc.php  install.txt     readme.txt
prueba         index.php      mail.class.php
```

1. Muestre los nombres de archivo que terminan en `.php`.
2. Muestre los nombres de archivo que tengan la letra `e` en segunda posición.
3. Muestre los nombres de archivo cuya primera letra esté comprendida entre `a` y `e`.
4. Muestre los nombres de archivo que no comienzan por una vocal.

#### Expresiones complejas (ksh, bash)

5. Muestre los nombres de archivo que no terminan en `.php`.
6. Muestre los nombres de archivo que no terminan ni con `.txt` ni con `.php`.

##### c. Ejercicio 3: separador de comandos

¿Cómo se escriben los dos comandos siguientes en la misma línea?

```
$ cd /tmp
```

```
$ ls -l
```

## 2. Redirecciones

### a. Ejercicio 1

Liste todos los procesos del sistema y redirija el resultado a un archivo.

### b. Ejercicio 2

Sea el comando **who -A**, que genera un mensaje de error:

```
$ who -A
who : opción inválida -- 'A'
```

1. Relance este comando y redirija los errores a un archivo.
2. Relance este comando y haga desaparecer los errores sin redirigir a un archivo en disco.

### c. Ejercicio 3

Ejecute los comandos siguientes:

```
$ touch fic_existe
$ chmod 600 fic_existe fic_noexiste
chmod: no se puede acceder a "noexiste": ...
```

# Soluciones del capítulo Mecanismos esenciales del shell

## 1. Funcionalidades varias

### a. Ejercicio 1: comandos internos y externos

¿Son los comandos **umask** y **chmod** comandos internos?

```
$ type umask
umask is a shell builtin

$ type chmod
chmod is a tracked alias for /usr/bin/chmod
```

**umask** es un comando interno. **chmod** es un comando externo. El comando interno **type** permite saber si un comando se encuentra implementado de manera interna.

### b. Ejercicio 2: generación de nombres de archivo

Sea la siguiente lista de archivos:

```
$ ls
bd.class.php    header.inc.php  install.txt     readme.txt
prueba         index.php      mail.class.php
```

1. Muestre los nombres de archivo que terminan en **.php**.

```
$ ls *.php
bd.class.php  header.inc.php  index.php      mail.class.php
```

2. Muestre los nombres de archivo que tengan la letra **e** en segunda posición.

```
$ ls ?e*
header.inc.php  readme.txt
```

3. Muestre los nombres de archivo cuya primera letra esté comprendida entre **a** y **e**.

```
$ ls [a-e]*
bd.class.php  prueba
```

4. Muestre los nombres de archivo que no comienzan por una vocal.

```
$ ls [!aeiouy]*
bd.class.php  header.inc.php  mail.class.php  readme.txt
```

## Expresiones complejas (ksh, bash)

En bash, debemos activar el reconocimiento de expresiones complejas:

```
$ shopt -s extglob
```

5. Muestre los nombres de archivo que no terminan en **.php**.

```
$ ls !(*.php)
prueba      install.txt  readme.txt ...
```