

Examen Parcial

Ejercicio 1: Justificar la respuesta (cada respuesta 0.5 puntos, total 2 puntos)

- A. Todos los lenguajes de programación son compilados..
☐ Verdadero
☒ Falso
- B. Los links simbólicos son archivos independientes y no hace falta que estén en el mismo file system del archivo al que apuntan.
☐ Verdadero
☒ Falso
- C. Dado un archivo con los permisos `rwxr-xr--`. Indicar la respuesta válida
☐ En octal es 755
☐ Los integrantes del grupo solo pueden leer
☐ Todos pueden ejecutar el archivo al tener permiso el dueño
☒ Todos pueden leer pero no todos pueden ejecutarlo
- D. ¿Qué sucede cuando se ejecuta la siguiente secuencia de comandos
`$ mkdir mi_dir`
`$ touch mi_dir/mi_file`
`$ rmdir mi_dir`
 Sólo una respuesta es válida.
☐ Se elimina solamente `mi_file`
☐ Se eliminan `mi_dir` y `mi_file`
☒ No se elimina nada y se produce un mensaje de error por directorio no vacío
☐ Se solicita confirmación para proceder con el borrado, si se aprueba, se borra todo

Ejercicio 2: (cada respuesta 1 punto, total 3 puntos)

- ✓ A. ¿Qué es un hipervisor y cuál es su función principal?
- ✓ B. ¿Qué es un compilador?
- ✓ C. Mencionar diferencias entre lenguajes de alto nivel y lenguajes de bajo nivel

TUIA - Entorno de Programación

Ejercicio 3 (total 5 puntos)

- a) Detectar y corregir 2 errores, cada uno 0.5 puntos → 1 punto
- b) Indicar qué realiza el script luego de las correcciones → 1.5 puntos
(tener en cuenta que requiere argumentos por línea de comandos)
- c) Completar la función g → 2.5 puntos

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

```
function f {
```

```
    $X=$1
```

```
    $Y=$2
```

```
    if [ -d $X ]
```

```
        for i in $(seq $Y)
```

```
        do
```

```
            F=$(( $X / $i ))
```

```
            touch $F
```

```
            echo "Data $i" > $F
```

```
        done
```

```
    else
```

```
        echo "Impossible!"
```

```
        exit 1
```

```
    fi
```

```
}
```

```
function g {
```

```
    #Validar que se reciban 2 argumentos
```

```
    #argumento1 debe ser un directorio válido
```

```
    #argumento2 debe ser un entero positivo
```

```
    #Debe generar un archivo llamado /tmp/(valor argumento2).data
```

```
    #con el contenido de todos los archivos dentro de argumento1, argumento2 veces
```

```
    #(simplificación: en argumento1 sólo habrán archivos de texto)
```

```
    echo
```

```
}
```

```
A=$1
```

```
B=$2
```

```
[ $# -lt 2 ] && echo "No!!!" && exit 1
```

```
f $A $B
```

```
g $A $B
```

```
exit 0
```