



UNIVERSIDAD VERACRUZANA  
FACULTAD DE ESTADISTICA E INFORMATICA  
XALAPA, VERACRUZ

PROGRAMA EDUCATIVO  
REDES Y SERVICIOS DE COMPUTO

EXPERIENCIA EDUCATIVA  
PROGRAMACION EN LA ADMINISTRACION DE  
REDES

DOCENTE  
JUAN LUIS LOPEZ HERRERA

TRABAJO  
PROYECTO FINAL DE LA EXPERIENCIA EDUCATIVA

ALUMNO  
BENJAMIN RIVERA ROJAS

## Documentación del proyecto de Programación en la Administración en Redes

Instrucciones:

Realizar una implementación de un sistema en Linux y Windows, usando bash y Powershell que implemente las siguientes acciones:

- Alta, Modificación y Eliminación de Usuarios, Directorios en cada equipo al que se tiene acceso, Llaves públicas y privadas de cada usuario.
- Subir archivos por cada usuario a los directorios a los que se tiene permiso,
- Crear directorios por cada usuario dentro de los que ya se tiene permiso.
- Listar los procesos de el mismo usuario en el otro equipo, desde Windows a Linux y viceversa.
- Detener procesos que corren en el otro equipo que sean del mismo usuario.
- Listar archivos y directorios en el otro equipo del usuario.
- Enviar y recibir archivos comprimidos, encriptados y firmados a otro usuario o a sus directorios en el otro equipo.
- Enviar y recibir mensaje encriptados a otro usuario de otro equipo.
- Poder revisar los mensajes recibidos por otros usuarios desde el otro equipo.

### Script principal de Powershell:

```
Add-Type -AssemblyName System.Windows.Forms
Add-Type -AssemblyName System.Drawing

function llenarComboBox($comboBox, $columna){
    foreach ($item in $columna)
    {
        $comboBox.Items.Add($item)
    }
}

#Creación del Form (Interfaz)
$form = New-Object System.Windows.Forms.Form
$form.Text = 'Menú Principal'
$form.Size = New-Object System.Drawing.Size(500,300)
$form.StartPosition = 'CenterScreen'

#Creación de botón Aceptar
$okButton = New-Object System.Windows.Forms.Button
$okButton.Location = New-Object System.Drawing.Point(75,120)
$okButton.Size = New-Object System.Drawing.Size(75,23)
$okButton.Text = 'Aceptar'
$okButton.DialogResult = [System.Windows.Forms.DialogResult]::OK
$form.AcceptButton = $okButton
$form.Controls.add($okButton)

#Creación de botón Cancelar
$cancelButton = New-Object System.Windows.Forms.Button
$cancelButton.Location = New-Object System.Drawing.Point(150,120)
$cancelButton.Size = New-Object System.Drawing.Size(75,23)
$cancelButton.Text = 'Cancelar'
$cancelButton.DialogResult = [System.Windows.Forms.DialogResult]::Cancel
$form.CancelButton = $cancelButton
$form.Controls.add($cancelButton)

#Creación de Label "Elija una opción:"
$labOpcion = New-Object System.Windows.Forms.Label
$labOpcion.Location = New-Object System.Drawing.Point(10,50)
$labOpcion.Size = New-Object System.Drawing.Size(150,20)
$labOpcion.Text = 'Elija una opción:'
$form.Controls.add($labOpcion)
```

```

#Creación de comboBox
$cbOpcion = New-Object System.Windows.Forms.ComboBox
$cbOpcion.Location = New-Object System.Drawing.Point(160,50)
$cbOpcion.Size = New-Object System.Drawing.Size(310,20)
$cbOpcion.Items.AddRange(("Iniciar Programa", "Modificar Base de Datos"));
$cbOpcion.DropDownStyle = [System.Windows.Forms.ComboBoxStyle]::DropDownList
$form.Controls.add($cbOpcion)

#Ejecutar ventana
$form.TopMost = $true
$result = $form.ShowDialog()

#si el resultado de la ventana es OK hacer lo siguiente:
if($result -eq [System.Windows.Forms.DialogResult]::OK){
    #Guarda la opción seleccionada en la variable
    $opcion = $cbOpcion.SelectedItem
    #Guarda
    switch ($opcion){
        "Iniciar Programa" {
            Write-Host "Iniciando programa"
            $form1 = New-Object System.Windows.Forms.Form
            $form1.Text = 'Inicio de programa'
            $form1.Size = New-Object System.Drawing.Size(500,300)
            $form1.StartPosition = 'CenterScreen'

            $form1.AcceptButton = $okButton
            $form1.Controls.add($okButton)

            $form1.CancelButton = $cancelButton
            $form1.Controls.add($cancelButton)

            $labOpcion = New-Object System.Windows.Forms.Label
            $labOpcion.Location = New-Object System.Drawing.Point(10,50)
            $labOpcion.Size = New-Object System.Drawing.Size(150,20)
            $labOpcion.Text = 'Elige una opción:'
            $form1.Controls.add($labOpcion)

            $cbOpcion = New-Object System.Windows.Forms.ComboBox
            $cbOpcion.Location = New-Object System.Drawing.Point(160,50)
            $cbOpcion.Size = New-Object System.Drawing.Size(310,20)
            $cbOpcion.Items.AddRange(("Subir archivos", "Crear directorios",
"Listar procesos", "Detener procesos", "Listar archivos y directorios",
"Enviar archivos", "Enviar mensajes", "Revisar mensajes"));
            $cbOpcion.DropDownStyle =
[System.Windows.Forms.ComboBoxStyle]::DropDownList
            $form1.Controls.add($cbOpcion)

            $labUser = New-Object System.Windows.Forms.Label
            $labUser.Location = New-Object System.Drawing.Point(10,20)
            $labUser.Size = New-Object System.Drawing.Size(150,20)
            $labUser.Text = 'Ingrese usuario: '
            $form1.Controls.add($labUser)

            $txtUser = New-Object System.Windows.Forms.TextBox
            $txtUser.Location = New-Object System.Drawing.Point(160,20)
            $txtUser.Size = New-Object System.Drawing.Size(100,20)
            $form1.Controls.add($txtUser)

            $form1.TopMost = $true
            $result = $form1.ShowDialog()

            $opcion1=$cbOpcion.SelectedItem
            $usuarioDB= $txtUser.Text

            if($result -eq [System.Windows.Forms.DialogResult]::OK -and
(Invoke-Sqlcmd -Query "USE proyecto_programacion; SELECT Nombre FROM
Usuarios;" | Select-Object -ExpandProperty Nombre) -eq $usuarioDB){
                switch($opcion1){
                    "Subir archivos" {
                        Write-Host "Subiendo archivos..."

                        $OpenFileDialog = New-Object
System.Windows.Forms.OpenFileDialog

```

```

'C:\Users\$usuarioDB' $OpenFileDialog.initialDirectory =
$OpenFileDialog.filter = "All files (*.*)|*.*"
$OpenFileDialog.ShowDialog()

$archivo=$OpenFileDialog.FileName

Write-Host $archivo

$archivoSinRuta = Split-Path $archivo -Leaf

Copy-Item $archivo -Destination
\\192.168.163.131\samba\tmp

#Seleccionar el directorio a donde se subir el archivo

$IDUsuario = ((Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT ID_User FROM Usuarios WHERE Nombre =
'$usuarioDB';") | Select-Object -ExpandProperty ID_User)
$directoriodestino = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT Directorios.Directorio FROM Directorios INNER
JOIN Usuarios ON Usuarios.ID_User = Directorios.ID_User WHERE Usuarios.ID_User
= '$IDUsuario';" )

$directoriosSubida = ($directoriodestino | Select-
Object -ExpandProperty Directorio)

$form2 = New-Object System.Windows.Forms.Form
$form2.Text = 'Seleccionar el directorio'
$form2.Size = New-Object System.Drawing.Size(500,300)
$form2.StartPosition = 'CenterScreen'

$form2.AcceptButton = $okButton
$form2.Controls.add($okButton)

$form2.CancelButton = $cancelButton
$form2.Controls.add($cancelButton)

$labFolder = New-Object System.Windows.Forms.Label
$labFolder.Location = New-Object
System.Drawing.Point(10,50)
$labFolder.Size = New-Object
System.Drawing.Size(150,20)
$labFolder.Text = 'Elige un directorio:'
$form2.Controls.add($labFolder)

$cbFolder = New-Object System.Windows.Forms.ComboBox
$cbFolder.Location = New-Object
System.Drawing.Point(160,50)
$cbFolder.Size = New-Object
System.Drawing.Size(310,20)
$lenarComboBox $cbFolder $directoriosSubida
$cbFolder.DropDownStyle =
[System.Windows.Forms.ComboBoxStyle]::DropDownList
$form2.Controls.add($cbFolder)

$form2.TopMost = $true
$result2 = $form2.ShowDialog()

$directoriosSeleccionado = $cbFolder.SelectedItem

Write-Host $directoriosSeleccionado

#Mandar se al a linux para que mueva el fichero a
$directoriosSeleccionado
Set-Content -Value "1" -Path
\\192.168.163.131\samba\tasks\bandera.txt -Force
Set-Content -Value "mv
/home/benjamin/carpetacompartida/tmp/$archivoSinRuta $directoriosSeleccionado
#" -Path \\192.168.163.131\samba\tasks\comandos.sh

}
"Crear directorios" {
Write-Host "Creando directorios..."

```

```

#Menú en donde te permita escribir el nombre del
directorio que se quiere crear
$form3 = New-Object System.Windows.Forms.Form
$form3.Text = 'Creación de directorios'
$form3.Size = New-Object System.Drawing.Size(500,300)
$form3.StartPosition = 'CenterScreen'

$form3.AcceptButton = $okButton
$form3.Controls.add($okButton)

$form3.CancelButton = $cancelButton
$form3.Controls.add($cancelButton)

$labDir = New-Object System.Windows.Forms.Label
$labDir.Location = New-Object
System.Drawing.Point(10,20)
$labDir.Size = New-Object System.Drawing.Size(150,20)
$labDir.Text = 'Ingrese el nombre del nuevo
directorio: '
$form3.Controls.add($labDir)

$txtDir = New-Object System.Windows.Forms.TextBox
$txtDir.Location = New-Object
System.Drawing.Point(160,20)
$txtDir.Size = New-Object System.Drawing.Size(100,20)
$form3.Controls.add($txtDir)

$form3.TopMost = $true
$result3 = $form3.ShowDialog()

$nomDirectorio = $txtDir.Text

$IDUsuario = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT ID_User FROM Usuarios WHERE Nombre =
'$usuarioDB';") | Select-Object -ExpandProperty ID_User
$directoriodestino = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT Directorios.Directorio FROM Directorios INNER
JOIN Usuarios ON Usuarios.ID_User = Directorios.ID_User WHERE Usuarios.ID_User
= $IDUsuario;" )

$directoriosSubida = ($directoriodestino | Select-
Object -ExpandProperty Directorio)

#Otro menú en el que te muestra los directorios
disponibles de la base de datos

$form4 = New-Object System.Windows.Forms.Form
$form4.Text = 'Seleccionar el directorio'
$form4.Size = New-Object System.Drawing.Size(500,300)
$form4.StartPosition = 'CenterScreen'

$form4.AcceptButton = $okButton
$form4.Controls.add($okButton)

$form4.CancelButton = $cancelButton
$form4.Controls.add($cancelButton)

$labFolder = New-Object System.Windows.Forms.Label
$labFolder.Location = New-Object
System.Drawing.Point(10,50)
$labFolder.Size = New-Object
System.Drawing.Size(150,20)
$labFolder.Text = 'Elige un directorio:'
$form4.Controls.add($labFolder)

$cbFolder = New-Object System.Windows.Forms.ComboBox
$cbFolder.Location = New-Object
System.Drawing.Point(160,50)
$cbFolder.Size = New-Object
System.Drawing.Size(310,20)
llenarComboBox $cbFolder $directoriosSubida
$cbFolder.DropDownStyle =
[System.Windows.Forms.ComboBoxStyle]::DropDownList
$form4.Controls.add($cbFolder)

$form4.TopMost = $true

```

```

$result4 = $form4.ShowDialog()

$dirDestino = $cbFolder.SelectedItem

#Comandos para Linux
Set-Content -Value "1" -Path
\\192.168.163.131\samba\tasks\bandera.txt -Force
Set-Content -Value "mkdir $dirDestino$nomDirectorio" -
Path \\192.168.163.131\samba\tasks\comandos.sh

}
"Listar procesos" {
    Write-Host "Listando procesos..."
    #Crear un Archivo en donde estén almacenados los
procesos
    #Procesos.txt o Procesos.csv
    Set-Content -Value "1" -Path
\\192.168.163.131\samba\tasks\bandera.txt -Force
    Set-Content -Value "ps -e -o %p, -o lstart -o ,%C, -o
%mem -o ,%c > /home/benjamin/pruebas/output.csv; cp
/home/benjamin/pruebas/output.csv
/home/benjamin/carpetacompartida/tmp/output.csv #" -Path
\\192.168.163.131\samba\tasks\comandos.sh
    Start-Sleep -Seconds 1

    Get-Content \\192.168.163.131\samba\tmp\output.csv
}
"Detener procesos" {
    Write-Host "Deteniendo procesos..."
    #Archivo que tenga los procesos
    Set-Content -Value "1" -Path
\\192.168.163.131\samba\tasks\bandera.txt -Force
    Set-Content -Value "ps -e -o %p -o,%c >
/home/benjamin/pruebas/output.csv; cp /home/benjamin/pruebas/output.csv
/home/benjamin/carpetacompartida/tmp/output.csv #" -Path
\\192.168.163.131\samba\tasks\comandos.sh

    $CSVProcesos = Import-csv
"\\192.168.163.131\samba\tmp\output.csv"

    #Un Menú donde estén los procesos
    $form5 = New-Object System.Windows.Forms.Form
    $form5.Text = 'Matar procesos'
    $form5.Size = New-Object System.Drawing.Size(500,300)
    $form5.StartPosition = 'CenterScreen'

    $form5.AcceptButton = $okButton
    $form5.Controls.add($okButton)

    $form5.CancelButton = $cancelButton
    $form5.Controls.add($cancelButton)

    $labProcess = New-Object System.Windows.Forms.Label
    $labProcess.Location = New-Object
System.Drawing.Point(10,50)
    $labProcess.Size = New-Object
System.Drawing.Size(150,20)
    $labProcess.Text = 'Elige un directorio:'
    $form5.Controls.add($labProcess)

    $cbProcess = New-Object System.Windows.Forms.ComboBox
    $cbProcess.Location = New-Object
System.Drawing.Point(160,50)
    $cbProcess.Size = New-Object
System.Drawing.Size(310,20)
    foreach($line in $CSVProcesos){
        $cbProcess.Items.Add($line.COMMAND)
    }
    $cbProcess.DropDownStyle =
[System.Windows.Forms.ComboBoxStyle]::DropDownList
    $form5.Controls.add($cbProcess)

    $form5.TopMost = $true
    $result5 = $form5.ShowDialog()

```

```

#Mandar señal
$procesoAmatar = $cbProcess.SelectedItem

foreach($line in $CSVProcesos){
    if($line.command -eq $procesoAmatar){
        $pidAMatar = $line.pid
    }
}

Set-Content -Value "1" -Path
\\192.168.163.131\samba\tasks\bandera.txt -Force
Set-Content -Value "kill -9 $pidAMatar #" -Path
\\192.168.163.131\samba\tasks\comandos.sh
Start-Sleep -Seconds 1

}
"Listar archivos y directorios" {
    Write-Host "Listando archivos..."
    #Seleccionar directorio al cual se obtendrán los
archivos
    $IDUsuario = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT ID_User FROM Usuarios WHERE Nombre =
'$usuarioDB';") | Select-Object -ExpandProperty ID_User
    $directorioDestino = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT Directorios.Directorio FROM Directorios INNER
JOIN Usuarios ON Usuarios.ID_User = Directorios.ID_User WHERE Usuarios.ID_User
= $IDUsuario;" )

    $directoriosSubida = ($directorioDestino | Select-
Object -ExpandProperty Directorio)

    #Otro menú en el que te muestra los directorios
disponibles de la base de datos

    $form6 = New-Object System.Windows.Forms.Form
    $form6.Text = 'Seleccionar el directorio'
    $form6.Size = New-Object System.Drawing.Size(500,300)
    $form6.StartPosition = 'CenterScreen'

    $form6.AcceptButton = $okButton
    $form6.Controls.add($okButton)

    $form6.CancelButton = $cancelButton
    $form6.Controls.add($cancelButton)

    $labDir1 = New-Object System.Windows.Forms.Label
    $labDir1.Location = New-Object
System.Drawing.Point(10,50)
    $labDir1.Size = New-Object System.Drawing.Size(150,20)
    $labDir1.Text = 'Elije un directorio:'
    $form6.Controls.add($labDir1)

    $cbDir1 = New-Object System.Windows.Forms.ComboBox
    $cbDir1.Location = New-Object
System.Drawing.Point(160,50)
    $cbDir1.Size = New-Object System.Drawing.Size(310,20)
    $llenarComboBox $cbDir1 $directoriosSubida
    $cbDir1.DropDownStyle =
[System.Windows.Forms.ComboBoxStyle]::DropDownList
    $form6.Controls.add($cbDir1)

    $form6.TopMost = $true
    $result6 = $form6.ShowDialog()

    $dirSeleccionado = $cbDir1.SelectedItem

    Set-Content -Value "1" -Path
\\192.168.163.131\samba\tasks\bandera.txt -Force
    Set-Content -Value "ls -la $dirSeleccionado >
/home/benjamin/pruebas/output.csv; cp /home/benjamin/pruebas/output.csv
/home/benjamin/carpetacompartida/tmp/output.csv #" -Path
\\192.168.163.131\samba\tasks\comandos.sh
    Start-Sleep -Seconds 1
    Get-Content -Path
\\192.168.163.131\samba\tmp\output.csv
}

```



```

        "Enviar archivos" {
            Write-Host "Enviando archivos..."
            #Selección de archivo y guardar ruta en variable
        }
        "Enviar mensajes" {
            Write-Host "Enviando mensajes..."
            #Carpeta llamada "CarpetaCompartida\mensajes"
            #Enviar archivos txt con mensajes a esa carpeta
        }
        "Revisar mensajes" {
            Write-Host "Revisando mensajes..."
            #Leer archivos.txt de la carpeta mensajes
        }
        default {}
    }
}
else{
    Write-host "El usuario no existe en la base de datos"
}
}

"Modificar Base de Datos" {
    #Menú con las opciones a seleccionar
    $form10 = New-Object System.Windows.Forms.Form
    $form10.Text = 'Inicio de programa'
    $form10.Size = New-Object System.Drawing.Size(500,300)
    $form10.StartPosition = 'CenterScreen'

    $form10.AcceptButton = $okButton
    $form10.Controls.add($okButton)

    $form10.CancelButton = $cancelButton
    $form10.Controls.add($cancelButton)

    $labopcion10 = New-Object System.Windows.Forms.Label
    $labopcion10.Location = New-Object System.Drawing.Point(10,50)
    $labopcion10.Size = New-Object System.Drawing.Size(150,20)
    $labopcion10.Text = 'Elije una opción:'
    $form10.Controls.add($labopcion10)

    $cbopcion10 = New-Object System.Windows.Forms.ComboBox
    $cbopcion10.Location = New-Object System.Drawing.Point(160,50)
    $cbopcion10.Size = New-Object System.Drawing.Size(310,20)
    $cbopcion10.Items.AddRange(("Alta de usuarios", "Modificación de
usuarios", "Baja de usuarios", "Alta de directorios", "Modificación de
directorios", "Baja de directorios"));
    $cbopcion10.DropDownStyle =
[System.Windows.Forms.ComboBoxStyle]::DropDownList
    $form10.Controls.add($cbopcion10)

    $form10.TopMost = $true
    $result10 = $form10.ShowDialog()

    $opcionMod = $cbopcion10.SelectedItem

    if($result10 -eq [System.Windows.Forms.DialogResult]::OK){
        Write-Host "Modificación de la base de datos"
        switch($opcionMod){
            "Alta de usuarios" {
                Write-Host ""
                #Se darán de alta usuarios en SQL Server, los campos
son ID_User, Nombre, Llave_Publica, Llave_Privada

                $form11 = New-Object System.Windows.Forms.Form
                $form11.Text = 'Modificación de la base de datos SQL
SERVER'

                $form11.Size = New-Object System.Drawing.Size(500,300)
                $form11.StartPosition = 'CenterScreen'

                $form11.AcceptButton = $okButton
                $form11.Controls.add($okButton)

                $form11.CancelButton = $cancelButton
                $form11.Controls.add($cancelButton)
            }
        }
    }
}

```



```

        $labSQLUser = New-Object System.Windows.Forms.Label
        $labSQLUser.Location = New-Object
System.Drawing.Point(10,20)
        $labSQLUser.Size = New-Object
System.Drawing.Size(150,20)
        $labSQLUser.Text = 'Ingreso usuario: '
        $form11.Controls.add($labSQLUser)

        $txtSQLUser = New-Object System.Windows.Forms.TextBox
        $txtSQLUser.Location = New-Object
System.Drawing.Point(160,20)
        $txtSQLUser.Size = New-Object
System.Drawing.Size(100,20)
        $form11.Controls.add($txtSQLUser)

        $labPuKey = New-Object System.Windows.Forms.Label
        $labPuKey.Location = New-Object
System.Drawing.Point(10,50)
        $labPuKey.Size = New-Object
System.Drawing.Size(150,20)
        $labPuKey.Text = 'Ingreso llave publica: '
        $form11.Controls.add($labPuKey)

        $txtPuKey = New-Object System.Windows.Forms.TextBox
        $txtPuKey.Location = New-Object
System.Drawing.Point(160,50)
        $txtPuKey.Size = New-Object
System.Drawing.Size(100,20)
        $form11.Controls.add($txtPuKey)

        $labPrKey = New-Object System.Windows.Forms.Label
        $labPrKey.Location = New-Object
System.Drawing.Point(10,80)
        $labPrKey.Size = New-Object
System.Drawing.Size(150,20)
        $labPrKey.Text = 'Ingreso llave privada: '
        $form11.Controls.add($labPrKey)

        $txtPrKey = New-Object System.Windows.Forms.TextBox
        $txtPrKey.Location = New-Object
System.Drawing.Point(160,80)
        $txtPrKey.Size = New-Object
System.Drawing.Size(100,20)
        $form11.Controls.add($txtPrKey)

        $form11.TopMost = $true
        $result11 = $form11.ShowDialog()

        $SQLUser = $txtSQLUser.Text
        $SQLPuKey = $txtPuKey.Text
        $SQLPrKey = $txtPrKey.Text

        #Comandos SQL para insertar los datos
        Invoke-Sqlcmd -Query "USE proyecto_programacion;
INSERT INTO usuarios (Nombre, Llave_Publica, Llave_Privada) VALUES
('$SQLUser', '$SQLPuKey', '$SQLPrKey');"
        Invoke-Sqlcmd -Query "USE proyecto_programacion;
SELECT * FROM usuarios;"
    }

    "Modificacion de usuarios" {
        write-Host ""
        #Ventana para escribir el nombre del usuario, si
        existe en la base de datos podrás editarlo, sino no

        $form12 = New-Object System.Windows.Forms.Form
        $form12.Text = 'Modificación de la base de datos SQL
SERVER'

        $form12.Size = New-Object System.Drawing.Size(500,300)
        $form12.StartPosition = 'CenterScreen'

        $form12.AcceptButton = $okButton
        $form12.Controls.add($okButton)

        $form12.CancelButton = $cancelButton

```

```

$form12.Controls.add($cancelButton)

$labExistU = New-Object System.Windows.Forms.Label
$labExistU.Location = New-Object
System.Drawing.Point(10,20)
$labExistU.Size = New-Object
System.Drawing.Size(150,20)
$labExistU.Text = 'Ingreso usuario: '
$form12.Controls.add($labExistU)

$txtExistU = New-Object System.Windows.Forms.TextBox
$txtExistU.Location = New-Object
System.Drawing.Point(160,20)
$txtExistU.Size = New-Object
System.Drawing.Size(100,20)
$form12.Controls.add($txtExistU)

$form12.TopMost = $true
$result12 = $form12.ShowDialog()

$userExists = $txtExistU.Text

$query = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT Nombre FROM Usuarios WHERE Nombre =
'$userExists'") | Select-Object -ExpandProperty Nombre

if($result12 -eq
[System.Windows.Forms.DialogResult]::OK -and $query -eq $userExists){
    Write-Host "Se modificará $query"
    $formMD = New-Object System.Windows.Forms.Form
    $formMD.Text = 'Modificación de la base de datos
SQL SERVER'
    $formMD.Size = New-Object
System.Drawing.Size(500,300)
    $formMD.StartPosition = 'CenterScreen'

    $formMD.AcceptButton = $okButton
    $formMD.Controls.add($okButton)

    $formMD.CancelButton = $cancelButton
    $formMD.Controls.add($cancelButton)

    $labModUser = New-Object
System.Windows.Forms.Label
$labModUser.Location = New-Object
System.Drawing.Point(10,20)
$labModUser.Size = New-Object
System.Drawing.Size(150,20)
$labModUser.Text = 'Ingrese el nuevo nombre: '
$formMD.Controls.add($labModUser)

    $txtModUser = New-Object
System.Windows.Forms.TextBox
$txtModUser.Location = New-Object
System.Drawing.Point(160,20)
$txtModUser.Size = New-Object
System.Drawing.Size(100,20)
$formMD.Controls.add($txtModUser)

    $labPuKey = New-Object System.Windows.Forms.Label
$labPuKey.Location = New-Object
System.Drawing.Point(10,60)
$labPuKey.Size = New-Object
System.Drawing.Size(150,20)
$labPuKey.Text = 'Ingrese la Llave Publica: '
$formMD.Controls.add($labPuKey)

    $txtPuKey = New-Object
System.Windows.Forms.TextBox
$txtPuKey.Location = New-Object
System.Drawing.Point(160,60)
$txtPuKey.Size = New-Object
System.Drawing.Size(100,20)
$formMD.Controls.add($txtPuKey)

    $labPrKey = New-Object System.Windows.Forms.Label

```

```

        $labPrKey.Location = New-Object
        System.Drawing.Point(10,90)
        $labPrKey.Size = New-Object
        System.Drawing.Size(150,20)
        $labPrKey.Text = 'Ingrese la Llave Privada: '
        $formMD.Controls.add($labPrKey)

        $txtPrKey = New-Object
        System.Windows.Forms.TextBox
        $txtPrKey.Location = New-Object
        System.Drawing.Point(160,90)
        $txtPrKey.Size = New-Object
        System.Drawing.Size(100,20)
        $formMD.Controls.add($txtPrKey)

        $formMD.TopMost = $true
        $resultMD = $formMD.ShowDialog()

        $modUser = $txtModUser.Text
        $prKey = $txtPrKey.Text
        $puKey = $txtPuKey.Text

        Invoke-Sqlcmd -Query "USE proyecto_programacion;
        UPDATE Usuarios SET Nombre = '$modUser', Llave_Publica = '$puKey',
        Llave_Privada= '$prKey' WHERE Nombre= '$query';"
        Invoke-Sqlcmd -Query "USE proyecto_programacion;
        SELECT * FROM usuarios;"

    }

}

"Baja de usuarios" {
    write-Host "Dar de baja a los usuarios de SQL"
    $form13 = New-Object System.Windows.Forms.Form
    $form13.Text = 'Modificación de la base de datos SQL

SERVER'

    $form13.Size = New-Object System.Drawing.Size(500,300)
    $form13.StartPosition = 'CenterScreen'

    $form13.AcceptButton = $okButton
    $form13.Controls.add($okButton)

    $form13.CancelButton = $cancelButton
    $form13.Controls.add($cancelButton)

    $labExistU = New-Object System.Windows.Forms.Label
    $labExistU.Location = New-Object
    System.Drawing.Point(10,20)
    $labExistU.Size = New-Object
    System.Drawing.Size(150,20)
    $labExistU.Text = 'Ingrese usuario: '
    $form13.Controls.add($labExistU)

    $txtExistU = New-Object System.Windows.Forms.TextBox
    $txtExistU.Location = New-Object
    System.Drawing.Point(160,20)
    $txtExistU.Size = New-Object
    System.Drawing.Size(100,20)
    $form13.Controls.add($txtExistU)

    $form13.TopMost = $true
    $result13 = $form13.ShowDialog()

    $userExists = $txtExistU.Text

    $query = (Invoke-Sqlcmd -Query "USE
    proyecto_programacion; SELECT Nombre FROM Usuarios WHERE Nombre =
    '$userExists'") | Select-Object -ExpandProperty Nombre

    if($result13 -eq
[System.Windows.Forms.DialogResult]::OK -and $query -eq $userExists){
        write-Host "Se modificará $query"
        $formED = New-Object System.Windows.Forms.Form
        $formED.Text = 'Modificación de la base de datos

SQL SERVER'

```

```

        $formED.Size = New-Object
System.Drawing.Size(500,300)
        $formED.StartPosition = 'CenterScreen'

        $formED.AcceptButton = $okButton
        $formED.Controls.add($okButton)

        $formED.CancelButton = $cancelButton
        $formED.Controls.add($cancelButton)

        $labEliUser = New-Object
System.Windows.Forms.Label
        $labEliUser.Location = New-Object
System.Drawing.Point(10,20)
        $labEliUser.Size = New-Object
System.Drawing.Size(150,20)
        $labEliUser.Text = 'Ingrese el nombre del usuario
que desee eliminar: '
        $formED.Controls.add($labEliUser)

        $txtEliUser = New-Object
System.Windows.Forms.TextBox
        $txtEliUser.Location = New-Object
System.Drawing.Point(160,20)
        $txtEliUser.Size = New-Object
System.Drawing.Size(100,20)
        $formED.Controls.add($txtEliUser)

        $formED.TopMost = $true
        $resultED = $formED.ShowDialog()

        $eliUser = $txtEliUser.Text

        Invoke-Sqlcmd -Query "USE proyecto_programacion;
DELETE FROM Usuarios WHERE Nombre = '$eliUser';"
        Invoke-Sqlcmd -Query "USE proyecto_programacion;
SELECT * FROM usuarios;"
    }
}

"Alta de directorios" {
    write-Host ""
    #Se darán de alta directorios en SQL Server, los
campos son ID_Directorio, Directorio, ID_User

    #consultas
    #Seleccionar el directorio a donde se subir el archivo

    $nomUsuarios = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT Nombre FROM Usuarios;")

    $SQLUsuarios = ($nomUsuarios | select-object -
ExpandProperty Nombre)

    $form14 = New-Object System.Windows.Forms.Form
    $form14.Text = 'Modificación de la base de datos SQL
SERVER'

    $form14.Size = New-Object System.Drawing.Size(500,300)
    $form14.StartPosition = 'CenterScreen'

    $form14.AcceptButton = $okButton
    $form14.Controls.add($okButton)

    $form14.CancelButton = $cancelButton
    $form14.Controls.add($cancelButton)

    $labSQLDir = New-Object System.Windows.Forms.Label
    $labSQLDir.Location = New-Object
System.Drawing.Point(10,20)
    $labSQLDir.Size = New-Object
System.Drawing.Size(150,40)
    $labSQLDir.Text = 'Ingrese directorio: ej:
/home/benja/'
    $form14.Controls.add($labSQLDir)

```

```

        $txtSQLDir = New-Object System.Windows.Forms.TextBox
        $txtSQLDir.Location = New-Object
System.Drawing.Point(160,20)
        $txtSQLDir.Size = New-Object
System.Drawing.Size(100,40)
        $form14.Controls.add($txtSQLDir)

        $labIDUser = New-Object System.Windows.Forms.Label
        $labIDUser.Location = New-Object
System.Drawing.Point(10,70)
        $labIDUser.Size = New-Object
System.Drawing.Size(150,20)
        $labIDUser.Text = 'Elije un usuario:'
        $form14.Controls.add($labIDUser)

        $cbIDUser = New-Object System.Windows.Forms.ComboBox
        $cbIDUser.Location = New-Object
System.Drawing.Point(160,70)
        $cbIDUser.Size = New-Object
System.Drawing.Size(310,20)
        $lenarComboBox $cbIDUser $SQLUsuarios
        $cbIDUser.DropDownStyle =
[System.Windows.Forms.ComboBoxStyle]::DropDownList
        $form14.Controls.add($cbIDUser)

        $form14.TopMost = $true
        $result14 = $form14.ShowDialog()

        $txtSQLDir = $txtSQLDir.Text
        $cbIDUser = $cbIDUser.SelectedItem

        $consultaAltaDir = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT Directorios.Directorio FROM Directorios INNER
JOIN Usuarios ON Usuarios.ID_User = Directorios.ID_User;" )

        #Comandos SQL para insertar los datos

        $IDUsersQL = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT ID_User FROM usuarios WHERE Nombre =
'$cbIDUser';") | Select-Object -ExpandProperty ID_User

        Invoke-Sqlcmd -Query "USE proyecto_programacion;
INSERT INTO directorios (Directorio, ID_User) VALUES ('$txtSQLDir',
'$IDUsersQL');"
        Invoke-Sqlcmd -Query "USE proyecto_programacion;
SELECT * FROM directorios;"
    }

    "Modificacion de directorios" {
        Write-Host "Modificación de Directorios"
        $form15 = New-Object System.Windows.Forms.Form
        $form15.Text = 'Modificación de la base de datos SQL
SERVER'

        $form15.Size = New-Object System.Drawing.Size(500,300)
        $form15.StartPosition = 'CenterScreen'

        $form15.AcceptButton = $okButton
        $form15.Controls.add($okButton)

        $form15.CancelButton = $cancelButton
        $form15.Controls.add($cancelButton)

        $labExistU2 = New-Object System.Windows.Forms.Label
        $labExistU2.Location = New-Object
System.Drawing.Point(10,20)
        $labExistU2.Size = New-Object
System.Drawing.Size(150,20)
        $labExistU2.Text = 'Ingreso usuario: '
        $form15.Controls.add($labExistU2)

        $txtExistU2 = New-Object System.Windows.Forms.TextBox
        $txtExistU2.Location = New-Object
System.Drawing.Point(160,20)

```

```

        $txtExistU2.Size = New-Object
System.Drawing.Size(100,20)
        $form15.Controls.add($txtExistU2)

        $form15.TopMost = $true
        $result15 = $form15.ShowDialog()

        $userExists2 = $txtExistU2.Text

        $query = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT Nombre FROM Usuarios WHERE Nombre =
'$userExists2'" ) | Select-Object -ExpandProperty Nombre

        if($result15 -eq
[System.Windows.Forms.DialogResult]::OK -and $query -eq $userExists2){
            Write-Host "Se modificará $query"
            #Seleccinar directorio al cual se obtendrán los
archivos

            $IDUsuario = ((Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT ID_User FROM Usuarios WHERE Nombre = '$query';")
| Select-Object -ExpandProperty ID_User)

            $directorioDestino = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT Directorios.Directorio FROM Directorios INNER
JOIN Usuarios ON Usuarios.ID_User = Directorios.ID_User WHERE Usuarios.ID_User
= '$IDUsuario';" )

            $directoriosSubida = ($directorioDestino | Select-
Object -ExpandProperty Directorio)

            $formD1 = New-Object System.Windows.Forms.Form
            $formD1.Text = 'Modificación de la base de datos
SQL SERVER'

            $formD1.Size = New-Object
System.Drawing.Size(500,300)
            $formD1.StartPosition = 'CenterScreen'

            $formD1.AcceptButton = $okButton
            $formD1.Controls.add($okButton)

            $formD1.CancelButton = $cancelButton
            $formD1.Controls.add($cancelButton)

            $labModDir = New-Object System.Windows.Forms.Label
            $labModDir.Location = New-Object
System.Drawing.Point(10,20)
            $labModDir.Size = New-Object
System.Drawing.Size(150,40)
            $labModDir.Text = 'Seleccione el directorio a
cambiar: '

            $formD1.Controls.add($labModDir)

            $cbModDir = New-Object
System.Windows.Forms.ComboBox
            $cbModDir.Location = New-Object
System.Drawing.Point(160,20)
            $cbModDir.Size = New-Object
System.Drawing.Size(310,20)
            $lenarComboBox $cbModDir $directoriosSubida
            $cbModDir.DropDownStyle =
[System.Windows.Forms.ComboBoxStyle]::DropDownList
            $formD1.Controls.add($cbModDir)

            $labNewDir = New-Object System.Windows.Forms.Label
            $labNewDir.Location = New-Object
System.Drawing.Point(10,90)
            $labNewDir.Size = New-Object
System.Drawing.Size(150,20)
            $labNewDir.Text = 'Escriba el nuevo directorio: '
            $formD1.Controls.add($labNewDir)

            $txtNewDir = New-Object
System.Windows.Forms.TextBox
            $txtNewDir.Location = New-Object
System.Drawing.Point(160,90)
            $txtNewDir.Size = New-Object
System.Drawing.Size(100,20)

```

```

        $formD1.Controls.add($txtNewDir)

        $formD1.TopMost = $true
        $resultD1 = $formD1.ShowDialog()

        $modDir = $cbModDir.SelectedItem
        $newDir = $txtNewDir.Text

        $IDUser = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT ID_User FROM usuarios WHERE Nombre = '$query';")
| Select-Object -ExpandProperty ID_User
        Invoke-Sqlcmd -Query "USE proyecto_programacion;
UPDATE Directorios SET Directorio = '$newDir', ID_User = '$IDUser' WHERE
Directorio = '$modDir';"
        Invoke-Sqlcmd -Query "USE proyecto_programacion;
SELECT * FROM directorios;"
    }
}

"Baja de directorios" {
    Write-Host ""
    $form16 = New-Object System.Windows.Forms.Form
    $form16.Text = 'Modificación de la base de datos SQL
SERVER'

    $form16.Size = New-Object System.Drawing.Size(500,300)
    $form16.StartPosition = 'CenterScreen'

    $form16.AcceptButton = $okButton
    $form16.Controls.add($okButton)

    $form16.CancelButton = $cancelButton
    $form16.Controls.add($cancelButton)

    $labExistU = New-Object System.Windows.Forms.Label
    $labExistU.Location = New-Object
System.Drawing.Point(10,20)
    $labExistU.Size = New-Object
System.Drawing.Size(150,20)
    $labExistU.Text = 'Ingrese usuario: '
    $form16.Controls.add($labExistU)

    $txtExistU = New-Object System.Windows.Forms.TextBox
    $txtExistU.Location = New-Object
System.Drawing.Point(160,20)
    $txtExistU.Size = New-Object
System.Drawing.Size(100,20)
    $form16.Controls.add($txtExistU)

    $form16.TopMost = $true
    $result16 = $form16.ShowDialog()

    $userExists = $txtExistU.Text

    $query = (Invoke-Sqlcmd -Query "USE
proyecto_programacion; SELECT Nombre FROM Usuarios WHERE Nombre =
'$userExists'") | Select-Object -ExpandProperty Nombre

    if($result16 -eq
[System.Windows.Forms.DialogResult]::OK -and $query -eq $userExists){
        Write-Host "Se modificará $query"
        $formDE = New-Object System.Windows.Forms.Form
        $formDE.Text = 'Modificación de la base de datos
SQL SERVER'

        $formDE.Size = New-Object
System.Drawing.Size(500,300)
        $formDE.StartPosition = 'CenterScreen'

        $formDE.AcceptButton = $okButton
        $formDE.Controls.add($okButton)

        $formDE.CancelButton = $cancelButton
        $formDE.Controls.add($cancelButton)

        $labEliUser = New-Object
System.Windows.Forms.Label
        $labEliUser.Location = New-Object
System.Drawing.Point(10,20)

```



```

System.Drawing.Size(150,20) $labEliUser.Size = New-Object
directorio: ' $labEliUser.Text = 'Ingrese la ruta del
$formDE.Controls.add($labEliDir)

$txtEliUser = New-Object
System.Windows.Forms.TextBox $txtEliUser.Location = New-Object
System.Drawing.Point(160,20) $txtEliUser.Size = New-Object
System.Drawing.Size(100,20) $formDE.Controls.add($txtEliDir)

$formDE.TopMost = $true
$resultDE = $formDE.ShowDialog()

$eliDir = $txtEliDir.Text

Invoke-Sqlcmd -Query "USE proyecto_programacion;
DELETE FROM Directorios WHERE Nombre = '$eliDir';"
Invoke-Sqlcmd -Query "USE proyecto_programacion;
SELECT * FROM Directorios;"
}
}
}
}
}
}
default{}
}
}
else{
    Write-Host "Adios!"
}

```

### Script Principal de Linux:

```

#!/bin/bash

#Código que realizará las acciones necesarios acorde al proyecto de
programación

#Creación de Menú para realizar las distintas acciones
optionMainMenu=$(dialog --title "Menu principal:" \
    --stdout \
    --menu "¿Qué acción desea realizar?" 0 0 3 \
        1 "Iniciar el programa" \
        2 "Edición de usuarios" \
        3 "Edición de directorios" \
        4 "Salir")
echo "$optionMainMenu"

#Variables para el ingreso a mysql
userMysql="root"
password="Yuihirasawa1"

#Instancia Case acorde a lo seleccionado en el Menú anterior
case $optionMainMenu in

```

```

#En caso de selecciona la opción 1. Iniciar programa se hace:
1)
    #Menú donde se ingresará el nombre del usuario, se sacarán los
    usuarios de la base de datos y se guardaran en un variable (resultado
    esperado 1 usuario)
    userDialog=$(dialog --title "Iniciando el programa" \
        --stdout \
        --inputbox "Ingresa el nombre del usuario" 0
0)

    sql="SELECT usuario FROM usuarios WHERE usuario = '$userDialog'"
    i=0
    while IFS=$'\t' read usuariosql
    do
        USUARIOSQL[$i]=$usuariosql
        ((i++))
    done < <(mysql -u $userMysql -p$password proyecto_programacion -
e "$sql;")

    echo "El usuario ingresado es: $userDialog"
    echo
    echo "El usuario de la base de datos es: ${USUARIOSQL[1]}"

    #Se guardará el usuario dentro de la variable userConfirm, la
    razón por la cual es [1] es porque [0] es el nombre de columna
    userConfirm=${USUARIOSQL[1]}

    #Si el usuario ingresado coincide con el que está en la Base de
    Datos mostrará el siguiente Menú
    if [ $userDialog == $userConfirm ]
    then
        echo "Usuario existe y es: $userDialog"
        optionMenu1=$(dialog --title "Inicio del programa" \
            --stdout \
            --menu "Selecciona una opción:" 0 0 3 \
                1 "Subir archivos" \
                2 "Crear directorios" \
                    3 "Listar procesos" \
                    4 "Detener procesos" \
                    5 "Listar archivos y Directorios" \
                        6 "Enviar archivos..." \
                        7 "Enviar mensaje" \
                        8 "Revisar mensajes" \
                        9 "Salir" )
        echo "La opción elegida fueeee: $optionMenu1"

        #Otro case para este Menú
        case $optionMenu1 in
#En caso de seleccionar la opción 1. Subir Archivos
1)

```

```

#Mostrará ventana donde te dejará seleccionar
archivos de Linux y poder subirlos a Windows
echo "Subir archivos"
optionMenu1Sub1=$(dialog --title "Selecciona el
archivo que deseas subir" \
--stdout \
--fselect "/home/benjamin/" 14 70 )

#En caso de que el archivo exista mostrará un Menú
yesno para confirmar la subida
if [ -f "$optionMenu1Sub1" ]
then
    dialog --title "Se subirá el archivo
${optionMenu1Sub1}" \
--yesno "¿Estás seguro?" 0 0

#Guardar solamente la última parte de la liga
completa de la ubicación de un archivo
archivoSinliga=$(echo "$optionMenu1Sub1" | rev |
cut -d'/' -f 1 | rev)

#Se copia el archivo a la carpeta compartida
cp $optionMenu1Sub1
/home/benjamin/carpetaCompartida/tmp/$archivoSinliga

#Se saca la id del usuario para poder saber los
directorios:
idUser="SELECT id_user FROM usuarios WHERE
usuario = '$userConfirm'"

j=0
while IFS=$'\t' read ids
do
    IDUSUARIO[$j]=$ids
    ((j++))

done < <(mysql -u $userMysql -p$password
proyecto_programacion -e "$idUser;")

echo "La id del usuario es: ${IDUSUARIO[1]}"

#Se sacan los directorios que se encuentran en la
base de datos:
sql="SELECT directorio FROM directorios INNER
JOIN usuarios ON usuarios.id_user = directorios.id_user WHERE
directorios.id_user = ${IDUSUARIO[1]}"
i=0
while IFS=$'\t' read directoriosSql
do

```

```

DIRECTORIOSQL[$i]="$i
$directoriosSql"
        ((i++))
        done < <(mysql -u $userMysql -p$password
proyecto_programacion -e "$sql;" | tail -n +2)

        #Mostrará un menú con los directorios a donde se
podrá subir el archivo
        directorioSeleccionado=$(dialog --title
"Seleccionar directorio" \
                                --stdout \
                                --menu "Selecciona el
directorio:" 0 0 0 \
                                ${DIRECTORIOSQL[@]})

        #Quitar número de DIRECTORIOSQL
                                directorioDestino=$(echo
"${DIRECTORIOSQL[$directorioSeleccionado]}" | cut -d ' ' -f2)

        directorioWindows=$(echo $directorioDestino | tr
 '/' '\\')
        #Aquí va lo que irá dentro del archivo que leerá
powershell
        sharedFolder=$(echo
"//192.168.163.131/SAMBA/tmp/$archivoSinliga" | tr '/' '\\')
        echo "Move-Item -Path $sharedFolder -Destination
$directorioWindows -Force" >
/home/benjamin/carpetaCompartida/tasks/comandos.ps1

        mv
/home/benjamin/carpetaCompartida/tasks/band.txt
/home/benjamin/carpetaCompartida/tasks/bandera.txt

        fi
        ;;
#Inicio del punto 2

        2)

                echo "Crear directorios"
                #Mostrará una ventana donde nos permitirá escribir el
nombre del nuevo directorio
                folderDialog=$(dialog --title "Creando dirrectorios"
\
                                --stdout \
                                --inputbox "Ingresa el
nombre del nuevo directorio que deseas crear:" 0 0)

                #Se saca la id del usuario para poder saber los
directorios:

```

```

idUser="SELECT id_user FROM usuarios WHERE usuario =
'$userConfirm'"

j=0
while IFS=$'\t' read ids
do
    IDUSUARIO[$j]=$ids
    ((j++))
done <<(mysql -u $userMysql -p$password
proyecto_programacion -e "$idUser;")

echo "La id del usuario es: ${IDUSUARIO[1]}"

#Se sacan los directorios que se encuentran en la
base de datos:

sql="SELECT directorio FROM directorios INNER JOIN
usuarios ON usuarios.id_user = directorios.id_user WHERE
directorios.id_user = ${IDUSUARIO[1]}"

i=0
while IFS=$'\t' read directoriosSql
do
    DIRECTORIOSQL[$i]=$i
$directoriosSql"

    ((i++))
done <<(mysql -u $userMysql -p$password
proyecto_programacion -e "$sql;" | tail -n +2)

echo "${DIRECTORIOSQL[0]}"
echo "${DIRECTORIOSQL[1]}"
echo "${DIRECTORIOSQL[2]}"

#Mostrará un menú con los directorios a donde se
podrá crear el directorio
directorioSeleccionado=$(dialog --title "Seleccionar
directorio" \
    --stdout \
    --menu "Selecciona el directorio
donde quieres insertar dicho directorio:" 0 0 0 \
    ${DIRECTORIOSQL[@]})

#Quitar número de DIRECTORIOSQL
directorioDestino=$(echo
"${DIRECTORIOSQL[$directorioSeleccionado]}" | cut -d ' ' -f2)

#Código donde mandará el código para que powershell
cree el directorio en el directorio seleccionado
echo "$folderDialog se creará en $directorioDestino"

```

```

#Mandar señal para que el directorio se cree en el
directorio destino
    rutaFinal=$(echo "$directorioDestino" | tr '/' '\\')
    echo "New-Item -Path $rutaFinal -Name $folderDialog -
ItemType Directory -Force" > carpetaCompartida/tasks/comandos.ps1
    mv /home/benjamin/carpetaCompartida/tasks/band.txt
/home/benjamin/carpetaCompartida/tasks/bandera.txt

;;

#Inicio de la opción 3

3)
    echo "Listar procesos"
    #Solicitar un archivo que posea los procesos que se
estén ejecutando en Windows, dicho archivo estará ubicado en la carpeta
compartida

    #Una vez creado el archivo en
carpetaCompartida/tmp/procesos.txt
    i=0
    while IFS= read -r line
    do
        echo "$1 $line"
        PROCESOS[$i]="$i $line"
        ((i++))
    done <
/home/benjamin/carpetaCompartida/tmp/procesos.txt

    #Ventana que permite observar los procesos en
ejecución
    #salida=$(dialog --title "Procesos de Windows en
Ejecución:" \
    #      --stdout \
    #      --menu "Procesos:" 0 0 0 \
    #      ${PROCESOS[@]})

    #Quitar número de PROCESOS
    procesosBuenos=$(echo
"${PROCESOS[$salida]}" | cut -d ' ' -f2)

    echo "El proceso seleccionado fue: $procesosBuenos"
    ;;

4)

#inicio de la opción 4

    echo "Detener procesos"

```

```

        #Solicitar un archivo que posea los procesos que se
estén ejecutando en Windows, dicho archivo estará ubicado en la carpeta
compartida

        echo "(Get-Process | Select-Object -Property
ProcessName) | Export-CSV -Path //192.168.163.131/samba/tmp/buffer.csv -
Force" | tr '/' '\\' > carpetaCompartida/tasks/comandos.ps1
        mv /home/benjamin/carpetaCompartida/tasks/band.txt
/home/benjamin/carpetaCompartida/tasks/bandera.txt

        sleep 5

                                #Una vez creado el archivo en
carpetaCompartida/tmp/buffer.

                                i=0
                                while IFS="," read -r line
                                do
                                        echo "$i $line"

                                                PROCESOS[$i]="$i $line"
                                                ((i++))
                                done < <(tail -n +3
/home/benjamin/carpetaCompartida/tmp/buffer.csv | perl -p -i -e
"s/\r//g")

                                #Ventana que permite observar los
procesos en ejecución

                                salida=$(dialog --title "Procesos
de Windows en Ejecución:" \

                                                --stdout \
                                                --menu "Proceso a
matar:" 0 0 0 \

                                                ${PROCESO
S[@]})

                                #Quitar número de PROCESOS
procesoBueno=$(echo
"${PROCESOS[$salida]}" | cut -d ' ' -f 2)

                                echo "El proceso seleccionado
fue: $procesoBueno"

                                #mandar a matar a $procesoBueno
                                #echo "Stop-Process -Name $procesoBueno" >
/home/benjamin/carpetaCompartida/tasks/comandos.ps1
                                #echo "1" >
/home/benjamin/carpetaCompartida/tasks/bandera.txt

                                ;;

```



```

5)
#Inicio de la opción 5, Listar archivos
#Se saca la id del usuario para
poder saber los directorios:
idUser="SELECT id_user FROM
usuarios WHERE usuario = '$userConfirm'"
j=0
while IFS=$'\t' read ids
do
    IDUSUARIO[$j]=$ids
    ((j++))
done < <(mysql -u $userMysql -
p$password proyecto_programacion -e "$idUser;")

echo "La id del usuario es:
${IDUSUARIO[1]}"

echo "Listar Archivos"

#Seleccionamiento de directorios de $userConfirm
sql="SELECT directorio FROM directorios INNER JOIN
usuarios ON usuarios.id_user = directorios.id_user WHERE
directorios.id_user = ${IDUSUARIO[1]}"
i=0
while IFS=$'\t' read
directoriosSql
do
    DIRECTORIOSQL[$i]="$i
$directoriosSql"
    ((i++))
done < <(mysql -u $userMysql -
p$password proyecto_programacion -e "$sql;" | tail -n +2)

#Mostrará un menú con los
directorios a los cuales se podrán listar sus archivos
directorioListar=$(dialog --title
"Seleccionar directorio" \
-
-stdout \
-
-menu "Selecciona el directorio donde quieres insertar dicho directorio:"
0 0 0 \

${DIRECTORIOSQL[@]})

#Quitar número de DIRECTORIOSQL

```

```

                                directorioDestino=$(echo
"${DIRECTORIOSQL[$directorioListar]}" | cut -d ' ' -f2)

                                #Mandar señal a Windows para que saque los archivos
de directorioDestino
                                rutaFinal=$(echo "$directorioDestino" | tr '/' '\\')

                                (echo "(Get-Childitem $rutaFinal | Select-Object -
Property Mode, Length, Name) | Export-CSV -Path
//192.168.163.131/samba/tmp/buffer.csv -force") | tr '/' '\\ ' >
carpetaCompartida/tasks/comandos.ps1
                                mv /home/benjamin/carpetaCompartida/tasks/band.txt
/home/benjamin/carpetaCompartida/tasks/bandera.txt

                                echo "Los archivos de $rutaFinal son:"

                                while IFS="," read -r mode length name
                                do
                                        echo "$mode $length $name"
                                done < <(tail -n +3 carpetaCompartida/tmp/buffer.csv)
                                                                ;;

                                6)

#Inicio de la opción 6

                                echo "Enviar archivos"
                                #Mostrará ventana donde te dejará
seleccionar archivos de Linux y poder subirlos a Windows
                                echo "Subir archivos"
                                optionMenu1Sub1=$(dialog --title
"Selecciona el archivo que deseas subir" \
                                                                --stdout
\
                                                                --fselect
"/home/benjamin/" 14 70 )

                                #En caso de que el archivo exista
mostrará un Menú yesno para confirmar la subida
                                if [ -f "$optionMenu1Sub1" ]
                                then
                                        dialog --title "Se subirá
el archivo ${optionMenu1Sub1}" \
                                                                --yesno "¿Estás
seguro?" 0 0

                                #Guardar solamente la
última parte de la liga completa de la ubicación de un archivo
                                archivoSinliga=$(echo
"$optionMenu1Sub1" | rev | cut -d '/' -f 1 | rev)

```

```

#Se copia el archivo a la
carpeta compartida
cp $optionMenu1Sub1
/home/benjamin/carpetaCompartida/$archivoSinliga

#Se saca la id del
usuario para poder saber los directorios:
idUser="SELECT id_user
FROM usuarios WHERE usuario = '$userConfirm'"

j=0
while IFS=$'\t' read ids
do
    IDUSUARIO[$j]=$ids
    ((j++))
done < <(mysql -u
$userMysql -p$password proyecto_programacion -e "$idUser;")

echo "La id del usuario es: ${IDUSUARIO[1]}"

#Se sacan los directorios
que se encuentran en la base de datos:
sql="SELECT directorio
FROM directorios INNER JOIN usuarios ON usuarios.id_user =
directorios.id_user WHERE directorios.id_user = ${IDUSUARIO[1]}"
i=0
while IFS=$'\t' read
directoriosSql
do
    DIRECTORIOSQL[$i]
    ="$i $directoriosSql"
    ((i++))
done < <(mysql -u
$userMysql -p$password proyecto_programacion -e "$sql;" | tail -n +2)

#Mostrará un menú con los
directorios a donde se podrá subir el archivo
directorioSeleccionado=$(
dialog --title "Seleccionar directorio" \
--stdout \
--menu "Selecciona el directorio:" 0 0 0 \
${DIRECTORIOSQL[@]})

```

```

                                #Quitar número de
DIRECTORIOSQL
                                directorioDestino=$(echo
"${DIRECTORIOSQL[$directorioSeleccionado]}" | cut -d ' ' -f2)

                                #Aqui va lo que irá
dentro del archivo que leerá powershell
                                echo "Los archivos se
enviarán a: $directorioDestino"

                                fi

                                ;;

7)

#Inicio de la opción 7

                                echo "Enviar mensaje"
                                mensaje=$(dialog --title "Enviar
Mensajes" \
                                --stdout \
                                --inputbox
"Escriba su mensaje" 0 0)

                                echo "$mensaje \t" >>
/home/benjamin/carpetaCompartida/msg/mensajes.txt

                                ;;

8)

                                echo "Revisar mensajes"
                                opcionMsg=$(dialog --title "Recibir mensajes:" \
--yesno)

                                ;;

*)

                                echo "..."
                                ;;

esac

else

                                echo "Usuario no existe en la base de datos"

fi

;;

2)

#Menú para lo que desee hacer con los usuarios: (Modificar base
de datos y en equipo)

                                echo "Editando usuarios..."
                                optionMenu2=$(dialog --title "Modificación de usuarios:" \
--stdout \

```

```

        --menu "Seleccionar una acción" 0 0 0 \
            1 "Dar de alta un usuario" \
            2 "Modificar usuario" \
            3 "Eliminar un usuario" \
            4 "Salir" )

case $optionMenu2 in
    1)
        #Dar de alta a un usuario en MySQL
        datosAltaMysql=$(dialog --title "Alta de usuarios de
MySQL" \

            --separate-widget $" \n" \
            --form "Introduce los datos del usuarios" \
            0 0 0 \
            "Nombre:" 1 1 "$nombre" 1 10 20 0 \
            "Llave publica:" 2 1 "$llavePub" 2 10 20 0 \
            "Llave privada:" 3 1 "$llavePri" 3 10 20 0 \
            3>&1 1>&2 2>&3 3>&-)

        #Consulta
        nombre=$(echo "$datosAltaMysql" | sed -n 1p)
        llavePub=$(echo "$datosAltaMysql" | sed -n 2p)
        llavePri=$(echo "$datosAltaMysql" | sed -n 3p)
        insertarSQL="INSERT INTO usuarios (usuario,
llave_publica, llave_privada) VALUES ('$nombre', '$llavePub',
'$llavePri')"

        mysql -u $userMysql -p$password proyecto_programacion -e
"$insertarSQL;"

        ;;
    2)
        #Modificar un usuario en MySQL
        #Pedir un usuario
        userDialog=$(dialog --title "Iniciando el programa" \
            --stdout \
            --inputbox "Ingresa el nombre del usuario" 0
0)

        sql="SELECT usuario FROM usuarios WHERE usuario =
'$userDialog'"

        i=0
        while IFS=$'\t' read usuariosql2
        do
            USUARIOSQL[$i]=$usuariosql2
            ((i++))
        done < <(mysql -u $userMysql -p$password
proyecto_programacion -e "$sql;")

```

```

        echo "El usuario ingresado es: $userDialog"
        echo ""
        echo "El usuario de la base de datos es:
${USUARIOSQL[1]}"

        #Se guardará el usuario dentro de la variable
userConfirm, la razón por la cual es [1] es porque [0] es el nombre de
columna

        userConfirm=${USUARIOSQL[1]}

        echo "$userConfirm"
        if [ "$userConfirm" == "$userDialog" ]
        then
            datosModMysql=$(dialog --title "Modificacion de
usuarios de MySQL" \
                                --separate-widget
                                "$"\n" \
                                --form "Introduce los
datos del usuarios" \
                                0 0 0 \
                                "Nombre:" 1 1
                                "Llave publica:" 2 1
                                "Llave privada:" 3 1
                                3>&1 1>&2 2>&3 3>&- )

            #Consulta
            nombre=$(echo "$datosModMysql" | sed
-n 1p)
            llavePub=$(echo "$datosModMysql" |
sed -n 2p)
            llavePri=$(echo "$datosModMysql" |
sed -n 3p)

            actualizarSQL="UPDATE usuarios SET
usuario='$nombre', llave_publica='$llavePub', llave_privada='$llavePri'
WHERE usuario = '$userDialog'"

            mysql -u $userMysql -p$password
proyecto_programacion -e "$actualizarSQL;"

        fi

        ;;

3)
        #Eliminar un Usuario de MySQL
        #Eliminar un directorio en MYSQL

```

```

#Mostrará una ventana donde nos permitirá
escribir el nombre del nuevo directorio
userConfirm=$(dialog --title "Eliminación
de directorio" \
                                --stdout \
                                --inputbox "Ingresa el
nombre del usuario:" 0 0)

#Se saca la id del usuario para poder
saber los directorios:
idUser="SELECT id_user FROM usuarios
WHERE usuario = '$userConfirm'"
j=0
while IFS=$'\t' read ids
do
    IDUSUARIO[$j]=$ids
    ((j++))
done < <(mysql -u $userMysql -p$password
proyecto_programacion -e "$idUser;")

echo "La id del usuario es:
${IDUSUARIO[1]}"
userDelete="${IDUSUARIO[1]}"

dialog --title "Eliminacion de Usuario" \
        --yesno "¿Estás seguro?" 0 0
ans=$?
if [ $ans -eq 0 ]
then
    mysql -u $userMysql -
p$password proyecto_programacion -e "DELETE FROM usuarios WHERE
id_user='$userDelete';"
else
    echo "Ta bien"
fi
;;

*)
echo "Saliendo..."
;;
esac
;;

3)
echo "Editando directorios..."
optionMenu3=$(dialog --title "Modificación de directorios:" \
                    --stdout \

```



```

                                --menu "Seleccionar una acción" 0
0 0 \
                                1 "Dar de alta un
directorio" \
                                2 "Modificar directorio"
\
                                3 "Eliminar un
directorio" \
                                4 "Salir" )

    case $optionMenu3 in
        1)
            #Dar de alta un directorio en MySQL
            datosAltaDirMysql=$(dialog --title "Alta de usuarios de
MySQL" \
                                --separate-widget $" \n" \
                                --form "Introduce los datos del
usuarios" \
                                0 0 0 \
                                "Directorio:" 1 1 "$directorio" 1
10 20 0 \
                                "ID_User:" 2 1 "$iduser" 2 10 20
0 \
                                3>&1 1>&2 2>&3 3>&-)

            #Consulta
            directorio=$(echo "$datosAltaDirMysql" |
sed -n 1p)
            iduser=$(echo "$datosAltaDirMysql" | sed
-n 2p)

            insertarDSQL="INSERT INTO directorios
(directorio, id_user) VALUES ('$directorio', '$iduser')"

            mysql -u $userMysql -p$password proyecto_programacion -e
"$insertarDSQL;"
;;
        2)
            #Modificar directorio
            userDialog=$(dialog --title "Iniciando el
programa" \
                                --stdout \
                                --inputbox "Ingresa el nombre del
usuario" 0 0)

            sql="SELECT usuario FROM usuarios WHERE
usuario = '$userDialog'"

            i=0
            while IFS=$'\t' read usuariosql
            do

```

```

                                USUARIOSQL[$i]=$usuariosql
                                ((i++))
                                done < <(mysql -u $userMysql -p$password
proyecto_programacion -e "$sql;")

                                echo "El usuario ingresado es:
$userDialog"

                                echo ""
                                echo "El usuario de la base de datos es:
${USUARIOSQL[1]}"

                                #Se guardará el usuario dentro de la
                                variable userConfirm, la razón por la cual es [1] es porque [0] es el
                                nombre de columna

                                idConfirm="${USUARIOSQL[1]}"

                                userConfirm=$(mysql -u $userMysql -p$password
proyecto_programacion -e "Select id_user FROM directorios WHERE id_user =
'$idConfirm';" | tail -n +2)

                                if [ "$userConfirm" == "$userDialog" ]
                                then

                                directorioListar=$(dialog --title
"Seleccionar directorio" \
                                -
                                -stdout \
                                -
                                -menu "Selecciona el directorio donde quieres insertar dicho directorio:"
0 0 0 \

                                ${DIRECTORIOSQL[@]})

                                #Quitar número de DIRECTORIOSQL
                                directorioFinal=$(echo
"${DIRECTORIOSQL[$directorioListar]}" | cut -d ' ' -f2)

                                datosModMysqlD=$(dialog --title "Modificacion de
directorios de MySQL" \
                                --separate-widget "$\n" \
                                --form "Introduce los
datos del directorio" \
                                0 0 0 \
                                "Directorio:" 1 1
"$nombre" 1 10 20 0 \

                                3>&1 1>&2 2>&3 3>&-)

```

```

#Consulta
direc=$(echo "$datosModMySQLD" |
sed -n 1p)

actualizarDSQL="UPDATE directorios
SET directorio='$direc', id_user='$idser' WHERE
directorio='$disrectorioFinal';"

mysql -u $userMysql -p$password
proyecto_programacion -e "$actualizarDSQL"

fi

;;

3)

#Eliminar un directorio en MYSQL
#Mostrará una ventana donde nos permitirá escribir el
nombre del nuevo directorio
userConfirm=$(dialog --title "Eliminación de directorio"
\
--stdout \
--inputbox "Ingresa el nombre
del usuario:" 0 0)

#Se saca la id del usuario para poder saber los
directorios:
idUser="SELECT id_user FROM usuarios WHERE usuario =
'$userConfirm'"
j=0
while IFS=$'\t' read ids
do
IDUSUARIO[$j]=$ids
((j++))
done < <(mysql -u $userMysql -p$password
proyecto_programacion -e "$idUser;")

echo "La id del usuario es: ${IDUSUARIO[1]}"

#Se sacan los directorios que se encuentran en la base de
datos:
sql="SELECT directorio FROM directorios INNER JOIN
usuarios ON usuarios.id_user = directorios.id_user WHERE
directorios.id_user = ${IDUSUARIO[1]}"
i=0
while IFS=$'\t' read directoriosSql
do
DIRECTORIOSQL[$i]="$i $directoriosSql"
((i++))

```

```

done < <(mysql -u $userMysql -p$password
proyecto_programacion -e "$sql;" | tail -n +2)

#echo "${DIRECTORIOSQL[0]}"
#echo "${DIRECTORIOSQL[1]}"
#echo "${DIRECTORIOSQL[2]}"

#Mostrará un menú con los directorios de los cuales se
podrán eliminar el directorio
directorioSeleccionado=$(dialog --title "Seleccionar
directorio" \
                                --stdout \
                                --menu "Selecciona el directorio donde
quieres insertar dicho directorio:" 0 0 0 \
                                ${DIRECTORIOSQL[@]})

#Quitar número de DIRECTORIOSQL
directorioDestino=$(echo
"${DIRECTORIOSQL[$directorioSeleccionado]}" | cut -d ' ' -f2)

dialog --title "Eliminacion de Directorio" \
        --yesno "¿Estás seguro?" 0 0
ans=$?
if [ $ans -eq 0 ]
then
    mysql -u $userMysql -
p$password proyecto_programacion -e "DELETE FROM directorios WHERE
directorio='$directorioDestino';"
else
    echo "ta bien"
fi
;;
*)
echo "Saliendo..."
;;
esac
;;

*)
echo "Saliendo..."
dialog --title "Adios!" \
        --infobox "Hasta luego..." 0 0
;;
esac

```

### Script JOB de powershell:

```
Start-Job -ScriptBlock {  
    while((Test-Path '\\192.168.163.131\samba\tasks\existe.txt') -eq $true){  
        if((Test-Path '\\192.168.163.131\samba\tasks\bandera.txt') -eq  
"$true"){  
            Write-Host "Entre a trabajar"  
            \\192.168.163.131\samba\tasks\comandos.ps1 | Invoke-Expression  
            start-sleep -Seconds 3  
            Write-Host "Cambié a 0 la bandera"  
            Rename-Item -Path '\\192.168.163.131\samba\tasks\bandera.txt' -  
NewName '\\192.168.163.131\samba\tasks\band.txt' -Force  
            Start-Sleep 1  
        }  
    }  
}
```

### Script 'Job' de Bash:

```
#!/bin/bash  
  
perl -p -i -e "s/\r//g"  
/home/benjamin/carpetaCompartida/tasks/bandera.txt  
existe="/home/benjamin/carpetaCompartida/tasks/existe.txt"  
bandera=$(cat /home/benjamin/carpetaCompartida/tasks/bandera.txt)  
  
cd /home/benjamin/carpetaCompartida/tasks  
  
while(test -e $existe)  
do  
    perl -p -i -e "s/\r//g"  
/home/benjamin/carpetaCompartida/tasks/bandera.txt  
    bandera=$(cat /home/benjamin/carpetaCompartida/tasks/bandera.txt)  
  
    sleep 1  
    if [ $bandera == "1" ]  
    then  
  
        ./comandos.sh  
  
        echo "0" > /home/benjamin/carpetaCompartida/tasks/bandera.txt  
    fi  
done  
  
echo "Se murio 'existe'"
```