

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

```
#fichero funciones
```

```
function cambiar_contrasena()
{
    #comprueba la contraseña
    comprobar_contrasena $2 $3

    result=$?
    #Si las 2 contraseñas no coinciden salimos
    if [ $result == 1 ]; then
        return 100
    else #Si coinciden se modifica

        #Se modifica la contraseña del usuario especificado
        echo -e "$2\n$2" | (passwd $1) 2> /dev/null
        result=$?

        #Contraseña modificada correctamente
        if [ $result == 0 ]; then
            return 140

        else #Error
            mostrar_error $result
        fi
    fi
}
```

```
function bloquear_usuario ()
{
    #Se bloquea la cuenta de usuario especificado
    usermod --lock --expiredate 2010-10-10 $1 2> /dev/null
    result=$?

    #Cuenta de usuario bloqueada correctamente
    if [ $result == 0 ]; then
        return 141

    else #Error
        mostrar_error $result
    fi
}
```

```
function desbloquear_usuario ()
{
    #Se desbloquea la cuenta de usuario especificado
    usermod --unlock --expiredate 99999 $1 2> /dev/null
    result=$?

    #Cuenta de usuario desbloqueada correctamente
    if [ $result == 0 ]; then
        return 142

    else #Error
        mostrar_error $result
    fi
}
```

```
function modificar_home ()
{
    #Se modifica la ruta home del usuario especificado
    usermod -m -d $2 $1 2> /dev/null
    result=$?
```

```
#Home modificado correctamente
if [ $result == 0 ]; then
    return 143

else #Error
    mostrar_error $result
fi

}

function modificar_comentario ()
{

    #Se modifica el comentario del usuario especificado
    usermod -c "$2" "$1" #2> /dev/null
    result=$?

    #Comentario modificado correctamente
    if [ $result == 0 ]; then
        return 144

    else #Error
        mostrar_error $result
    fi

}

function modificar_shell ()
{

    #Se modifica el shell del usuario especificado
    usermod -s /bin/$2 $1 2> /dev/null
    result=$?

    #Shell modificado correctamente
    if [ $result == 0 ]; then
        return 145

    else #Error
        mostrar_error $result
    fi

}

function mostrar_error ()
{

    if [ $1 == 1 ] || [ $1 == 2 ] || [ $1 == 3 ]; then
        #Formato incorrecto
        return 150
    elif [ $1 == 6 ]; then
        #Usuario no existe
        return 151
    elif [ $1 == 8 ]; then
        #El usuario está en uso
        return 152
    elif [ $1 == 10 ]; then
        #No se ha podido modificar la base de datos de grupos
        return 153
    elif [ $1 == 11 ]; then
        return 154
    elif [ $1 == 12 ]; then
        return 155
    fi

}

function modificar_nombre_grupo ()
{

    #Se modifica el nombre del grupo
```

```

groupmod -n $2 $1 2> /dev/null
result=$?

#Nombre del grupo modificado correctamente
if [ $result == 0 ]; then
    return 160
#Errores
elif [ $result == 2 ] || [ $result == 3 ]; then
    return 161
elif [ $result == 4 ] || [ $result == 6 ]; then
    return 162
elif [ $result == 9 ]; then
    return 163
elif [ $result == 10 ]; then
    return 164
fi
}

function agregar_usuario_grupo ()
{
    #Se añade el usuario al grupo especificado
    gpasswd -a $2 $1 > /dev/null 2> /dev/null
    result=$?

    #Usuario añadido al grupo correctamente
    if [ $result == 0 ]; then
        return 170
    #Error
    elif [ $result == 3 ]; then
        return 171
    fi
}

function eliminar_usuario_grupo ()
{
    #Se elimina el usuario del grupo especificado
    gpasswd -d $2 $1 > /dev/null 2> /dev/null
    result=$?

    #Usuario eliminado al grupo correctamente
    if [ $result == 0 ]; then
        return 180
    elif [ $result == 3 ]; then
        return 181
    fi
}

#Funcion mostrar usuario
function mostrar_usuarios()
{
    #Bucle para almacenar las lineas del archivo txt/$idioma/$1
    cont=0
    usuario="" #Almacena el texto usuario
    separador="" #Almacena el texto separador
    coment="" #Almacena el texto coment
    h="" #Almacena el texto home
    s="" #Almacena el texto shell

    while read n ; do
        if [ $cont -eq 0 ]; then
            usuario=$n;
        elif [ $cont -eq 1 ]; then
            separador=$n;
        elif [ $cont -eq 2 ]; then
            coment=$n;
        elif [ $cont -eq 3 ]; then
            h=$n;
        elif [ $cont -eq 4 ]; then
            s=$n;
        fi
    done
}

```

```

        let cont=$cont+1
done < txt/$idioma/$1 #archivo que se le pasa al bucle

l_usuarios="";
usuarios=`awk -v p1=$usuario -v p2=$separador -v p3=$coment -v p4=$h -v p5=$s '$3 >= 1000 &&
$3 <= 60000 {print p1, $1, p2, p3, $5, p4, $6, p5, $7, "."}' FS=":" /etc/passwd`

#IFS es la variable interna del bash que por defecto separa los campos por espacios en blanco
IFS=. #Definimos que separe los campos por el caracter .
for i in $usuarios
do
    l_usuarios+="$i";
done

}

#Funcion mostrar grupo
function mostrar_grupos()
{
    #Bucle para almacenar las lineas del archivo txt/$idioma/$1
    cont=0
    grupo="" #Almacena el texto grupo
    miembros="" #Almacena el texto miembros

    while read n ; do
        if [ $cont -eq 0 ]; then
            grupo=$n;
        elif [ $cont -eq 1 ]; then
            miembros=$n;
        fi
        let cont=$cont+1
    done < txt/$idioma/$1 #archivo que se le pasa al bucle

    l_grupos="";
    grupos=`awk -v p1=$grupo -v p2=$miembros '$3 >= 1000 && $3 <= 60000 {print p1, $1, p2, $4,
    "."}' FS=":" /etc/group`

    #IFS es la variable interna del bash que por defecto separa los campos por espacios en blanco
    IFS=. #Definimos que separe los campos por el caracter .
    for i in $grupos
    do
        l_grupos+="$i";
    done

}

#Funcion crear usuario
function crear_usuario()
{
    dir=""; #Variable que guarda el directorio home
    if [ $5 == "/home/_Username_" ]; then
        dir="/home/$1";
    else
        dir=$5;
    fi

    comprobar_contraseña $2 $3
    result=$?

    #Si las 2 contraseñas no coinciden salimos
    if [ $result == 1 ]; then
        return 100
    else
        #Se intenta crear el usuario
        useradd $1 -c $4 -md $dir -s $6 2> /dev/null
        result=$?

        #Errores useradd
        if [ $result == 9 ]; then
            #Nombre de usuario en uso
            return 101
        elif [ $result == 10 ]; then

```

```

        return 102
    elif [ $result == 11 ]; then
        return 103
    elif [ $result == 3 ]; then
        return 106
    else
        #Si todo va bien se usa passwd para que la contraseña
        #esté encriptada
        echo -e "$2\n$2" | (passwd $1) 2> /dev/null
        result=$?

        #Errores passwd
        if [ $result == 0 ]; then
            #Usuario creado correctamente
            if [ $7 == "Y" ] || [ $7 == "y" ] || [ $7 == "s" ] || [ $7 == "S" ];
            then
                bloquear_usuario $1;
            fi
            return 104
        elif [ $result == 1 ] || [ $result == 3 ] || [ $result == 4 ] || [ $result ==
5 ]; then
            return 105
        fi
    fi
fi

}

#Funcion eliminar usuario
function eliminar_usuario()
{
    #guardamos el directorio home del usuario
    userdel -r $1 2> /dev/null;
    result=$?;

    if [ $result == 1 ]; then
        #No se puede acceder al archivo de usuarios
        return 110
    elif [ $result == 6 ]; then
        #usuario no existe
        return 111
    elif [ $result == 8 ]; then
        #Usuario conectado
        return 112
    elif [ $result == 10 ]; then
        #No se puede acceder al archivo de grupos
        return 113
    elif [ $result == 12 ]; then
        #No se a podido borrar el directorio home
        return 116
    else
        #Borrado correctamente
        return 115
    fi
}

#Funcion crear grupo
function crear_grupo()
{
    groupadd $1
    result=$?

    if [ $result == 3 ]; then
        return 120;
    elif [ $result == 9 ]; then
        return 121;

```

```
        elif [ $result == 10 ]; then
            return 122;
        else
            return 124;
        fi
    }

#Funcion eliminar grupo
function eliminar_grupo()
{
    groupdel $1
    result=$?;

    if [ $result == 6 ]; then
        #No existe el grupo
        return 130
    elif [ $result == 8 ]; then
        #grupo con usuarios
        return 131
    elif [ $result == 10 ]; then
        #no se puede acceder
        return 132
    elif [ $result == 0 ]; then
        return 133
    fi
}

#Comprobar contraseña
function comprobar_contrasena()
{
    if [ $1 == $2 ]; then
        return 0;
    else
        return 1;
    fi
}

#Leer opciones
function leer_opciones() {
    idioma=`grep -w "idioma" txt/opciones.txt | cut -d: -f2`;
    shell=`grep -w "shell" txt/opciones.txt | cut -d: -f2`;
}

#Cambiar opciones
function cambiar_opciones() {
    echo idioma:$1 > txt/opciones.txt
    echo shell:/bin/$2 >> txt/opciones.txt
}
```