

logo_unime.jpg

UNIVERSITA' DEGLI STUDI DI MESSINA

**DIPARTIMENTO DI SCIENZE MATEMATICHE E INFORMATICHE,
SCIENZE FISICHE E SCIENZE DELLA TERRA**

Corso di Laurea Triennale in Informatica

Progetto Laboratorio di Amministrazione dei Sistemi

Giuseppe Allegra
527304

Giovanni Pagano
530975

ANNO ACCADEMICO 2024/25

Indice

1	Descrizione del problema	3
1.1	Soluzione proposta	3
2	Implementazione	4
2.1	Creazione di un Utente (<code>create_user</code>)	4
2.2	Modifica Utente (<code>confirm_mod_user</code>)	5
2.3	Eliminazione di un Utente (<code>delete_non_root_users</code>)	7
2.4	Creazione di un Gruppo (<code>create_group</code>)	7
2.5	Eliminazione di un Gruppo (<code>delete_group</code>)	7
3	GUI	8
4	Risultati ottenuti	10
5	Sviluppi Futuri	10

1 Descrizione del problema

L'amministrazione degli utenti in un sistema Ubuntu richiede l'uso di comandi complessi che spesso non risultano intuitivi per chi non ha familiarità con la linea di comando. Le operazioni comuni di gestione utenti, come creare, modificare o eliminare account, aggiungere utenti a gruppi e altre funzionalità simili, richiedono competenze specifiche e una buona conoscenza dei comandi di sistema. Questo crea una barriera per gli amministratori meno esperti o per situazioni in cui è preferibile un'interfaccia visiva.

1.1 Soluzione proposta

Il programma implementa una GUI (interfaccia grafica) per facilitare la gestione degli utenti su Ubuntu. La GUI è costruita con Tkinter e CustomTkinter, permettendo agli amministratori di eseguire azioni comuni senza la necessità di inserire comandi complessi da terminale. Le caratteristiche della soluzione includono:

- **Esecuzione di comandi di sistema:** L'interfaccia semplifica l'esecuzione di comandi di amministrazione, mostrando i risultati direttamente nella GUI.
- **Miglioramento dell'esperienza utente:** Attraverso la presentazione di notifiche e messaggi di errore in caso di input non validi, la GUI rende l'esperienza di gestione più chiara e intuitiva.

Questa soluzione è particolarmente adatta a semplificare e velocizzare le operazioni di amministrazione, rendendole accessibili anche a utenti non esperti di comandi di sistema.

2 Implementazione

L'applicazione `UbuntuCommandManager` fornisce un'interfaccia grafica per semplificare la gestione degli utenti e dei gruppi su un sistema Ubuntu. Utilizza `tkinter` per creare la GUI e la libreria `subprocess` per eseguire comandi di sistema senza l'uso diretto del terminale.

Di seguito viene spiegata l'implementazione dei comandi principali.

2.1 Creazione di un Utente (`create_user`)

Creazione Utente

```
command = ["sudo", "useradd", "-m", username]
```

- **Comando:** `sudo useradd -m <username>`

Impostazione Password

```
passwd_command = ["sudo", "passwd", username]
```

- **Comando:** `sudo passwd <username>`

La funzione di creazione di un utente nel sistema Linux prevede una serie di controlli e operazioni per assicurare che il nuovo utente sia creato in modo sicuro e corretto. Ecco come è strutturata:

-Validazione del Nome Utente:

Il nome utente viene richiesto tramite una finestra di dialogo e verificato per assicurarsi che rispetti i requisiti di base, come l'uso di soli caratteri alfanumerici e una lunghezza minima di 3 caratteri. Se questi criteri non sono soddisfatti, viene mostrato un messaggio di errore.

-Richiesta della Password:

Una volta confermato il nome utente, l'applicazione richiede una password per il nuovo utente. La password è richiesta due volte (come input nascosto, per sicurezza) per evitare errori di battitura, ma in questo caso è fornita direttamente dal flusso di input in `subprocess`

-Esecuzione del Comando di Creazione:

Il comando `sudo useradd -m <username>` viene eseguito per creare un utente con una nuova directory `home`. Questo comando viene inviato tramite la libreria `subprocess.run`, che restituisce `stdout` e `stderr` per confermare se l'operazione è avvenuta con successo.

Se il comando è eseguito senza errori, il programma salva il comando nella cronologia e procede al passo successivo per impostare la password. Impostazione della Password:

Una volta creato l'utente, il programma imposta la password tramite il comando `passwd`. Utilizzando `subprocess.Popen`, la password viene inviata direttamente come input tramite `stdin` (la password viene ripetuta per conferma).

Se l'operazione è eseguita correttamente, viene mostrato un messaggio di successo e la lista degli utenti è aggiornata nell'interfaccia grafica.

2.2 Modifica Utente (`confirm_mod_user`)

A seconda dell'opzione selezionata, si eseguono i seguenti comandi:

- **Blocco dell'Utente:**

```
command = ["sudo", "usermod", "-L", selected_user]
```

- **Comando:** `sudo usermod -L <username>`
- **Descrizione:** Blocca l'account dell'utente.

- **Sblocco dell'Utente:**

```
command = ["sudo", "usermod", "-U", selected_user]
```

- **Comando:** `sudo usermod -U <username>`
- **Descrizione:** Sblocca l'account dell'utente.

- **Cambio Password:**

```
passwd_command = ["sudo", "passwd", selected_user]
process = subprocess.Popen(passwd_command, stdin=
    subprocess.PIPE, stdout=subprocess.PIPE, stderr=
    subprocess.PIPE, text=True)
process.communicate(input=f"{new_password}\n{
    new_password}\n")
```

- **Comando:** `sudo passwd <username>`
- **Descrizione:** La funzione di modifica della password richiede un input di sicurezza per assicurare che solo utenti autorizzati possano cambiare la password:

Selezione Utente:

L'amministratore seleziona l'utente desiderato dalla lista.

Richiesta della Nuova Password:

Viene richiesto all'amministratore di inserire una nuova password tramite una finestra di dialogo. La password è inserita come input nascosto.

Esecuzione del Comando di Cambio Password:

Per modificare la password, l'applicazione utilizza `sudo passwd {username}`, che richiede la password due volte per conferma. Questa funzione è gestita tramite `subprocess.Popen`, con la password inviata come input tramite `stdin`. L'operazione è eseguita e il risultato è mostrato nell'interfaccia. Un messaggio di successo viene visualizzato se il comando è completato senza errori.

- **Cambio Gruppo:**

```
command = ["sudo", "usermod", "-g", new_group,
    selected_user]
```

- **Comando:** `sudo usermod -g <new_group> <username>`
- **Descrizione:** Cambia il gruppo principale dell'utente.

- **Cambio Nome Utente:**

```
command = ["sudo", "usermod", "-l", new_username,
    selected_user]
```

- **Comando:** `sudo usermod -l <new_username> <username>`
- **Descrizione:** Cambia il nome utente e rinomina la directory home.

2.3 Eliminazione di un Utente (delete_non_root_users)

```
command = ["sudo", "userdel", "-r", selected_user]
```

- **Comando:** `sudo userdel -r <username>`
- **Descrizione:** Elimina l'utente e la directory home.

2.4 Creazione di un Gruppo (create_group)

```
command = ["sudo", "groupadd", group_name]
```

- **Comando:** `sudo groupadd <group_name>`
- **Descrizione:** Crea un nuovo gruppo.

2.5 Eliminazione di un Gruppo (delete_group)

```
command = ["sudo", "groupdel", group_name]
```

- **Comando:** `sudo groupdel <group_name>`
- **Descrizione:** Elimina un gruppo esistente.

3 GUI

Eccome come si presenta la GUI all'esecuzione dei comandi:

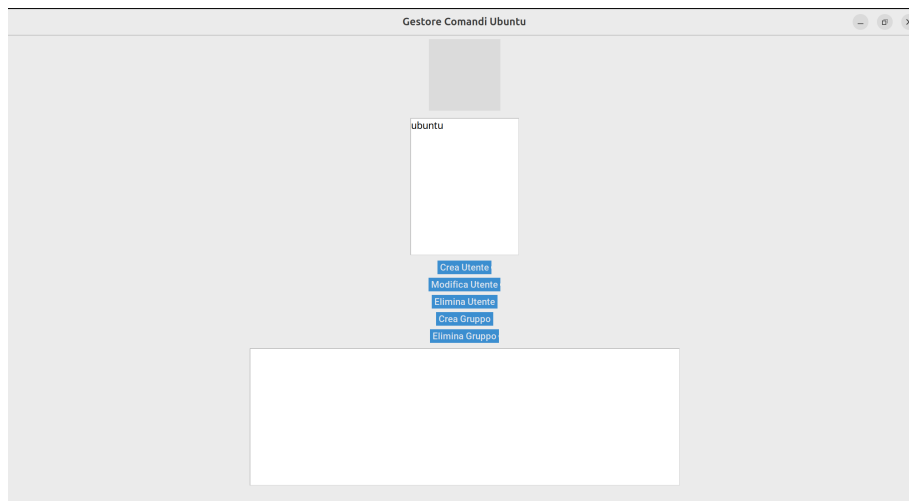


Figura 1: GUI di partenza

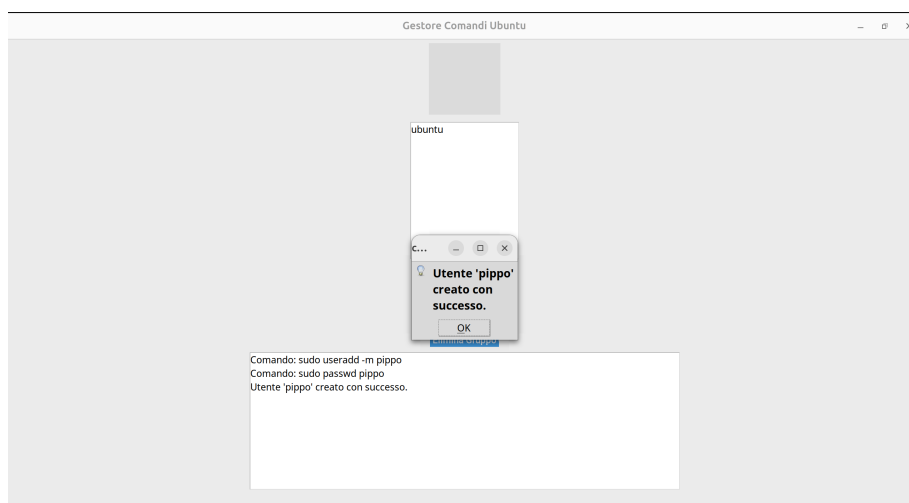


Figura 2: Creazione di un utente

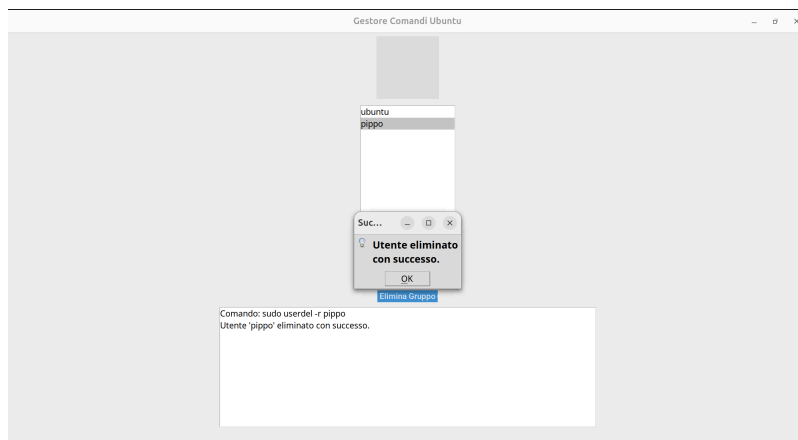


Figura 3: Eliminazione di un utente

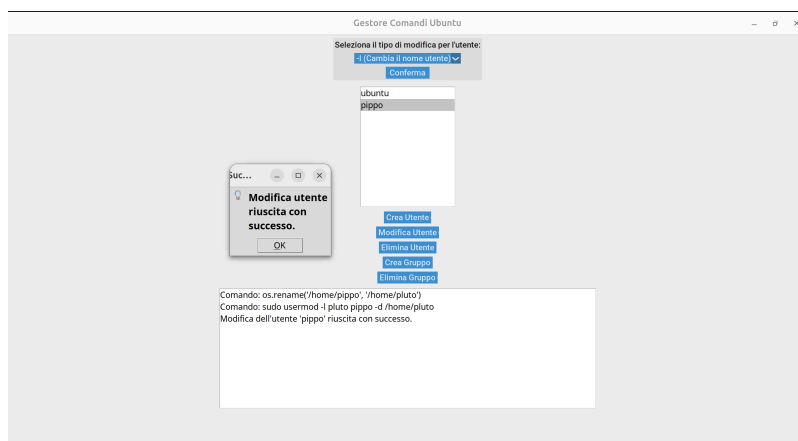


Figura 4: Esempio modifica di un utente: modifica del nome

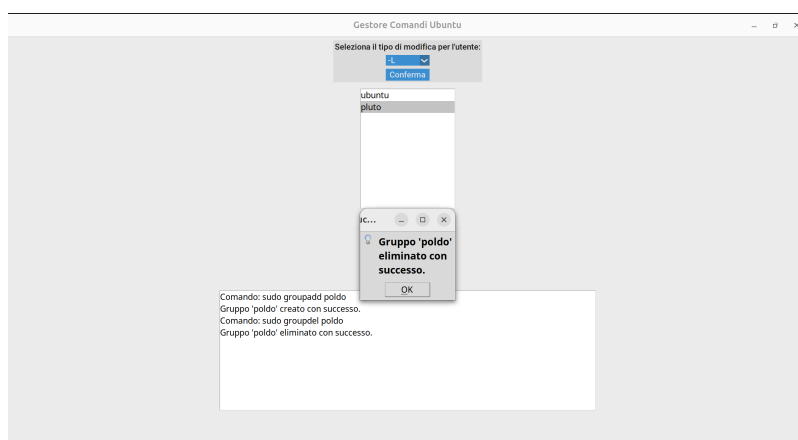


Figura 5: Creazione/Eliminazione di un gruppo

4 Risultati ottenuti

La soluzione implementata con il programma **UbuntuCommandManager** ha raggiunto i seguenti risultati:

-Semplificazione della Gestione Utenti: Il sistema consente ora di creare e gestire utenti con facilità, riducendo il margine di errore legato ai comandi manuali da terminale. L'interfaccia grafica offre un'esperienza più intuitiva e guidata, particolarmente utile per utenti meno esperti. **-Interazione Efficace con il Sistema:** La GUI permette di visualizzare rapidamente i risultati dei comandi, facilitando la verifica delle operazioni. Inoltre, l'output immediato rende più rapido il feedback per l'amministratore, che può così intervenire in caso di errori.

5 Sviluppi Futuri

Per arricchire le funzionalità e migliorare l'usabilità, si potrebbe considerare l'implementazione dei seguenti sviluppi:

-Gestione avanzata dei Gruppi: Implementare una sezione per la gestione dei gruppi, che permetta modificare e visualizzare i gruppi esistenti, nonché di assegnare utenti a specifici gruppi direttamente dalla GUI.

-Supporto Multi-Piattaforma: Ampliare la compatibilità del programma per altri sistemi Linux oltre a Ubuntu, come Fedora o Debian, includendo controlli per l'adattamento dei comandi specifici per ciascuna distribuzione.

-Funzionalità di Backup e Ripristino: Aggiungere funzionalità per creare backup delle configurazioni utente e per ripristinare facilmente configurazioni precedenti.

-Integrazione della Gestione dei Permessi: Sviluppare un'interfaccia che consenta di assegnare e modificare i permessi e i privilegi degli utenti, ampliando così le capacità amministrative della GUI.