

MANUAL DE ADMINISTRACIÓN DE USUARIOS - LFS

TABLA DE CONTENIDOS

1. Ver Usuarios
 2. Crear Usuarios
 3. Eliminar Usuarios
 4. Modificar Usuarios
 5. Configuración Automática
 6. Solución de Problemas
-

VER USUARIOS

Comandos Básicos (cualquier usuario puede ejecutar):

Ver tu usuario actual
whoami

Ver tu UID, GID y grupos
id

Ver tus grupos
groups

Ver Todos los Usuarios (solo lectura):

Ver todos los usuarios
cat /etc/passwd

Solo nombres de usuario
cut -d: -f1 /etc/passwd

Usuarios normales (no sistema)
awk -F: '\$3 >= 1000 {print \$1}' /etc/passwd

Script para listar configuraciones de usuario(/usr/local/bin/list-users-config):

```
cat > /usr/local/bin/list-users-config << 'EOF'
#!/bin/bash
echo "=== CONFIGURACIÓN DE USUARIOS ==="
```

```
echo ""
```

```
for user in $(awk -F: '($3 >= 1000 {print $1})' /etc/passwd); do  
    home=$(getent passwd "$user" | cut -d: -f6)  
    echo "👤 $user (HOME: $home)"
```

```
for config in .bashrc .bash_profile .profile .inputrc; do  
    if [ -f "$home/$config" ]; then  
        lines=$(wc -l < "$home/$config")  
        size=$(stat -c%s "$home/$config")  
        echo "✅ $config: $lines líneas, $size bytes"  
    else  
        echo "❌ $config: No existe"  
    fi  
done  
echo ""  
done  
EOF  
chmod +x /usr/local/bin/list-users-config
```

Información Detallada de un Usuario:

```
# Información del usuario  
getent passwd nombre_usuario
```

```
# Información detallada (si está instalado)  
finger nombre_usuario 2>/dev/null
```

+ CREAR USUARIOS ⚠️ SOLO ROOT

ADVERTENCIA: Solo el usuario root puede crear nuevos usuarios.

Método Básico:

```
# Crear usuario con directorio /home  
useradd -m nombre_usuario
```

```
# Establecer contraseña  
passwd nombre_usuario
```

Opciones Avanzadas:

```
# Crear usuario con configuración completa  
useradd -m -s /bin/bash -c "Nombre Completo" -U nombre_usuario
```

```
# Con UID específico  
useradd -m -u 1001 nombre_usuario
```

```
# Con fecha de expiración
useradd -m -e 2026-12-31 nombre_usuario
```

```
# Con grupo primario específico
useradd -m -g usuarios nombre_usuario
```

Script de Creación Segura(/usr/local/bin/newuser):

```
cat > /usr/local/bin/newuser << 'EOF'
#!/bin/bash
# Crear nuevo usuario con configuración completa

if [ $# -eq 0 ]; then
    echo "Uso: $0 <nombre_usuario> [comentario]"
    exit 1
fi

USERNAME="$1"
COMMENT="${2:-Usuario creado el $(date +%Y-%m-%d)}"

echo "=== CREANDO NUEVO USUARIO: $USERNAME ==="
echo "Comentario: $COMMENT"
echo ""


# Verificar si ya existe
if id "$USERNAME" &>/dev/null; then
    echo "❌ Error: El usuario $USERNAME ya existe"
    exit 1
fi

# 1. Crear usuario
echo "1. Creando usuario..."
useradd -m -c "$COMMENT" -s /bin/bash "$USERNAME"
if [ $? -ne 0 ]; then
    echo "❌ Error al crear usuario"
    exit 1
fi
echo "✅ Usuario creado"

# 2. Establecer contraseña
echo "2. Estableciendo contraseña..."
passwd "$USERNAME"
if [ $? -ne 0 ]; then
    echo "⚠️ Advertencia: Problema con contraseña, pero usuario creado"
fi


# 3. Configurar grupos
```

```

echo "3. Configurando grupos..."
DEFAULT_GROUPS="users"
# Agregar a grupos adicionales si existen
for group in wheel audio video; do
    if getent group "$group" >/dev/null; then
        DEFAULT_GROUPS="$DEFAULT_GROUPS,$group"
    fi
done
usermod -aG "$DEFAULT_GROUPS" "$USERNAME"
echo "  Grupos: $DEFAULT_GROUPS"

# 4. Configurar directorios personales
echo "4. Configurando directorios..."
USER_HOME=$(getent passwd "$USERNAME" | cut -d: -f6)

# Crear directorios estándar
mkdir -p
"$USER_HOME"/{Documents,Downloads,Music,Pictures,Videos,Desktop,Public,Templates,
bin,.local/bin}
chown -R "$USERNAME:$USERNAME" "$USER_HOME"

# 5. Verificar configuración
echo "5. Verificando configuración..."
echo ""
echo "=== RESUMEN ==="
echo "Usuario: $USERNAME"
echo "UID: $(id -u "$USERNAME")"
echo "Home: $USER_HOME"
echo "Shell: $(getent passwd "$USERNAME" | cut -d: -f7)"
echo "Grupos: $(id -Gn "$USERNAME" | tr ' ' ',')"
echo ""
echo "Archivos creados en $USER_HOME:"
ls -la "$USER_HOME" | grep -E "^(-|d)" | head -10
echo ""
echo "  Usuario $USERNAME creado exitosamente"
echo "Para iniciar sesión: su - $USERNAME"
EOF
chmod +x /usr/local/bin/newuser

```

ELIMINAR USUARIOS SOLO ROOT

ADVERTENCIA CRÍTICA: Solo root puede eliminar usuarios.

Métodos de Eliminación:

Eliminar usuario PERO conservar su directorio /home

```
userdel nombre_usuario
```

```
# Eliminar usuario Y su directorio /home (irreversible)
```

```
userdel -r nombre_usuario
```

```
# Forzar eliminación (si el usuario está logeado)
```

```
userdel -f nombre_usuario
```

Proceso Seguro de Eliminación:

```
# Verificar que existe
```

```
id nombre_usuario
```

```
# Ver si está logeado
```

```
who | grep nombre_usuario
```

```
# Ver qué se eliminará
```

```
ls -la /home/nombre_usuario
```

```
# Luego eliminar
```

```
userdel -r nombre_usuario
```

Script de Eliminación Segura(/usr/local/bin/deleteuser):

```
cat > /usr/local/bin/deleteuser << 'EOF'
```

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

```
# Eliminar usuario de forma segura
```

```
if [ "$(id -u)" -ne 0 ]; then
```

```
    echo "✗ ERROR: Este script debe ejecutarse como root"
```

```
    echo "  Ejecuta: sudo $0 $@"
```

```
    exit 1
```

```
fi
```

```
if [ $# -eq 0 ]; then
```

```
    echo "Uso: $0 <nombre_usuario> [opciones]"
```

```
    echo ""
```

```
    echo "Opciones:"
```

```
    echo "  -k  Mantener directorio /home"
```

```
    echo "  -f  Forzar eliminación (si el usuario está logeado)"
```

```
    echo ""
```

```
    echo "Ejemplos:"
```

```
    echo "  $0 juan          # Eliminar usuario y su /home"
```

```
    echo "  $0 juan -k       # Eliminar usuario pero mantener /home"
```

```
    echo "  $0 juan -f       # Forzar eliminación"
```

```
    exit 1
```

```
fi
```

```
USER="$1"
```

```
KEEP_HOME=0
FORCE=0
```

```
# Procesar opciones
shift
while getopts "kf" opt; do
    case $opt in
        k) KEEP_HOME=1 ;;
        f) FORCE=1 ;;
        *) echo "Opción inválida: -$OPTARG" >&2; exit 1 ;;
    esac
done
```

```
echo "=== ELIMINACIÓN DE USUARIO: $USER ==="
echo ""
```

```
# Verificar si el usuario existe
if ! id "$USER" &>/dev/null; then
    echo "❌ Usuario '$USER' no existe"
    exit 1
fi
```

```
# Obtener información del usuario
USER_HOME=$(getent passwd "$USER" | cut -d: -f6)
USER_UID=$(id -u "$USER")
USER_GROUPS=$(id -Gn "$USER" | tr ' ' ',')
```

```
echo "📋 INFORMACIÓN DEL USUARIO:"
echo "  Nombre: $USER"
echo "  UID: $USER_UID"
echo "  Home: $USER_HOME"
echo "  Grupos: $USER_GROUPS"
echo ""
```

```
# Verificar si está logeado
LOGGED_IN=$(who | grep "^$USER" | wc -l)
if [ "$LOGGED_IN" -gt 0 ]; then
    echo "⚠️ ADVERTENCIA: El usuario $USER está logeado"
    who | grep "^$USER"
    echo ""
```

```
if [ "$FORCE" -eq 0 ]; then
    echo "❌ No se puede eliminar mientras esté logeado"
    echo "  Usa la opción -f para forzar o pídele que cierre sesión"
    exit 1
else
    echo "⚠️ Forzando eliminación..."
    pkill -u "$USER" 2>/dev/null
```

```

        sleep 2
    fi
fi

# Mostrar contenido del home
if [ -d "$USER_HOME" ]; then
    echo "📁 CONTENIDO DE $USER_HOME:"
    du -sh "$USER_HOME"
    echo "  Archivos principales:"
    ls -la "$USER_HOME" | head -10
    echo ""
fi

# Confirmación final
if [ "$KEEP_HOME" -eq 1 ]; then
    ACTION="eliminar usuario PERO MANTENER directorio $USER_HOME"
else
    ACTION="eliminar usuario y directorio $USER_HOME (IRREVERSIBLE)"
fi

echo "¿Estás seguro de querer $ACTION? (si/NO): "
read -r CONFIRM

if [[ ! "$CONFIRM" =~ ^[Ss][li]?$ ]]; then
    echo "❌ Operación cancelada"
    exit 0
fi

# Proceso de eliminación
echo ""
echo "🗑️ Eliminando usuario $USER..."

# Opciones para userdel
OPTS=""
if [ "$KEEP_HOME" -eq 1 ]; then
    echo " → Manteniendo directorio /home"
else
    OPTS="$OPTS -r"
    echo " → Eliminando directorio /home"
fi

if [ "$FORCE" -eq 1 ]; then
    OPTS="$OPTS -f"
    echo " → Modo forzado activado"
fi

# Ejecutar userdel
if userdel $OPTS "$USER"; then

```

```

echo ""
echo "✅ USUARIO ELIMINADO EXITOSAMENTE"
echo ""

# Información post-eliminación
if [ "$KEEP_HOME" -eq 1 ] && [ -d "$USER_HOME" ]; then
    echo "📁 Directorio $USER_HOME todavía existe"
    echo "  Propietario: $(stat -c '%U:%G' "$USER_HOME")"
    echo "  Tamaño: $(du -sh "$USER_HOME" | cut -f1)"
    echo ""
    echo "⚠️ NOTA: El directorio home se mantuvo pero ahora pertenece a root"
    echo "  Si quieres eliminarlo manualmente:"
    echo "    rm -rf $USER_HOME"
fi

echo "📋 RESUMEN:"
echo "  Usuario eliminado: $USER"
echo "  UID liberado: $USER_UID"
echo "  Grupos afectados: $USER_GROUPS"

else
    echo ""
    echo "❌ ERROR al eliminar el usuario"
    echo "  Intenta manualmente: userdel -r $USER"
    exit 1
fi
EOF
chmod +x /usr/local/bin/deleteuser

```

🔧 MODIFICAR USUARIOS ⚠️ SOLO ROOT

Cambiar Contraseña:

```

# Como root, cambiar contraseña de cualquier usuario
passwd nombre_usuario

```

```

# Como usuario normal, solo puedes cambiar TU PROPIA contraseña
passwd

```

Modificar Propiedades:

```

# Cambiar shell
usermod -s /bin/bash nombre_usuario

# Cambiar directorio home (y mover archivos)
usermod -d /nuevo/home -m nombre_usuario

```



```
# Agregar a grupos adicionales
usermod -aG audio,video,wheel nombre_usuario
```

```
# Cambiar UID
usermod -u 1005 nombre_usuario
```

```
# Cambiar comentario/descripción
usermod -c "Nuevo Nombre" nombre_usuario
```

Bloquear/Desbloquear Cuenta:

```
# Bloquear cuenta (no puede hacer login)
usermod -L nombre_usuario
passwd -l nombre_usuario
```

```
# Desbloquear cuenta
usermod -U nombre_usuario
passwd -u nombre_usuario
```

Configurar Fecha de Expiración:

```
# Establecer expiración
usermod -e 2026-12-31 nombre_usuario
```

```
# Ver fecha de expiración
chage -l nombre_usuario
```

```
# Quitar expiración
usermod -e "" nombre_usuario
```

Script de actualización de usuarios(/usr/local/bin/update-user-configs):

```
cat > /usr/local/bin/update-user-configs << 'EOF'
#!/bin/bash
# Actualizar configuraciones de usuarios existentes
# Versión mejorada - actualiza TODOS los archivos de configuración

SKEL_DIR="/etc/skel"
BACKUP_DIR="/tmp/user_backups_$(date +%Y%m%d_%H%M%S)"
LOG_FILE="/var/log/user-config-update.log"
CONFIG_FILES=".bashrc .bash_profile .profile .inputrc"

# Verificar si se ejecuta como root
if [ "$(id -u)" -ne 0 ]; then
    echo "❌ ERROR: Este script debe ejecutarse como root"
    echo "  Ejecuta: sudo $0"
    exit 1
fi
```

```

echo "=== ACTUALIZANDO CONFIGURACIONES DE USUARIOS ==="
echo "Fecha: $(date)"
echo ""

# Crear directorio de backups
mkdir -p "$BACKUP_DIR"
echo "📁 Backup de archivos originales en: $BACKUP_DIR"
echo ""

# Función para actualizar un usuario
update_user() {
    local user="$1"
    local home_dir=$(getent passwd "$user" | cut -d: -f6)

    if [ -z "$home_dir" ] || [ ! -d "$home_dir" ]; then
        echo "❌ Usuario $user: No tiene directorio home válido" | tee -a "$LOG_FILE"
        return 1
    fi

    echo "👤 Actualizando: $user ($home_dir)"
    echo "👤 Actualizando: $user ($home_dir)" >> "$LOG_FILE"

    # Crear directorio de backup para este usuario
    mkdir -p "$BACKUP_DIR/$user"

    # Procesar cada archivo de configuración
    for file in $CONFIG_FILES; do
        local skel_file="$SKEL_DIR/$file"
        local user_file="$home_dir/$file"

        # Verificar si existe la plantilla
        if [ ! -f "$skel_file" ]; then
            echo "⚠️ Plantilla $file no existe en $SKEL_DIR/"
            continue
        fi

        # Hacer backup del archivo actual si existe
        if [ -f "$user_file" ]; then
            cp "$user_file" "$BACKUP_DIR/$user/${file}.bak" 2>/dev/null
            echo "📄 Backup: $file guardado en $BACKUP_DIR/$user/${file}.bak"
        fi

        # ACTUALIZAR ARCHIVO (siempre, no solo si no existe)
        echo "🔄 Actualizando: $file"

        # Opción 1: Reemplazar completamente (comentada por seguridad)
        # cp "$skel_file" "$user_file"
    done
}

```

```

# Opción 2: Mejor - Copiar solo si el usuario no lo ha modificado mucho
if [ ! -f "$user_file" ] || [ $(wc -l < "$user_file" 2>/dev/null || echo 0) -lt 5 ]; then
    # Archivo no existe o es muy pequeño, reemplazar
    cp "$skel_file" "$user_file"
    echo " 🟢 Reemplazado: $file (archivo pequeño/inexistente)"
else
    # Archivo existe y tiene contenido, preguntar
    echo " ⚠️ $file ya existe con contenido. ¿Reemplazar? (s=si, n=no, d=diff): "
    read -n 1 -r choice
    echo
    case $choice in
        s|S)
            cp "$skel_file" "$user_file"
            echo " 🟢 Reemplazado: $file"
            ;;
        d|D)
            echo " 📊 Diferencias entre plantilla y archivo actual:"
            diff -u "$user_file" "$skel_file" | head -20
            echo " ¿Reemplazar igual? (s/n): "
            read -n 1 -r replace
            echo
            if [[ $replace =~ ^[Ss]$ ]]; then
                cp "$skel_file" "$user_file"
                echo " 🟢 Reemplazado: $file"
            else
                echo " ⏏️ Saltado: $file (se mantiene original)"
            fi
            ;;
        *)
            echo " ⏏️ Saltado: $file (se mantiene original)"
            ;;
    esac
fi

# Establecer permisos correctos
if [ -f "$user_file" ]; then
    chown "$user:$user" "$user_file"
    chmod 644 "$user_file"
fi


done

# Crear directorios útiles si no existen
for dir in bin Documents Downloads .local/bin; do
    if [ ! -d "$home_dir/$dir" ]; then
        mkdir -p "$home_dir/$dir"
        chown "$user:$user" "$home_dir/$dir"
        echo " 📁 Creado directorio: $dir"
    fi
done

```


```

        fi
    done


    echo "  Usuario $user actualizado"
    echo ""
}




# Registrar inicio en log
echo "Inicio de actualización: $(date)" > "$LOG_FILE"
echo "Backup en: $BACKUP_DIR" >> "$LOG_FILE"
echo "" >> "$LOG_FILE"

# Obtener lista de usuarios
echo "Obteniendo lista de usuarios..."
USERS=$(awk -F: ' $3 >= 1000 {print $1}' /etc/passwd)
USER_COUNT=$(echo "$USERS" | wc -w)

echo " Se procesarán $USER_COUNT usuarios"
echo "Usuarios: $USERS"
echo ""

# Actualizar todos los usuarios normales
COUNTER=0
for user in $USERS; do
    COUNTER=$((COUNTER + 1))
    echo "--- [{COUNTER}/$USER_COUNT] ---"
    update_user "$user"
done

# Preguntar sobre root
echo ""
echo "=== USUARIO ROOT ==="
echo "¿Actualizar configuración de root? (s/n): "
read -n 1 -r
echo
if [[ $REPLY =~ ^[Ss]$ ]]; then
    update_user "root"
else
    echo " Saltando actualización de root"
fi

# Resumen final
echo ""
echo "=== RESUMEN FINAL ==="
echo " Proceso completado"
echo " Backups guardados en: $BACKUP_DIR"
echo " Log detallado en: $LOG_FILE"
echo ""

```

```

echo "📊 Estadísticas:"
echo " - Usuarios procesados: $COUNTER"
echo " - Directorio de backup: $BACKUP_DIR"

# Mostrar contenido del directorio de backup
echo ""
echo "📁 Contenido del directorio de backup:"
ls -la "$BACKUP_DIR"/* 2>/dev/null | head -20

# Sugerencia para revertir cambios
echo ""
echo "💡 Para revertir cambios de un usuario específico:"
echo "  cp $BACKUP_DIR/nombre_usuario/.bashrc.bak /home/nombre_usuario/.bashrc"
echo "  chown nombre_usuario:nombre_usuario /home/nombre_usuario/.bashrc"
EOF

chmod +x /usr/local/bin/update-user-configs

```

⚙️ CONFIGURACIÓN AUTOMÁTICA

Configurar Nuevos Usuarios Automáticamente:

Los archivos en `/etc/skel/` se copian automáticamente a nuevos usuarios.

```

# Como root, configurar plantillas
mkdir -p /etc/skel

```

Configurar .bashrc:

```

cat > /etc/skel/.bashrc << 'EOF'
# ~/.bashrc
export PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:$HOME/bin"
export EDITOR=vi

# Configurar prompt
if [ "$(id -u)" -eq 0 ]; then
    PS1="\[\033[01;31m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]# "
else
    PS1="\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]$ "
fi

# Alias básicos
alias ll='ls -la'
alias la='ls -A'
alias l='ls -CF'

```

```
alias ls='ls --color=auto'
alias grep='grep --color=auto'
alias fgrep='fgrep --color=auto'
alias egrep='egrep --color=auto'
```

Alias de navegación

```
alias ..='cd ..'
alias ...='cd ../..'
alias ....='cd ../../..'
```

Alias de seguridad

```
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
```

Alias del sistema

```
alias df='df -h'
alias du='du -h'
alias mkdir='mkdir -pv'
alias h='history'
alias ports='netstat -tulnp'
```

Alias de paquetes (si usas apt)

```
alias update='apt-get update && apt-get upgrade'
alias install='apt-get install'
alias remove='apt-get remove'
```

Configuración del historial

```
HISTSIZE=1000
HISTFILESIZE=2000
HISTCONTROL=ignoreboth
shopt -s histappend
```

PATH para directorios locales

```
if [ -d "$HOME/.local/bin" ]; then
    PATH="$HOME/.local/bin:$PATH"
fi
EOF
```

Configurar .bash_profile:

```
cat > /etc/skel/.bash_profile << 'EOF'
# ~/.bash_profile
[ -f ~/.bashrc ] && . ~/.bashrc
```

Mensaje de inicio de sesión

```
if [ "$(id -u)" -eq 0 ]; then
```

```
    echo "Sesión root iniciada"
else
    echo "Sesión de $(whoami) iniciada"
fi

# Configuraciones adicionales para login shells
export LANG=es_ES.UTF-8
export LC_ALL=es_ES.UTF-8

# Variables de entorno adicionales
export PAGER=less
export VISUAL=vi

# Configuración de seguridad
umask 022

# Timeout de sesión (opcional)
# export TMOUT=300
EOF
```

Configurar .profile (para compatibilidad con otros shell):

```
cat > /etc/skel/.profile << 'EOF'
# ~/.profile
# Este archivo es para compatibilidad con otros shells

# Si existe bash, cargar .bashrc
if [ -n "$BASH_VERSION" ]; then
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# Configurar PATH para directorios locales
if [ -d "$HOME/bin" ]; then
    PATH="$HOME/bin:$PATH"
fi

if [ -d "$HOME/.local/bin" ]; then
    PATH="$HOME/.local/bin:$PATH"
fi

# Variables de entorno básicas
export EDITOR=vi
export VISUAL=vi
EOF
```

SOLUCIÓN DE PROBLEMAS

Problema: "Permission denied" al crear usuario

```
useradd juan
useradd: Permission denied.
```

```
# Solución: Necesitas ser root
su -
useradd -m juan
```

Problema: Usuario no puede hacer login

```
# Verificar como root
grep juan /etc/passwd
passwd -S juan
ls -ld /home/juan
chsh -l juan # Ver shells disponibles
```

Problema: Directorio /home no se creó

```
# Crear manualmente
mkdir /home/nombre_usuario
cp /etc/skel/* /home/nombre_usuario/ 2>/dev/null || true
chown -R nombre_usuario:nombre_usuario /home/nombre_usuario
chmod 700 /home/nombre_usuario
```

Problema: "user currently logged in"

```
# Ver sesiones activas
who
pkill -u nombre_usuario # Terminar sesiones (cuidado)
userdel -r nombre_usuario # Luego eliminar
```

TABLA DE REFERENCIA RÁPIDA

| Comando | Usuario | Descripción | Ejemplo |
|--------------------------|---------|----------------------------|--------------------------------|
| <code>whoami</code> | Todos | Ver usuario actual | <code>\$ whoami</code> |
| <code>id</code> | Todos | Ver UID, GID, grupos | <code>\$ id</code> |
| <code>passwd</code> | Propio | Cambiar tu contraseña | <code>\$ passwd</code> |
| <code>passwd user</code> | Root | Cambiar contraseña de otro | <code># passwd juan</code> |
| <code>useradd</code> | Root | Crear usuario | <code># useradd -m juan</code> |






| | | | |
|----------------------|--------|------------------------|---------------------------------------|
| <code>userdel</code> | Root | Eliminar usuario | <code># userdel juan</code> |
| <code>usermod</code> | Root | Modificar usuario | <code># usermod -aG wheel juan</code> |
| <code>groups</code> | Todos | Ver grupos del usuario | <code>\$ groups</code> |
| <code>chsh</code> | Propio | Cambiar tu shell | <code>\$ chsh -s /bin/bash</code> |

ESTRUCTURA DE ARCHIVOS IMPORTANTES




| Archivo | Propósito | Permisos |
|--------------------------|---------------------------|-----------------|
| <code>/etc/passwd</code> | Información de usuarios | Lectura pública |
| <code>/etc/shadow</code> | Contraseñas encriptadas | Solo root |
| <code>/etc/group</code> | Grupos del sistema | Lectura pública |
| <code>/etc/skel/</code> | Plantilla nuevos usuarios | Root |
| <code>/home/</code> | Directorios home usuarios | Root |

BUENAS PRÁCTICAS

Para Administradores (root):

1.  Siempre verifica antes de eliminar: `id usuario`
2.  Usa `userdel -r` solo cuando estés seguro
3.  Mantén UIDs consistentes (1000-1999 para usuarios normales)
4.  Usa grupos para gestionar permisos
5.  Documenta usuarios creados/eliminados

Para Usuarios Normales:

1.  Solo modifica TU PROPIA configuración
2.  Usa `passwd` regularmente
3.  Reporta problemas al administrador

4.  No compartas tu cuenta
-

FLUJO DE TRABAJO RECOMENDADO

Crear Usuario:

```
su -  
useradd -m -s /bin/bash juan  
passwd juan  
usermod -aG usuarios juan  
exit
```

Eliminar Usuario:

```
su -  
id juan  
userdel -r juan  
exit
```

Promover a Administrador:

```
su -  
usermod -aG wheel juan  
exit
```

ADVERTENCIAS IMPORTANTES

1. **NUNCA elimines el usuario `root`**
 2. **Los `UIDs` bajos (`<1000`) son usuarios del sistema**
 3. **`/etc/shadow` es sensible - solo `root` puede leerlo**
 4. **Haz `backup` antes de modificaciones masivas**
 5. **Prueba en un usuario de prueba primero**
-

¿NECESITAS AYUDA?

Flujo de Diagnóstico:

1. Identifica tu usuario: `$ whoami`
2. Determina qué necesitas hacer:
 - ¿Crear usuario? → Necesitas ser **root**
 - ¿Cambiar tu contraseña? → Usa `$ passwd`

- ¿Ver usuarios? → `$ cat /etc/passwd`

Preguntas Comunes:

- ¿Por qué no puedo crear usuarios? → No eres root
 - ¿Cómo me hago administrador? → Solo root puede agregarte al grupo wheel
 - ¿Dónde se guardan las contraseñas? → En `/etc/shadow` (solo root)
 - ¿Cómo recupero mi contraseña? → Contacta al administrador (root)
-



SITUACIONES DE EMERGENCIA

Acceso de Emergencia a root si olvidaste contraseña:

1. Arrancar en modo recovery/single-user
2. Montar sistema de archivos: `mount -o remount,rw /`
3. Cambiar contraseña: `passwd root`
4. Reiniciar

Bloquear Usuario Rápidamente:

`passwd -l usuario_problematico`

Ver Todos los Logins de un Usuario:

`last usuario`

`lastlog -u usuario`



Este manual cubre el 95% de las operaciones de usuario en LFS. Para casos especiales, consulta `man useradd`, `man userdel`, `man usermod`.