

Desafío 4

Paso a paso para ejecución de script	2
Ejecución	2
Explicación Técnica	3
main.sh	3
funciones.sh	5

Paso a paso para ejecución de script

1. **Descarga los scripts:**
 - `funciones.sh`
 - `main.sh`
 - `index.html`
2. **Asegúrate de tener permisos de superusuario.** Esto es necesario ya que el script realiza operaciones que requieren permisos elevados.
3. **Ubicación de los scripts:** Coloca ambos scripts en el mismo directorio de trabajo para facilitar su ejecución.
4. **Asegúrate de que `index.html` esté en el mismo directorio que los scripts.** Este archivo se copiará al directorio raíz de Tomcat.

Ejecución

Concede permisos de ejecución a los scripts:

```
chmod +x funciones.sh
```

```
chmod +x main.sh
```

```
chmod +x index.html
```

Ejecuta el script principal:

```
sudo bash main.sh
```

Explicación Técnica

main.sh

Este script es el script principal que llama a las funciones definidas en `funciones.sh`.

Incluye las funciones definidas en `funciones.sh` para que estén disponibles en este script.

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

```
# Incluir las funciones desde funciones.sh
```

```
source ./funciones.sh
```

Registra un mensaje en el sistema indicando que la ejecución del script principal ha comenzado.

```
log_message "Ejecución del script principal iniciada."
```

Llama a la función `detect_linux_distribution` para determinar la distribución de Linux en la que se está ejecutando el script.

```
# Detectar la distribución de Linux
```

```
log_message "Ejecución de la función para detectar distribución"
```

```
detect_linux_distribution
```

Llama a la función `detect_package_manager` para identificar el gestor de paquetes utilizado en el sistema y almacena el resultado en la variable `package_manager`.

```
# Detectar el gestor de paquetes
```

```
log_message "Ejecución de la función para detectar gestor de paquetes"
```

```
package_manager=$(detect_package_manager)
```

Llama a la función `update_package_manager` pasando el gestor de paquetes detectado para actualizar los paquetes del sistema.

```
# Actualizar el gestor de paquetes
```

```
log_message "Ejecución de la función para actualizar gestor de paquetes"
```

```
update_package_manager "$package_manager"
```

Llama a la función `install_apache_tomcat` pasando el gestor de paquetes detectado para instalar Apache Tomcat 10.

```
# Instalar Apache Tomcat
```

```
log_message "Ejecución de la función para instalar Tomcat"
```

```
install_apache_tomcat "$package_manager"
```

Llama a la función `check_and_create_tomcat_root` para verificar si el directorio raíz de Tomcat 10 existe, y si no, lo crea.

```
# Función para verificar y crear el directorio raíz de Tomcat 10
```

```
log_message "Ejecución de la función para verificar y crear el directorio raíz de Tomcat 10"
```

```
check_and_create_tomcat_root
```

Llama a la función `copy_index_html` para copiar el archivo `index.html` al directorio raíz de Tomcat 10.

```
# Copiar el archivo index.html
```

```
log_message "Ejecución de la función para copiar archivo index.html"
```

```
copy_index_html
```

Registra un mensaje en el sistema indicando que la ejecución del script principal ha terminado.

```
log_message "Ejecución del script principal completada."
```

funciones.sh

Este script define varias funciones usadas por `main.sh`.

Define la función `log_message` que registra un mensaje en el log del sistema.

Función de logging

```
log_message() {  
  
    local message=$1  
  
    logger "$message"  
  
}
```

Define la función `detect_linux_distribution` que detecta y muestra la distribución de Linux instalada leyendo el archivo `/etc/os-release`.

Función para detectar y mostrar la distribución de Linux instalada

```
detect_linux_distribution() {  
  
    local distribution=$(grep PRETTY_NAME /etc/os-release | cut -d=  
-f2 | tr -d ' ')  
  
    if [ -z "$distribution" ]; then  
  
        distribution="Unknown"  
  
    fi  
  
  
    echo "Distribución de Linux detectada: $distribution"  
  
    log_message "Distribución de Linux detectada: $distribution"  
  
}
```

Define la función `detect_package_manager` que detecta cuál gestor de paquetes está instalado en el sistema verificando la existencia de comandos como `apt`, `dnf`, `yum`, `pacman` y `zypper`.

Función para detectar el gestor de paquetes instalado

```
detect_package_manager() {  
  
    local package_manager="Unknown"  
  
    if command -v apt &>/dev/null; then  
        package_manager="APT"  
    elif command -v dnf &>/dev/null; then  
        package_manager="DNF"  
    elif command -v yum &>/dev/null; then  
        package_manager="YUM"  
    elif command -v pacman &>/dev/null; then  
        package_manager="Pacman"  
    elif command -v zypper &>/dev/null; then  
        package_manager="Zypper"  
    fi  
  
    echo $package_manager  
  
    log_message "Gestor de paquetes detectado: $package_manager"  
  
}
```

Define la función `update_package_manager` que actualiza el sistema utilizando el gestor de paquetes detectado. Utiliza un `case` para manejar diferentes gestores de paquetes.

```
# Función para actualizar el gestor de paquetes
```

```
update_package_manager() {  
  
    local package_manager=$1  
  
    case $package_manager in  
  
        "APT")  
  
            sudo apt update && sudo apt upgrade -y  
  
            ;;  
  
        "DNF")  
  
            sudo dnf upgrade -y  
  
            ;;  
  
        "YUM")  
  
            sudo yum upgrade -y  
  
            ;;  
  
        "Pacman")  
  
            sudo pacman -Syu --noconfirm  
  
            ;;  
  
        "Zypper")  
  
            sudo zypper update -y  
  
            ;;  
  
        *)  
  
            echo "Gestor de paquetes no reconocido."  
  
            log_message "Gestor de paquetes no reconocido."  
  
            ;;  
  
    esac  
}
```



```
        esac

        log_message "Actualización del sistema completada usando
$package_manager."
    }
}
```

Define la función `install_apache_tomcat` que instala Apache Tomcat 10 utilizando el gestor de paquetes detectado. Utiliza un `case` para manejar diferentes gestores de paquetes.

```
# Función para instalar Apache Tomcat
```

```
install_apache_tomcat() {

    local package_manager=$1

    case $package_manager in

        "APT")

            sudo apt install tomcat10 -y

            ;;

        "DNF")

            sudo dnf install tomcat10 -y

            ;;

        "YUM")

            sudo yum install tomcat10 -y

            ;;

        "Pacman")

            sudo pacman -S tomcat10 -y

            ;;

        "Zypper")

            sudo zypper install tomcat10 -y
```

```

        ;;
    *)
        echo "Gestor de paquetes no reconocido."
        log_message "Gestor de paquetes no reconocido."
        ;;
esac

log_message "Apache Tomcat instalado usando $package_manager."
}

```

Define la función `check_and_create_tomcat_root` que verifica si el directorio raíz de Tomcat 10 existe, y si no, lo crea.

Función para verificar y crear el directorio raíz de Tomcat 10

```

check_and_create_tomcat_root() {

    local tomcat_root_dir="/usr/share/tomcat10-root/default_root/"

    if [ -d "$tomcat_root_dir" ]; then

        echo "El directorio raíz de Tomcat ya existe en:
$tomcat_root_dir"

        log_message "El directorio raíz de Tomcat ya existe en:
$tomcat_root_dir."

    else

        echo "El directorio raíz de Tomcat no existe. Creando el
directorio en: $tomcat_root_dir"

        log_message "El directorio raíz de Tomcat no existe. Creando
el directorio en: $tomcat_root_dir."

        sudo mkdir -p "$tomcat_root_dir"
    fi
}

```

```
        echo "Directorio creado exitosamente."

        log_message "Directorio creado exitosamente."

    fi
}
```

Define la función `copy_index_html` que copia el archivo `index.html` al directorio raíz del servidor web Tomcat 10.

```
# Función para copiar un archivo index.html al directorio raíz del
servidor web
```

```
copy_index_html() {

    sudo cp ./index.html /usr/share/tomcat10-root/default_root/

    echo "Archivo index.html copiado al directorio raíz del servidor
web."

    log_message "Archivo index.html copiado al directorio raíz del
servidor web."

}
```