**S.I.G.A.T.**

**Sistemas Operativos III**SURU

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ROL** | **APELLIDO** | **NOMBRE** | **C.I.** | **E-MAIL** | **TEL/CEL** |
| Coordinador | Ribeiro | Alex | 5.499.977-6 | alexleonelribeiro@gmail.com | 095220022 |
| Sub-Coordinador | Troche | Gastón | 5.099.726-3 | gtroche12@gmail.com | 094050655 |
| Integrante 1 | Barreira | Santiago | 5.148.730-2 | Santii.bf@gmail.com | 093495491 |
| Integrante 2 | Fernández | Pedro | 5.154.832-8 | Pedro.nacho997@gmail.com | 091358982 |
| Integrante 3 | Cinelli | Germán | 4.664.131-3 | ger.cs28@gmail.com | 096506169 |

**Docente:** De Los Santos, Andy.

|  |
| --- |
| Fecha de Culminación:  31/10/2016 |

**TERCERA ENTREGA**

Indice 1, 2

1.1. ABMC Usuario (MenuPrincipal.sh) 3

1.2. ABMC Usuario (MenuPrincipal.sh) 4

1.3. ABMC Usuario (MenuPrincipal.sh) .5

1.4. ABMC Usuario (MenuSalir.sh) .6

1.5. ABMC Usuario (MenuSalir.sh) .7

1.6. ABMC Usuario (CearUsuario.sh) .8

1.7. ABMC Usuario (CearUsuario.sh) .9

1.8. ABMC Usuario (ModificarUsuario.sh) .10

1.9. ABMC Usuario (ModificarUsuario.sh) .11

1.10. ABMC Usuario (ModificarUsuarioNombre.sh) .12

1.11. ABMC Usuario (EliminarUsuario.sh) .13

1.12. ABMC Usuario (BloquearDesbloquear.sh) .14

1.13. ABMC Usuario (BloquearUsuario.sh) .15

1.14. ABMC Usuario (DesbloquearUsuario.sh) .16

1.15. ABMC Usuario (ConsultaUsuario.sh) .17

1.16. ABMC Usuario (ModificarUsuarioLogin.sh) .18

1.17. ABMC Usuario (ModificarUsuarioG.sh) .19

1.18. ABMC Usuario (ModificarUsuarioGP.sh) .20

1.19. ABMC Usuario (ModificarUsuarioComentario.sh) .21

1.20. ABMC Usuario (BloquearDesbloquear.sh) .22

1.21. ABMC Usuario (BloquearUsuario.sh) .23

1.22. ABMC Usuario (DesbloquearUsuario.sh) .24

1.23. ABMC Usuario (ModificarUsuario2.sh) .25

1.24. ABMC Usuario (ModificarUsuario2.sh) .26

1.25. ABMC Usuario (ModificarUsuarioUID.sh) .27

1.26. ABMC Usuario (ModificarUsuarioHome.sh) .28

1.27. ABMC Usuario (ModificarUsuarioShell.sh) .29

1.28. ABMC Usuario (ModificarUsuarioClave.sh) .30

1.29. ABMC Usuario (MenuGrupo.sh) .31

1.30. ABMC Usuario (MenuGrupo.sh) .32

1.31. ABMC Usuario (CrearGrupo.sh) .33

1.32. ABMC Usuario (CrearGrupo.sh) .34

1.33. ABMC Usuario (ModificarGrupo.sh) .35

1.34. ABMC Usuario (ModificarGrupoGID.sh) .36

1.35. ABMC Usuario (ModificarGrupoNombre.sh) .37

1.36. ABMC Usuario (EliminarGrupo.sh) .38

1.37. ABMC Usuario (ConsultaGrupo.sh) .39

2.1. Capturas (ABMC Usuario-Grupo) .40

2.2. Capturas (ABMC Usuario-Grupo) .41

2.3. Capturas (ABMC Usuario-Grupo) .42

2.4. Capturas (ABMC Usuario-Grupo) .43

2.5. Capturas (ABMC Usuario-Grupo) .44

2.6. Capturas (ABMC Usuario-Grupo) .45

2.7. Capturas (ABMC Usuario-Grupo) .46

2.8. Capturas (ABMC Usuario-Grupo) .47

3.1. ABMC (Animales) .48

3.2. ABMC (Animales) .49

3.3. ABMC (Animales) .50

3.4. ABMC (Animales) .51

3.5. ABMC (Animales) 52

3.6. ABMC (Animales) .53

3.7. ABMC (Animales) .54

3.8. ABMC (Animales) .55

3.9. ABMC (Animales) .56

3.10. ABMC (Animales) .57

3.11. ABMC (Animales) .58

3.12. ABMC (Animales) .59

3.13. ABMC (Animales) .60

3.14. ABMC (Animales) .61

3.15. ABMC (Animales) .62

3.16. ABMC (Animales) .63

3.17. ABMC (Animales) .64

3.18. ABMC (Animales) .65

4.1. Tutorial instalación máquina virtual .66

4.2. Tutorial instalación máquina virtual .67

4.3. Tutorial instalación máquina virtual .68

4.4. Tutorial instalación máquina virtual .69

4.5. Tutorial instalación máquina virtual .70

4.6. Tutorial instalación máquina virtual .71

4.7. Tutorial instalación máquina virtual .72

4.8. Tutorial instalación máquina virtual .73

2.4. Comando crontab (Respaldo) .74

2.5. Comando crontab (Respaldo) .75

ABMC Usuarios-Grupos

MenuPrincipal.sh

#!/bin/bash

#MenuPrincipal

while [ $m!=3 ]

do

clear

echo -e " \e[0;34m\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\e[0m"

echo -e "\e[0;34m|\e[0m" "\tMenu Principal\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯|\e[0m"

echo -e "\e[0;34m|\e[0m" "\t[1] Usuario\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" "\t[2] Grupo\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" "\t[3] Salir\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\e[0m"

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Opcion: "

read m

case $m in

1). MenuUsuario.sh

;;

2). MenuGrupo.sh

;;

3). MenuSalir.sh

;;

\*)echo -e "\n\e[0;31m [x] Opcion incorrecta...\e[0m"

sleep 2

clear

. MenuPrincipal.sh

;;

esac

done

MenuUsuario.sh

#!/bin/bash

#MenuUsuario

while [ m!=6 ]

do

clear

echo -e " \e[0;34m\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\e[0m"

echo -e "\e[0;34m|\e[0m" "\t USUARIOS\t " " \e[0;34m|\e[0m"

echo -e "\e[0;34m|¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯|\e[0m"

echo -e "\e[0;34m|\e[0m" " [1] Crear un Usuario\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [2] Modificar un Usuario" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [3] Eliminar un Usuario" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [4] Bloquear/Desbloquear" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [5] Consultas\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [6] Atras\t\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\e[0m"

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Opcion :"

read m

case $m in

1). CrearUsuario.sh

;;

2). ModificarUsuario.sh

;;

3). EliminarUsuario.sh

;;

4). BloquearDesbloquear.sh

;;

5). ConsultaUsuario.sh

;;

6)break

;;

\*)echo -e "\n\e[0;31m [x] Opcion incorrecta...\e[0m"

sleep 2

clear

;;

esac

done

MenuSalir.sh

#!/bin/bash

#MenuSalir

while [ m!=2 ]

do

clear

echo -e "\e[1;32mEsta seguro de que desea salir?\n\e[0m"

echo -e " [1] Si"

echo -e " [2] No"

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Opcion :"

read m

case $m in

1)

clear

echo -e -n "\e[1;36mCerrando sesion\e[0m"

sleep 1

clear

echo -e -n "\e[1;36mCerrando sesion.\e[0m"

sleep 1

clear

echo -e -n "\e[1;36mCerrando sesion..\e[0m"

sleep 1

clear

echo -e -n "\e[1;36mCerrando sesion...\e[0m"

sleep 2

exit

;;

2)

clear

echo -e -n "\e[0;34m>\e[0m" "\e[1;34mRedireccionando a Menu Principal...\e[0m"

sleep 2

break

;;

\*)echo -e "\n\e[0;31m [x] Opcion incorrecta...\e[0m"

sleep 2

clear

;;

esac

done

CrearUsuario.sh

#!/bin/bash

#CrearUsuario

clear

echo -e "\e[1;34m \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\e[0m"

echo -e "\e[1;34m|\e[0m" "\e[0;33m CREACION DE USUARIOS\e[0m" " \e[1;34m|\e[0m"

echo -e "\e[1;34m ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯\e[0m"

echo -e

sleep 2

echo -e -n "\e[1;33m Ingrese los siguientes datos...\e[0m"

sleep 3

clear

echo -e "\e[1;34m \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\e[0m"

echo -e "\e[1;34m|\e[0m" "\e[0;33m CREACION DE USUARIOS\e[0m" " \e[1;34m|\e[0m"

echo -e "\e[1;34m ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯\e[0m"

echo -e -n "\n\e[0;32m>\e[0m" "Usuario: "

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" == "" ]

then

echo -e -n "\n\e[0;32m>\e[0m" "Nombre: "

read n

echo -e -n "\n\e[0;32m>\e[0m" "Apellido: "

read a

useradd -d /home/$Usuario -mk /etc/skel -c "$n $a" -s /bin/bash -g "100" $Usuario

if [ $? == 0 ]

then

passwd $Usuario

sleep 2

clear

echo -e -n "\e[1;36mEspere\e[0m"

sleep 1

clear

echo -e -n "\e[1;36mEspere.\e[0m"

sleep 1

clear

echo -e -n "\e[1;36mEspere..\e[0m"

sleep 1

clear

echo -e -n "\e[1;36mEspere...\e[0m"

sleep 2

clear

echo -e -n "\e[0;32m> Usuario creado exitosamente. \e[0m"

sleep 2

clear

echo -e -n "\e[0;34m>\e[0m" "\e[1;34m@info del Usuario: \n\e[0m"

finger $Usuario

echo -e -n "\e[1;34m> Presione Enter para volver... \e[0m"

read x

else

echo -e -n "\n\e[0;31m [x] Se produjo un error...\e[0m"

sleep 2

fi

else

echo -e -n "\n\e[0;31m [x] El nombre de usuario ya existe...\e[0m"

sleep 2

clear

fi

ModificarUsuario.sh

#!/bin/bash

#ModificarUsuario

while [ m!=5 ]

do

clear

echo -e " \e[0;34m\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\e[0m"

echo -e "\e[0;34m|\e[0m" " Modificar Usuario" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯|\e[0m"

echo -e "\e[0;34m|\e[0m" " [1] Nombre de Usuario" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [2] Grupo\t\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [3] Comentario\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [4] Bloquear/Desbloquear" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [5] Opciones Avanzadas" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [6] Atras\t\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\e[0m"

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Opcion :"

read m

case $m in

1). ModificarUsuarioLogin.sh

;;

2). ModificarUsuarioG.sh

;;

3). ModificarUsuarioComentario.sh

;;

4). BloquearDesbloquear.sh

;;

5). ModificarUsuario2.sh

;;

6)break

;;

\*)echo -e "\n\e[0;31m [x] Opcion incorrecta...\e[0m"

sleep 2

clear

;;

esac

done

ModificarUsuarioNombre.sh

#ModificarUsuarioNombre

clear

echo "Ingrese nombre de Usuario: "

read Usuario

Usuario1=$(grep "$Usuario" /etc/passwd)

if [ "$Usuario1" != "" ]

then

echo "Ingrese nuevo nombre: "

read nom

usermod $Usuario -l $nom

else

echo " [x] El nombre de usuario no existe..."

sleep 2

clear

. ModificarUsuario.sh

fi

echo "Presione Enter para volver..."

read x

. MenuUsuario.sh

;;

esac

done

EliminarUsuario.sh

#!/bin/bash

#EliminarUsuario

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Usuario a eliminar: "

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

userdel -r $Usuario

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Usuario eliminado con exito... \e[0m"

sleep 2

else

echo -e -n "\n\e[0;31m \n[x] Ocurrio un error.\e[0m"

sleep 2

fi

else

echo -e -n "\n\e[0;31m [x] El nombre de usuario no existe...\e[0m"

sleep 2

clear

fi

BloquearDesbloquear.sh

#!/bin/bash

#BloquearDesbloquear

while [ m!=3 ]

do

clear

echo -e "\e[1;32m¿Que desea hacer?\n\e[0m"

echo -e " [1] Bloquear Usuario"

echo -e " [2] Desbloquear Usuario"

echo -e " [3] Volver"

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Opcion :"

read m

case $m in

1). BloquearUsuario.sh

;;

2). DesbloquearUsuario.sh

;;

3)break

;;

\*)echo -e "\n\e[0;31m [x] Opcion incorrecta...\e[0m"

sleep 2

clear

;;

esac

done

BloquearUsuario.sh

#!/bin/bash

#BloquearUsuario

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Usuario a Bloquear :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

usermod -L $Usuario

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Usuario Bloqueado con exito... \e[0m"

sleep 2

else

sleep 1

clear

echo -e -n "\e[0;31m [x] Ocurrio un error.\e[0m"

sleep 2

fi

else

echo -e -n "\n\e[0;31m [x] El nombre de usuario no existe...\e[0m"

sleep 2

clear

fi

#!/bin/bash

DesbloquearUsuario.sh

#DesbloquearUsuario

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Usuario a Desbloquear :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

usermod -U $Usuario

if [ $? == 0 ]

then

sleep 2

clear

echo -e -n "\e[0;32m> Usuario Desbloqueado con exito... \e[0m"

sleep 2

else

sleep 2

clear

echo -e -n "\e[0;31m [x] Ocurrio un error.\e[0m"

sleep 2

fi

else

echo -e -n "\n\e[0;31m [x] El nombre de usuario no existe...\e[0m"

sleep 2

clear

fi

ConsultaUsuario.sh

#!/bin/bash

#ConsultaUsuario

clear

echo -e -n "\e[0;32m>\e[0m" "Ingrese nombre de Usuario :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

#Usuario1=$(grep "$Usuario" /etc/passwd)

if [ "$Usuario1" != "" ]

then

sleep 1

echo -e -n "\n\e[0;34m>\e[0m" "\e[1;34m@info del Usuario: \n\e[0m"

finger $Usuario

echo -e -n "\n\e[0;34m>\e[0m" "\e[1;34m@Presione Enter para continuar...\n\e[0m"

read x

else

sleep 1

echo -e -n "\n\e[0;31m [x] El nombre de usuario no existe...\e[0m"

sleep 2

clear

fi

#!/bin/bash

ModificarUsuarioLogin.sh

#!/bin/bash

#ModificarUsuarioLogin

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese login del Usuario :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

echo -e -n "\n\e[0;34m>\e[0m" "Ingrese nuevo login :"

read log

usermod $Usuario -l $log

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Modificacion realizada con exito... \e[0m"

sleep 2

else

sleep 1

echo -e -n "\n\e[0;31m [x] Se produjo un error...\e[0m"

sleep 2

fi

else

echo -e -n "\n\e[0;31m [x] El nombre de usuario no existe...\e[0m"

sleep 2

clear

fi

ModificarUsuarioG.sh

#!/bin/bash

#ModificarUsuarioG

while [ m!=3 ]

do

clear

echo -e "\e[1;32m¿Que desea modificar?\n\e[0m"

echo -e " [1] Grupo Principal"

echo -e " [2] Grupo Secundario"

echo -e " [3] Volver"

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Opcion :"

read m

case $m in

1). ModificarUsuarioGP.sh

;;

2). ModificarUsuarioGS.sh

;;

3)break

;;

\*)echo -e "\n\e[0;31m [x] Opcion incorrecta...\e[0m"

sleep 2

clear

;;

esac

done

ModificarUsuarioGP.sh

#!/bin/bash

#ModificarUsuarioGP

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese nombre de Usuario :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese nombre o GID de un grupo: "

read gr

usermod -g $gr $Usuario

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Modificacion realizada con exito... \e\n\n[0m"

sleep 1

id $Usuario

echo -e -n "\n\e[1;34m> Presione Enter para volver... \e[0m"

read x

else

sleep 1

echo -e -n "\e[0;31m \n[x] Se produjo un error.\e[0m"

sleep 2

fi

else

echo -e -n "\n\e[0;31m [x] El nombre de usuario no existe..."

sleep 2

clear

fi

ModificarUsuarioComentario.sh

#!/bin/bash

#ModificarUsuarioComentario

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese nombre de Usuario :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

echo -e -n "\n\e[0;34m>\e[0m" "Ingrese nuevo comentario :"

read com

usermod $Usuario -c "$com"

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Modificacion realizada con exito... \e[0m"

sleep 2

else

sleep 1

echo -e -n "\e[0;31m \n[x] Ocurrio un error.\e[0m"

sleep 2

fi

else

echo -e -n "\e[0;31m \n[x] El nombre de usuario no existe...\e[0m"

sleep 2

clear

fi

BloquearDesbloquear.sh

#!/bin/bash

#BloquearDesbloquear

while [ m!=3 ]

do

clear

echo -e "\e[1;32m¿Que desea hacer?\n\e[0m"

echo -e " [1] Bloquear Usuario"

echo -e " [2] Desbloquear Usuario"

echo -e " [3] Volver"

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Opcion :"

read m

case $m in

1). BloquearUsuario.sh

;;

2). DesbloquearUsuario.sh

;;

3)break

;;

\*)echo -e "\n\e[0;31m [x] Opcion incorrecta...\e[0m"

sleep 2

clear

;;

esac

done

BloquearUsuario.sh

#!/bin/bash

#BloquearUsuario

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Usuario a Bloquear :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

usermod -L $Usuario

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Usuario Bloqueado con exito... \e[0m"

sleep 2

else

sleep 1

clear

echo -e -n "\e[0;31m [x] Ocurrio un error.\e[0m"

sleep 2

fi

else

echo -e -n "\n\e[0;31m [x] El nombre de usuario no existe...\e[0m"

sleep 2

clear

fi

#!/bin/bash

DesbloquearUsuario.sh

#DesbloquearUsuario

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Usuario a Desbloquear :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

usermod -U $Usuario

if [ $? == 0 ]

then

sleep 2

clear

echo -e -n "\e[0;32m> Usuario Desbloqueado con exito... \e[0m"

sleep 2

else

sleep 2

clear

echo -e -n "\e[0;31m [x] Ocurrio un error.\e[0m"

sleep 2

fi

else

echo -e -n "\n\e[0;31m [x] El nombre de usuario no existe...\e[0m"

sleep 2

clear

fi

ModificarUsuario2.sh

#!/bin/bash

#ModificarUsuario2

while [ m!=5 ]

do

clear

echo -e -n "\e[0;36m> Opciones Avanzadas\e[0m"

echo -e " \n\e[0;34m \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\e[0m"

echo -e "\e[0;34m|\e[0m" " Modificar Usuario" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯|\e[0m"

echo -e "\e[0;34m|\e[0m" " [1] UID\t\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [2] Home\t\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [3] Shell\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [4] Clave\t\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [5] Atras\t\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\e[0m"

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Opcion :"

read m

case $m in

1). ModificarUsuarioUID.sh

;;

2). ModificarUsuarioHome.sh

;;

3). ModificarUsuarioShell.sh

;;

4). ModificarUsuarioClave.sh

;;

5)break

;;

\*)echo -e "\n\e[0;31m [x] Opcion incorrecta...\e[0m"

sleep 2

clear

;;

esac

done

ModificarUsuarioUID.sh

#ModificarUsuarioUID

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese nombre de Usuario :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

echo -e -n "\n\e[0;34m>\e[0m" "Ingrese nuevo UID: "

read ui

usermod -u $ui $Usuario

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Modificacion realizada con exito... \e[0m"

sleep 1

echo -e "\n "

id $Usuario

sleep 3

else

sleep 1

echo -e -n "\e[0;31m \n[x] Error, el UID debe ser unico o no tiene los permisos necesarios...\e[0m"

sleep 2

fi

else

echo -e -n "\e[0;31m \n[x] El nombre de usuario no existe...\e[0m"

sleep 2

clear

fi

ModificarUsuarioHome.sh

#!/bin/bash

#ModificarUsuarioHome

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese nombre de Usuario :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

echo -e -n "\n\e[0;34m>\e[0m" "Ingrese nuevo Home: "

read hom

usermod $Usuario -d /home/$hom

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Modificacion realizada con exito... \e[0m"

sleep 2

else

sleep 1

echo -e -n "\e[0;31m \n[x] Ocurrio un error...\e[0m"

sleep 2

fi

else

echo -e -n "\n\e[0;31m [x] El nombre de usuario no existe...\e[0m"

sleep 2

clear

fi

ModificarUsuarioShell.sh

#!/bin/bash

#ModificarUsuarioShell

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese nombre de Usuario :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

echo -e -n "\n\e[0;34m>\e[0m" "Ingrese nueva Shell: "

read she

usermod -s /bin/$she $Usuario

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Modificacion realizada con exito... \e[0m"

sleep 2

else

speel 1

echo -e -n "\e[0;31m \n[x] Se produjo un error...\e[0m"

sleep 2

fi

else

echo -e -n "\n\e[0;31m [x] El nombre de usuario no existe...\e[0m"

sleep 2

clear

fi

ModificarUsuarioClave.sh

#!/bin/bash

#ModificarUsuarioClave

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese nombre de Usuario :"

read Usuario

Usuario1=$(cut -d ":" -f1 /etc/passwd | grep -wi "$Usuario" )

if [ "$Usuario1" != "" ]

then

passwd $Usuario

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Clave modificada con exito... \e[0m"

sleep 2

else

sleep 1

echo -e -n "\e[0;31m \n[x] Ocurrio un error.\e[0m"

sleep 2

fi

else

echo -e -n "\e[0;31m \n[x] El nombre de Usuario no existe...\e[0m"

sleep 2

clear

fi

MenuGrupo.sh

#!/bin/bash

#MenuGrupo

while [ m!=5 ]

do

clear

echo -e " \e[0;34m\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\e[0m"

echo -e "\e[0;34m|\e[0m" "\t GRUPOS\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯|\e[0m"

echo -e "\e[0;34m|\e[0m" " [1] Crear un Grupo\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [2] Modificar un Grupo" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [3] Eliminar un Grupo" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [4] Consultas\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\e[0m" " [5] Atras\t\t" " \e[0;34m|\e[0m"

echo -e "\e[0;34m|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\e[0m"

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Opcion :"

read m

case $m in

1). CrearGrupo.sh

;;

2). ModificarGrupo.sh

;;

3). EliminarGrupo.sh

;;

4). ConsultaGrupo.sh

;;

5)break

;;

\*)echo -e "\n\e[0;31m [x] Opcion incorrecta...\e[0m"

sleep 2

clear

;;

esac

done

CrearGrupo.sh

#!/bin/bash

#CrearGrupo

clear

echo -e "\e[1;34m \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\e[0m"

echo -e "\e[1;34m|\e[0m" "\e[0;33m CREACION DE GRUPOS \e[0m" " \e[1;34m|\e[0m"

echo -e "\e[1;34m ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯\e[0m"

echo -e

sleep 2

echo -e -n "\e[1;33m Ingrese los siguientes datos...\e[0m"

sleep 3

clear

echo -e "\e[1;34m \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\e[0m"

echo -e "\e[1;34m|\e[0m" "\e[0;33m CREACION DE GRUPOS \e[0m" " \e[1;34m|\e[0m"

echo -e "\e[1;34m ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯\e[0m"

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Grupo a Crear: "

read Grupo

Grupo1=$(grep "$Grupo" /etc/group)

if [ "$Grupo1" == "" ]

then

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese GID: "

read gr

groupadd -g $gr $Grupo

if [ $? == 0 ]

then

clear

echo -e -n "\e[1;36mEspere\e[0m"

sleep 1

clear

echo -e -n "\e[1;36mEspere.\e[0m"

sleep 1

clear

echo -e -n "\e[1;36mEspere..\e[0m"

sleep 1

clear

echo -e -n "\e[1;36mEspere...\e[0m"

sleep 2

clear

echo -e -n "\e[0;32m> Grupo creado exitosamente. \e[0m"

sleep 2

else

sleep 1

echo -e -n "\n\e[0;31m [x] Se produjo un error...\e[0m"

sleep 2

fi

else

echo -e -n "\n\e[0;31m [x] El nombre de Grupo ya existe...\e[0m"

sleep 2

clear

fi

ModificarGrupo.sh

#!/bin/bash

#ModificarGrupo

while [ m!=3 ]

do

clear

echo -e "\e[1;32m¿Que desea modificar?\n\e[0m"

echo -e " [1] GID de un Grupo"

echo -e " [2] Nombre de un Grupo"

echo -e " [3] Volver"

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Opcion :"

read m

case $m in

1). ModificarGrupoGID.sh

;;

2). ModificarGrupoNombre.sh

;;

3)break

;;

\*)echo -e "\n\e[0;31m [x] Opcion incorrecta...\e[0m"

sleep 2

clear

;;

esac

done

ModificarGrupoGID.sh

#!/bin/bash

#ModificarGrupoGID

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese nombre del grupo :"

read Grupo

Grupo1=$(grep "$Grupo" /etc/group)

if [ "$Grupo1" != "" ]

then

echo -e -n "\n\e[0;34m>\e[0m" "Ingrese nuevo GID para el Grupo: "

read gid

groupmod -g $gid $Grupo

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Modificacion realizada con exito... \e[0m"

sleep 2

else

sleep 1

echo -e -n "\e[0;31m \n[x] Ocurrio un error.\e[0m"

sleep 2

fi

else

echo -e -n "\e[0;31m \n[x] El nombre de Grupo no existe...\e[0m"

sleep 2

clear

fi

ModificarGrupoNombre.sh

#!/bin/bash

#ModificarGrupoNombre

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese nombre del grupo :"

read Grupo

Grupo1=$(grep "$Grupo" /etc/group)

if [ "$Grupo1" != "" ]

then

echo -e -n "\n\e[0;34m>\e[0m" "Ingrese nuevo nombre para el Grupo: "

read gr

groupmod -n $gr $Grupo

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Modificacion realizada con exito... \e[0m"

sleep 2

else

sleep 1

echo -e -n "\e[0;31m \n[x] Ocurrio un error.\e[0m"

sleep 2

fi

else

echo -e -n "\e[0;31m \n[x] El nombre de Grupo no existe...\e[0m"

sleep 2

clear

fi

EliminarGrupo.sh

#!/bin/bash

#EliminarGrupo

clear

echo -e -n "\n\e[0;32m>\e[0m" "Ingrese Grupo a eliminar :"

read Grupo

Grupo1=$(grep "$Grupo" /etc/group)

if [ "$Grupo1" != "" ]

then

groupdel $Grupo

if [ $? == 0 ]

then

sleep 1

clear

echo -e -n "\e[0;32m> Grupo eliminado... \e[0m"

sleep 2

else

echo -e -n "\e[0;31m \n[x] Ocurrio un error.\e[0m"

sleep 2

fi

else

echo -e -n "\e[0;31m \n[x] El nombre de Grupo no existe...\e[0m"

sleep 2

clear

fi

ConsultaGrupo.sh

#!/bin/bash

#ConsultaGrupo

clear

echo -e -n "\e[0;32m>\e[0m" "Ingrese nombre del Grupo :"

read Grupo

Grupo1=$(grep "$Grupo" /etc/group)

if [ "$Grupo1" != "" ]

then

sleep 1

echo -e -n "\n\e[0;34m>\e[0m" "\e[1;34m@info del Grupo: \n\e[0m"

grep $Grupo /etc/group

echo -e -n "\n\e[0;34m>\e[0m" "\e[1;34m@Presione Enter para continuar...\n\e[0m"

read x

else

echo -e -n "\n\e[0;31m [x] El nombre del Grupo no existe...\e[0m"

sleep 2

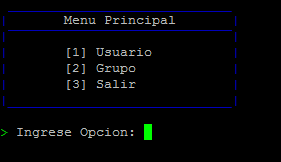
clear

fi

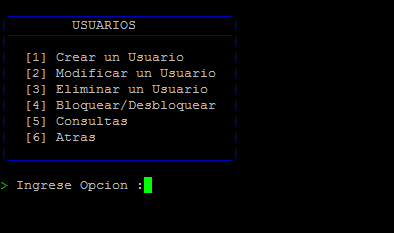
**Capturas del shellscript**

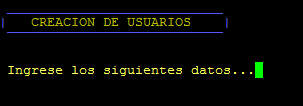
**MenuPrincipal.sh**

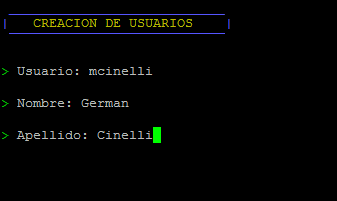
Menu principal del shellscript de altas, bajas y modificaciones tanto como para usuarios como para los grupos.



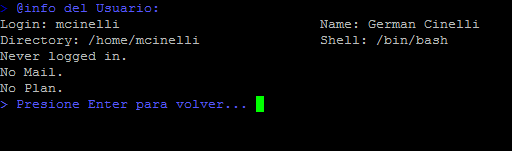
**MenuUsuario.sh**

****

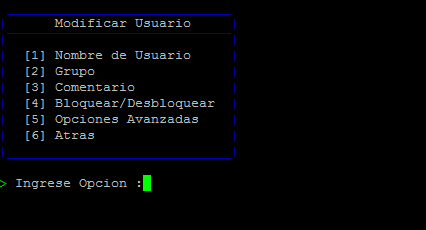
**CrearUsuario.sh**

****

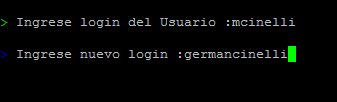
**5.png**

****

**ModificarUsuario.sh**

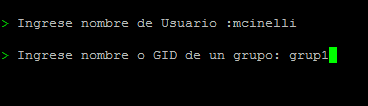
****

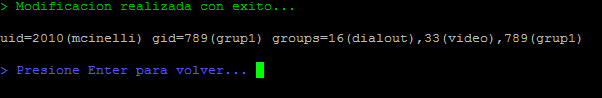
**ModificarUsuarioNombre.sh**

****

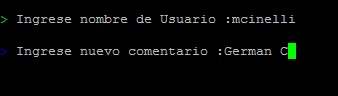
**9.png**

**ModificarUsuarioGrupo.sh**

****

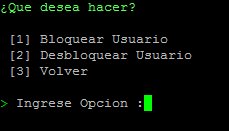
****

**ModificarUsuarioComentario.sh**

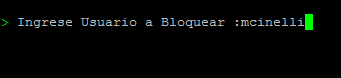
****

**9.png**

**BloquearDesbloquear.sh**

****

**BloquearUsuario.sh**

****

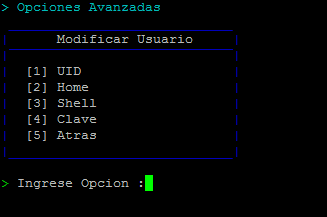
**14.png**

**15.png**

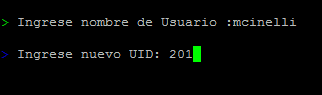
**DesbloquearUsuario.sh**

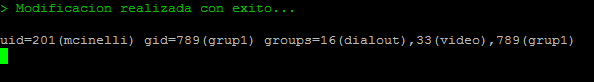
**16.png**

**ModificarUsuario2.sh**

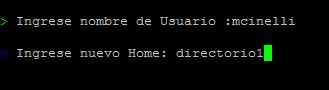
****

**ModificarUsuarioUID.sh**

****

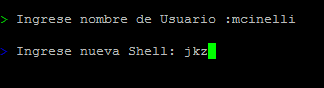
****

**ModificarUsuarioHome.sh**

****

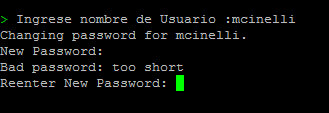
**9.png**

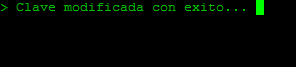
**ModificarUsuarioShell.sh**

****

**9.png**

**ModificarUsuarioClave.sh**

****

****

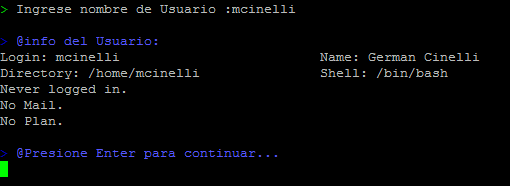
**EliminarUsuario.sh**

**24.png**

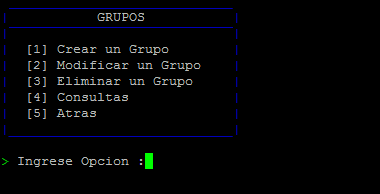
**9.png**

**ConsultaUsuario.sh**

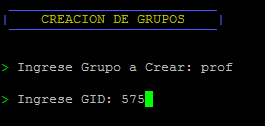
**25.png**

****

**MenuGrupo.sh**

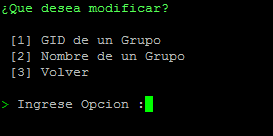
****

**CrearGrupo.sh**

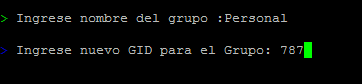
****

**29.png**

**ModificarGrupo.sh**

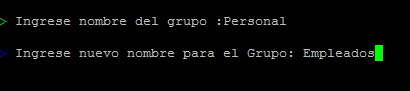
****

**ModificarGrupoGID.sh**

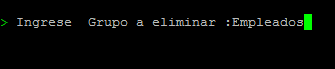
****

**9.png**

**ModificarUsuarioNombre.sh**

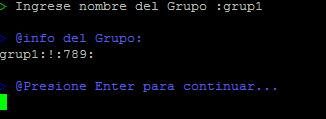
****

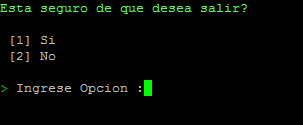
**9.png**

**EliminarGrupo.sh**

**36.png**

**ConsultaGrupo.sh**

****

**39.png40.pngMenuSalir.sh**

ABMC (Animales)

Código:

#!/bin/bash

function setcolors {

BLACK='033[0;30m'

RED='\033[0;31m'

GREEN='\033[0;32m'

BROWN='\033[0;33m'

BLUE='\033[0;34m'

PURPLE='\033[0;35m'

CYAN='\033[0;36m'

YELLOW='\033[1;33m'

WHITE='\033[1;37m'

printf '\e[8;30;100t'

clear

}

function ejecutar {

clear

setcolors

clear

menu

}

############################################################ MENU ###################################################################

function menu {

while [ $op!=5 ]

do

clear

echo -e "${YELLOW}Menu Principal"

echo

echo -e "${PURPLE}1) ${WHITE}Adminstrador general"

echo -e "${PURPLE}2) ${WHITE}Administrador individual"

echo -e "${PURPLE}3) ${WHITE}Auxiliar"

echo -e "${PURPLE}4) ${WHITE}Salir"

echo -e "${YELLOW}"

read -p "Ingrese una opción: " op

case $op in

1) admingeneral

;;

2) adminingeneral

;;

3) auxiliar

;;

4) exit

;;

\*) echo -e "${RED}¡Opción inválida!"

sleep 1.5 ;;

esac

done

}

#######################################################################################################################################

############################################################# USUARIOS ################################################################

#Usuario General

function admingeneral {

while [ $op!=6 ]

do

clear

echo -e "${YELLOW}Administracion general."

echo

echo -e "${PURPLE}1) ${WHITE}Usuarios"

echo -e "${PURPLE}2) ${WHITE}Animales"

echo -e "${PURPLE}3) ${WHITE}Atras"

echo -e "${YELLOW}"

read -p "Ingrese una opcion: " op

case $op in

1) .MenuPrincipal.sh

;;

2) animal

;;

3) menu

;;

\*) echo -e "${RED}Opcion invalida!"

sleep 1.5 ;;

esac

done

}

#Usuario Individual

function adminindividual {

while [ $op!=6 ]

do

clear

echo -e "${YELLOW}Administracion individual."

echo

echo -e "${PURPLE}1) ${WHITE}Animal"

echo -e "${PURPLE}2) ${WHITE}Bajas"

echo -e "${PURPLE}3) ${WHITE}Modificaciones"

echo -e "${PURPLE}4) ${WHITE}Consultas (Busquedas)"

echo -e "${PURPLE}5) ${WHITE}Listados"

echo -e "${PURPLE}6) ${WHITE}Volver"

echo -e "${YELLOW}"

read -p "Ingrese una opcion: " op

case $op in

1) ai\_animal

;;

2) ai

;;

3) modemp

;;

4) consemp

;;

5) listemp

;;

6) menu

;;

\*) echo -e "${RED}Opcion invalida!"

sleep 1.5 ;;

esac

done

}

#Usuario Auxiliar

function auxiliar {

while [ $op!=6 ]

do

clear

echo -e "${YELLOW}Animales"

echo

echo -e "${PURPLE}1) ${WHITE}Altas"

echo -e "${PURPLE}2) ${WHITE}Bajas"

echo -e "${PURPLE}3) ${WHITE}Modificaciones"

echo -e "${PURPLE}4) ${WHITE}Consultas (Busquedas)"

echo -e "${PURPLE}5) ${WHITE}Listados"

echo -e "${PURPLE}6) ${WHITE}Volver"

echo -e "${YELLOW}"

read -p "Ingrese una opcion: " op

case $op in

1) al

;;

2) bajemp

;;

3) modemp

;;

4) consemp

;;

5) listemp

;;

6) menu

;;

\*) echo -e "${RED}Opcion invalida!"

sleep 1.5 ;;

esac

done

}

####################################################################################################################################

############################################################## Animal ##############################################################

function animal {

while [ $op!=6 ]

do

clear

echo -e "${YELLOW}Animales:"

echo

echo -e "${PURPLE}1) ${WHITE}Toros"

echo -e "${PURPLE}2) ${WHITE}Vacas"

echo -e "${PURPLE}3) ${WHITE}Crias"

echo -e "${PURPLE}6) ${WHITE}Volver"

echo -e "${YELLOW}"

read -p "Ingrese una opcion: " op

case $op in

1) menu\_toro

;;

2) menu\_vaca

;;

3) menu\_cria

;;

4) menu

;;

\*) echo -e "${RED}Opcion invalida!"

sleep 1.5 ;;

esac

done

}

############# Toro

function menu\_toro {

while [ $op!=6 ]

do

clear

echo -e "${YELLOW}------- TOROS ------"

echo

echo -e "${PURPLE}1) ${WHITE}Altas"

echo -e "${PURPLE}2) ${WHITE}Bajas"

echo -e "${PURPLE}3) ${WHITE}Modificaciones"

echo -e "${PURPLE}4) ${WHITE}Consultas (Busquedas)"

echo -e "${PURPLE}5) ${WHITE}Listados"

echo -e "${PURPLE}6) ${WHITE}Volver"

echo -e "${YELLOW}"

read -p "Ingrese una opcion: " op

case $op in

1) alta\_toro

;;

2) baja\_toro

;;

3) mod\_toro

;;

4) cons\_toro

;;

5) list\_toro

;;

6) animal

;;

\*) echo -e "${RED}Opcion invalida!"

sleep 1.5 ;;

esac

done

}

############# Vaca

function menu\_vaca {

while [ $op!=6 ]

do

clear

echo -e "${YELLOW}------- VACAS ------"

echo

echo -e "${PURPLE}1) ${WHITE}Altas"

echo -e "${PURPLE}2) ${WHITE}Bajas"

echo -e "${PURPLE}3) ${WHITE}Modificaciones"

echo -e "${PURPLE}4) ${WHITE}Consultas (Busquedas)"

echo -e "${PURPLE}5) ${WHITE}Listados"

echo -e "${PURPLE}6) ${WHITE}Volver"

echo -e "${YELLOW}"

read -p "Ingrese una opcion: " op

case $op in

1) alta\_vaca

;;

2) baja\_vaca

;;

3) mod\_vaca

;;

4) cons\_vaca

;;

5) list\_vaca

;;

6) animal

;;

\*) echo -e "${RED}Opcion invalida!"

sleep 1.5 ;;

esac

done

}

############# Cria

function menu\_cria {

while [ $op!=6 ]

do

clear

echo -e "${YELLOW}------- CRIAS ------"

echo

echo -e "${PURPLE}1) ${WHITE}Altas"

echo -e "${PURPLE}2) ${WHITE}Bajas"

echo -e "${PURPLE}3) ${WHITE}Modificaciones"

echo -e "${PURPLE}4) ${WHITE}Consultas (Busquedas)"

echo -e "${PURPLE}5) ${WHITE}Listados"

echo -e "${PURPLE}6) ${WHITE}Volver"

echo -e "${YELLOW}"

read -p "Ingrese una opcion: " op

case $op in

1) alta\_cria

;;

2) baja\_cria

;;

3) mod\_cria

;;

4) cons\_cria

;;

5) listado\_cria

;;

6) animal

;;

\*) echo -e "${RED}Opcion invalida!"

sleep 1.5 ;;

esac

done

}

############# Alta

function alta\_toro {

sexo="Macho"

clear

echo

read -p "Identificacion: " id

read -p "Nombre: " nom

read -p "Tipo: " tipo

read -p "Raza: " raza

read -p "Fecha nacimiento: " fecha\_nac

echo $id:$sexo:$nom:$tipo:$raza:$fecha\_nac >> toro.txt

sleep 1.5

echo -e "${GREEN}Datos ingresados correctamente."

sleep 1.5

menu\_toro

}

############# Baja

function baja\_toro {

clear

read -p "Identificación: " id

if [ $? -ne 0 ]

then

echo -e "${RED}Eliminando..."

sleep 1

sed -i "/$id/d" toro.txt

echo -e "${GREEN}Registro de toro eliminado!"

sleep 1.5

else

echo "El registo no existe"

sleep 1.5

fi

menu\_toro

}

############# Modificacion

function mod\_toro {

clear

read -p "Identificación: " id

echo "Datos del toro: "

if grep $id toro.txt

then

echo "Que desea modificar:"

echo "1- Nombre"

echo "2- Tipo"

echo "Ingrese opcion"

read op

case $op in

1)

read -p "Nombre actual: " nom1

read -p "Nuevo nombre: " nom2

sed "s/$nom1/$nomb2/" toro.txt

echo "Modificacion exitosa"

;;

2)

read -p "Tipo actual: " tipo1

read -p "Nuevo tipo: " tipo2

sed "s/$tipo1/$tipo2/" toro.txt

echo "Modificacion exitosa"

;;

esac

else

echo "El toro no existe."

fi

menu\_toro

}

############# Consulta

function cons\_toro {

clear

read -p "Identificacion: " id

if grep $id toro.txt

then

read -p "Presione ENTER para continuar" j

else

echo "EL registro no existe"

fi

menu\_toro

}

############# Listado

function list\_toro {

echo "Identificacion : Sexo : Nombre : Tipo : Raza : Fecha nacimiento"

cat toro.txt

read -p "Presione ENTER para continuar" j

menu\_toro

}

############# Alta

function alta\_vaca {

sexo="Hembra"

clear

echo

read -p "Identificacion: " id

read -p "Nombre: " nom

read -p "Tipo: " tipo

read -p "Raza: " raza

read -p "Fecha nacimiento: " fecha\_nac

echo $id:$sexo:$nom:$tipo:$raza:$fecha\_nac >> vaca.txt

sleep 1.5

echo -e "${GREEN}Datos ingresados correctamente."

sleep 1.5

menu\_vaca

}

############# Baja

function baja\_vaca {

clear

read -p "Identificación: " id

echo -e "${RED}Eliminando..."

sleep 1

sed -i "/$id/d" vaca.txt

echo -e "${GREEN}Registro de vaca eliminado!"

sleep 2

}

############# Modificacion

function mod\_vaca {

clear

read -p "Identificación: " id

echo "Datos del vaca: "

grep $id vaca.txt

if [ $? -ne 0 ]

then

echo "Que desea modificar:"

echo "1- Nombre"

echo "2- Tipo"

echo "Ingrese opcion"

read op

case $op in

1)

read -p "Nombre actual: " nom1

read -p "Nuevo nombre: " nom2

sed "s/$nom1/$nomb2/" vaca.txt

echo "Modificacion exitosa"

;;

2)

read -p "Tipo actual: " tipo1

read -p "Nuevo tipo: " tipo2

sed "s/$tipo1/$tipo2/" vaca.txt

echo "Modificacion exitosa"

;;

esac

else

echo "La vaca no existe."

fi

}

############# Consulta

function cons\_vaca {

clear

read -p "Identificación: " id

grep $id vaca.txt

if [ $? -ne 0 ]

then

echo "s"

else

echo "Error al buscar"

fi

}

############# Listado

function list\_vaca {

echo "Identificacion : Sexo : Nombre : Tipo : Raza : Fecha nacimiento"

cat vaca.txt

}

############# Alta

function alta\_cria {

clear

echo

read -p "Identificacion: " id

read -p "Nombre: " nom

read -p "Tipo: " tipo

read -p "Raza: " raza

read -p "Sexo:" sexo

read -p "Fecha nacimiento: " fecha\_nac

echo $id:$sexo:$nom:$tipo:$raza:$fecha\_nac >> cria.txt

sleep 1.5

echo -e "${GREEN}Datos ingresados correctamente."

sleep 1.5

menu\_cria

}

############# Baja

function baja\_cria {

clear

read -p "Identificación: " id

echo -e "${RED}Eliminando..."

sleep 1

sed -i "/$id/d" cria.txt

echo -e "${GREEN}Registro de toro eliminado!"

sleep 2

}

############# Modificacion

function mod\_cria {

clear

read -p "Identificación: " id

echo "Datos de la cria: "

grep $id cria.txt

if [ $? -ne 0 ]

then

echo "Que desea modificar:"

echo "1- Nombre"

echo "2- Tipo"

echo "Ingrese opcion"

read op

case $op in

1)

read -p "Nombre actual: " nom1

read -p "Nuevo nombre: " nom2

sed "s/$nom1/$nomb2/" cria.txt

echo "Modificacion exitosa"

;;

2)

read -p "Tipo actual: " tipo1

read -p "Nuevo tipo: " tipo2

sed "s/$tipo1/$tipo2/" cria.txt

echo "Modificacion exitosa"

;;

esac

else

echo "La cria no existe."

fi

}

############# Consulta

function cons\_cria {

clear

read -p "Identificación: " id

grep $id cria.txt

if [ $? -ne 0 ]

then

echo "s"

else

echo "Error al buscar"

fi

}

############# Listado

function list\_cria {

echo "Identificacion : Sexo : Nombre : Tipo : Raza : Fecha nacimiento"

cat cria.txt

}

ejecutar

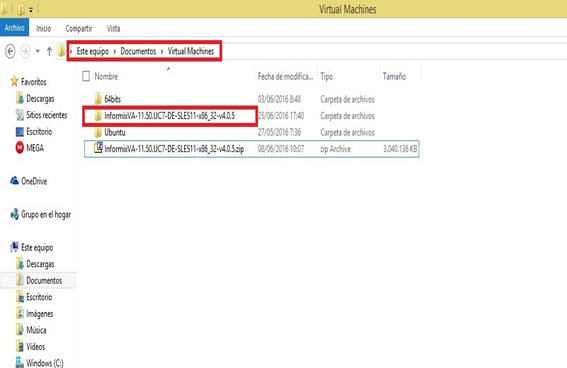
Instalación de máquina virtual

Requisitos para una instalación de una máquina virtual optima**:**

- VMware® Workstation-

- Una máquina virtual “informixVA” (en este caso se utilizó la versión 11.50.UC7).

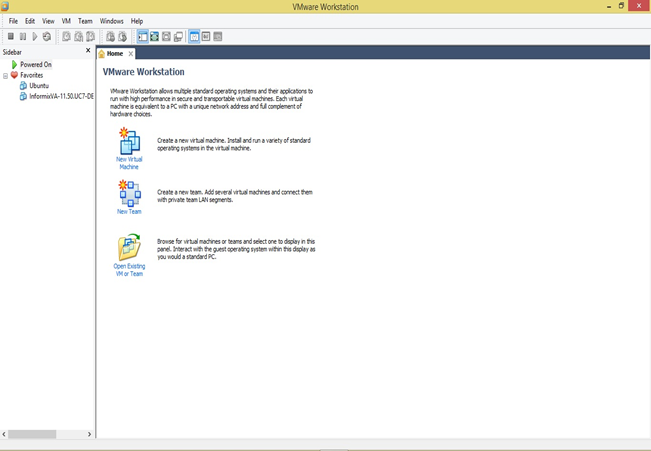
**Pasos previos a la instalación de nuestra máquina virtual:**

1. Se debe guardar la máquina virtual donde se pueda recordar fácilmente (una sugerencia en la carpeta documentos de la máquina) y esta estará dentro de una carpeta llamada “Virtual Machines”.

2. Luego será instalar el programa previamente descargado (el lugar de instalación será donde el cliente o consumidor lo vea más óptimo).

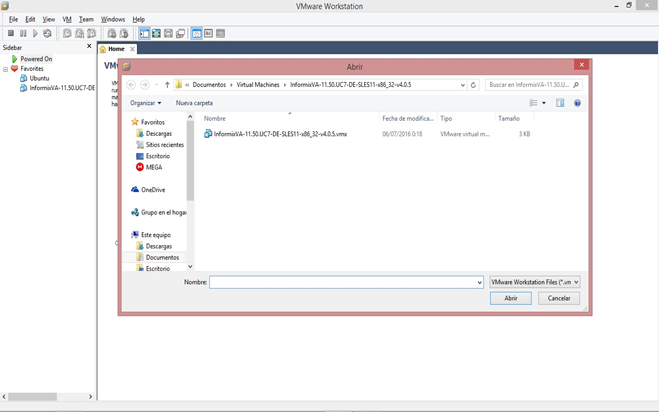
**Pasos para abrir nuestra máquina virtual y su funcionamiento:**

1. Lo primero que verá el usuario al querer abrir la máquina virtual a través del programa VMware® será:



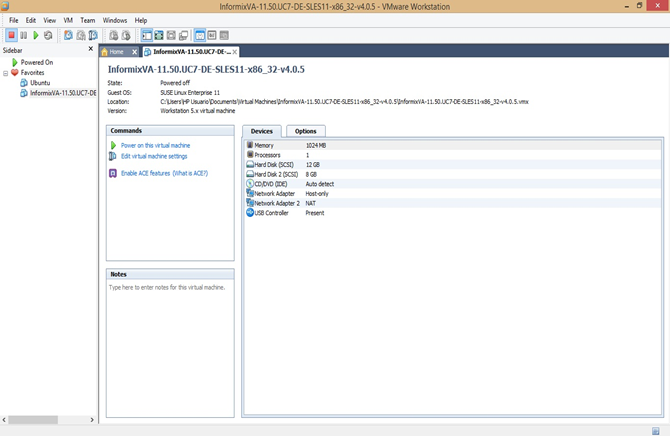
El usuario deberá seleccionar el tercer botón u opción “Open Existing VM or Team”.

2. Como siguiente paso será buscar la máquina virtual en la venta que nos aparece al terminar el paso anterior,

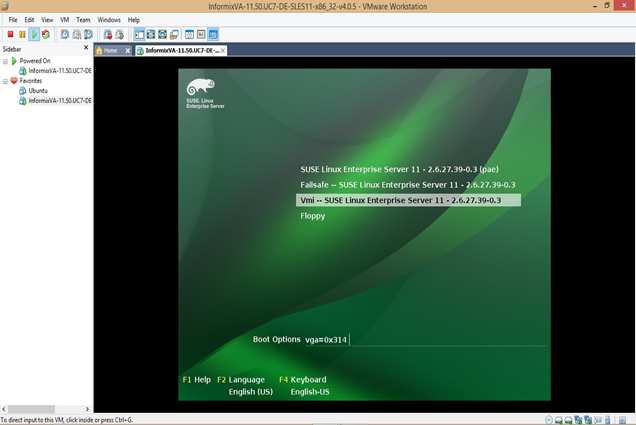


Al encontrarlo se dará clic al archivo y luego al botón que dice “Abrir”.

3. Al terminar el paso 2 se mostrará un viñeta o pestaña nueva donde estaba al principio en el paso 1, En la cual se verá las especificaciones de la máquina virtual como su RAM , ROM o sus procesadores estas especificaciones se podrán modificar como en este caso que se le dio 1GB de RAM y un procesador.

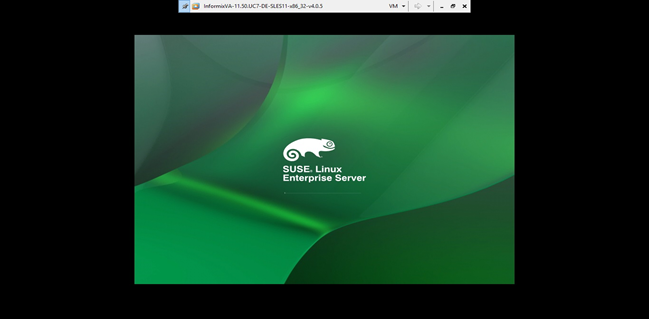


4. Al colocarle nuestras especificaciones más óptimas (si se llegó a modificar) se le dará en la primera opción de la izquierda que dice “Power on the virtual machine” que es para iniciar la máquina virtual.



Al terminar de cargar varias consolar que no se logran a ver su contenido saldrá esta pantalla donde se puede configurar el idioma del sistema y el idioma del teclado luego de configurar (si se requiere) le daremos ENTER para continuar a la siguiente pantalla.

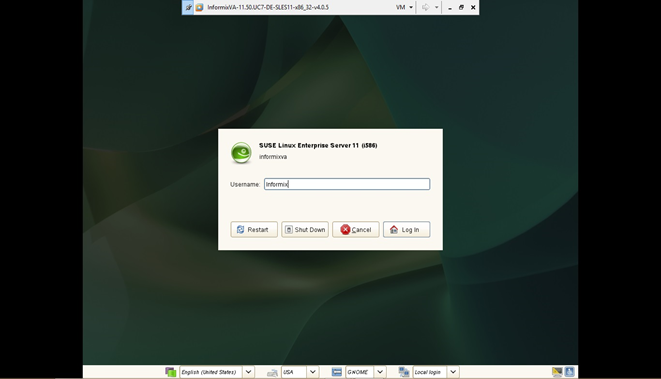
5. En este paso es solo tener paciencia ya que lo único que se podrá hacer es esperar a que cargue el sistema de la máquina virtual (si es la primera vez que se inicia tardará un poco más en las próximas será más rápido)



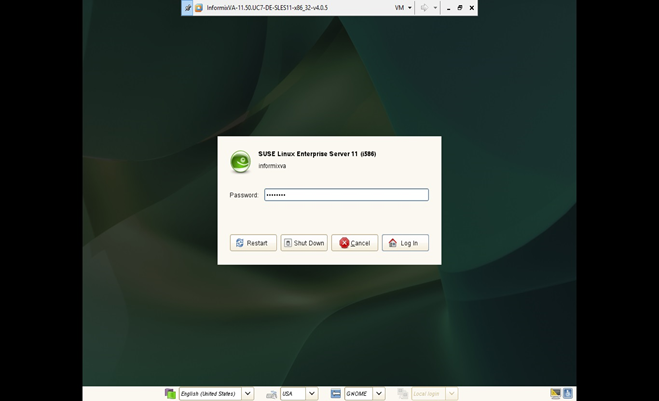
6. En este paso es el login o el ingreso de un usuario siendo la primera vez de su inicio solo tendrá dos usuarios por defecto:

-El usuario Root.

-El usuario informix (se utiliza mayormente para la conexión de una base de datos con el programa Putty que no viene al caso en este Manual).



7. En este paso se tendrá que ingresar la contraseña del usuario introducido en el anterior paso

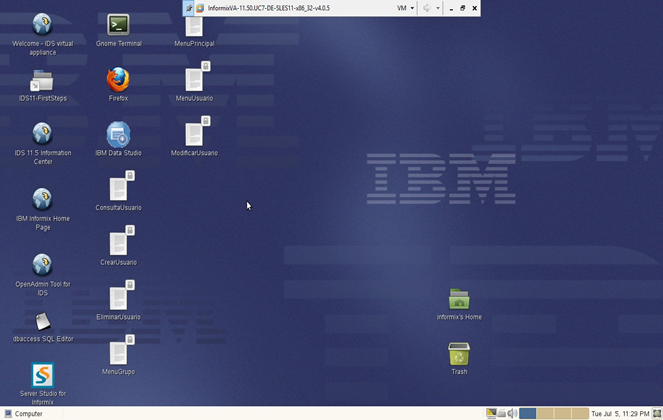


Cada usuario que hay por defecto tiene la contraseña igual que su nombre.

Nombre: root; Contraseña: root.

Nombre: informix; Contraseña: informix

8. Y por último esperar que inicie el usuario, estaría completa nuestra máquina virtual con un funcionamiento correcto.



¿Qué es cron?

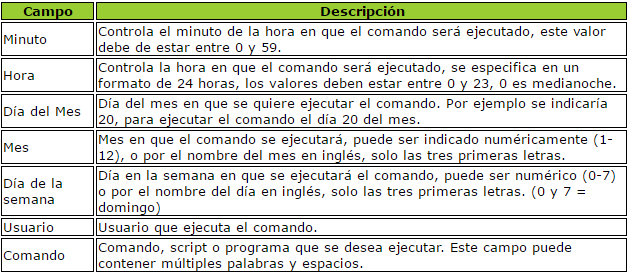
Cron es el nombre del programa que permite a usuarios Linux/Unix ejecutar automáticamente comandos o a una hora o fecha específica. Es usado normalmente para comandos de tareas administrativas, como respaldos, pero puede ser usado para ejecutar cualquier cosa.

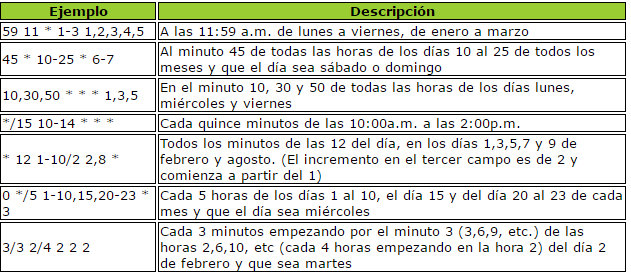
El comando crontab permite automatizar tareas

Parámetros:

crontab -l Lista las tareas

crontab -e Edita las tareas



Listado de ejemplos:

Un asterisco \* como valor en los primeros cinco campos, indicará inicio-fin del campo, es decir todo. Un \* en el campo de minuto indicará todos los minutos.

De éste modo Suru se encargará de hacer un respaldo de la base de datos de lunes a viernes de 23:30.

Comando: 30 23 \* \* 1-5 /home/usuario/respaldo.sh