

POWERSHELL (parte B)

- El programa ha de funcionar sin errores
- El programa ha de realizar lo que se plantea en el enunciado
- Utilizar siempre cmdlets, no utilizar alias de comandos del cmd
- Pruebas de SW: Comprobar todos los casos de entrada, contemplando los posibles errores del usuario
- Comentar en el propio código los aspectos más relevantes de cómo has resuelto el ejercicio
- Documentar el script: autor, fecha, finalidad, versión

1. **factorial.ps1** – Calcula el factorial de un número n , introducido por el usuario. El programa debe permitir volver a calcular el factorial de un número hasta que el usuario indique que no quiere continuar

El factorial de un entero positivo n , el factorial de n o n factorial se define como el producto de todos los números enteros positivos desde 1 (es decir, los números naturales) hasta n . Ejemplo: $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$

2. **buscagrandes.ps1** – Volcar en un fichero borrar.txt del escritorio del usuario el nombre y la ruta donde se encuentren los ficheros contenidos en cualquier unidad lógica (c:, d:, ...) cuyo tamaño sea mayor que el tamaño que haya indicado el usuario

3. **quehace.txt** - ¿Qué hace este programa? Soluciona los errores que pueda tener

```
n = "txt", "doc", "xls", "iso"
c = 0
ForEach ($s in $n)
{
    c += 1
    Write-Host "$s"
    Write-Host $c
}
```

4. **triangulo.ps1** – dibujar un triángulo rectángulo como el del ejemplo de altura n que indicará el usuario

```
A
B B
A A A
B B B B
A A A A A
```

5. **creadir.ps1** - Crear un menú interactivo para que un usuario final que no sepa usar cmdlets pueda gestionar directorios. El programa debe preguntar el nombre del fichero y la ubicación original/final en cada caso

- A. Crear directorio
- B. Copiar directorio
- C. Borrar directorio
- D. Mover directorio
- E. Salir