

# Manuale di Utilizzo - Sistema di Gestione Distribuzione Acqua

---

## Prerequisiti del Sistema

### Software Necessario

- **Python 3.8 o superiore** installato sul computer
- **Git** per scaricare il progetto
- **Browser web** moderno (Chrome, Firefox, Safari, Edge)

### Verifica Python

Aprire il Prompt dei Comandi (cmd) e digitare:

```
python --version
```

Se il comando non funziona, provare:

```
python3 --version
```

### Installazione Python (se mancante)

1. Andare su <https://www.python.org/downloads/>
2. Scaricare l'ultima versione stabile
3. Installare seguendo le istruzioni
4. Riavviare il computer

## Installazione del Progetto

### 1. Scaricare il Progetto

Creare una cartella per il progetto:

```
mkdir progetti_django  
cd progetti_django
```

Se disponibile tramite Git:

```
git clone [URL_DEL_REPOSITORY]  
cd progetto_2_una_acies
```

## 2. Installare Django

```
pip install django
```

Se il comando non funziona:

```
pip3 install django
```

## 3. Configurare il Database

Navigare nella cartella del progetto e eseguire:

```
python manage.py migrate
```

# Avvio del Progetto

## 1. Avviare il Server di Sviluppo

```
python manage.py runserver
```

## 2. Accedere all'Applicazione

- Aprire il browser web
- Digitare l'indirizzo: `http://127.0.0.1:8000`
- Il sistema mostrerà la pagina principale

## 3. Interruzione del Server

- Premere `Ctrl + C` nel Prompt dei Comandi
- Il server si arresterà

# Utilizzo del Sistema

## Navigazione Principale

Il sistema presenta tre sezioni principali:

### 1. Clienti - Solo visualizzazione

- Visualizzare elenco clienti
- Filtrare per città e ragione sociale
- Consultare dettagli anagrafica

## 2. Utenze - Gestione completa

- Visualizzare tutte le utenze
- Aggiungere nuove utenze
- Modificare utenze esistenti
- Eliminare utenze
- Filtrare per cliente, città, stato e date

## 3. Fatture - Solo visualizzazione

- Consultare elenco fatture
- Filtrare per date e importi
- Visualizzare dettagli fatturazione

## Funzionalità Principali

### Gestione Utenze

#### 1. Visualizzazione

- Cliccare su "Utenze" nel menu
- Utilizzare i filtri per ricerche specifiche
- Cliccare "Filtra" per applicare i criteri
- Cliccare "Reset" per eliminare i filtri

#### 2. Aggiunta Nuova Utenza

- Cliccare "Nuova Utenza"
- Compilare tutti i campi obbligatori
- Selezionare il cliente dall'elenco
- Scegliere lo stato (attivo/inattivo)
- Cliccare "Salva"

#### 3. Modifica Utenza

- Individuare l'utenza nell'elenco
- Cliccare l'icona "Modifica" (matita)
- Modificare i campi necessari
- Cliccare "Salva"

#### 4. Eliminazione Utenza

- Individuare l'utenza nell'elenco
- Cliccare l'icona "Elimina" (cestino)
- Confermare l'eliminazione
- **Attenzione:** Non sarà possibile eliminare utenze con letture associate

### Gestione Letture

#### 1. Visualizzazione Letture

- Dall'elenco utenze, cliccare l'icona "Visualizza letture"
- Consultare lo storico delle letture

## 2. Aggiunta Nuova Lettura

- Nella pagina letture utenza, cliccare "Nuova Lettura"
- Inserire data e valore
- Selezionare fattura (opzionale)
- Cliccare "Salva"

## Filtri e Ricerche

- **Filtro Clienti:** Cercare per ragione sociale o codice fiscale
- **Filtro Città:** Inserire nome città
- **Filtro Stato:** Selezionare attivo/inattivo
- **Filtro Date:** Utilizzare selettore calendario
- **Filtro Importi:** Inserire valori numerici

# Risoluzione Problemi

## Errori Comuni

### 1. "Python non riconosciuto"

**Problema:** Il sistema non trova Python **Soluzione:**

- Verificare installazione Python
- Aggiungere Python al PATH di sistema
- Riavviare il Prompt dei Comandi

### 2. "Django non installato"

**Problema:** Modulo Django mancante **Soluzione:**

```
pip install django
```

### 3. "Porta già in uso"

**Problema:** Server già attivo **Soluzione:**

- Terminare processi Python attivi
- Utilizzare porta alternativa:

```
python manage.py runserver 8001
```

### 4. "Database bloccato"

**Problema:** Database non accessibile **Soluzione:**

- Chiudere tutte le istanze dell'applicazione
- Eliminare file `db.sqlite3`
- Ricreare database:

```
python manage.py migrate
```

## 5. "Errore di migrazione"

**Problema:** Database non sincronizzato **Soluzione:**

```
python manage.py makemigrations  
python manage.py migrate
```

## Casi Avversi

### Perdita Dati

- Il sistema utilizza database SQLite locale
- Effettuare backup regolari del file `db.sqlite3`
- Conservare copie di sicurezza

### Prestazioni Lente

- Verificare disponibilità memoria RAM
- Chiudere applicazioni non necessarie
- Riavviare il server Django

### Errori di Visualizzazione

- Aggiornare la pagina browser (F5)
- Svuotare cache browser
- Verificare connessione internet per risorse esterne

## Manutenzione

### Backup dei Dati

1. Individuare il file `db.sqlite3`
2. Copiarlo in posizione sicura
3. Rinominarlo con data (es: `db_backup_2025-01-XX.sqlite3`)

### Aggiornamenti

- Verificare periodicamente aggiornamenti Django

- Testare modifiche in ambiente di sviluppo
- Effettuare backup prima di aggiornamenti

## Riavvio Sistema

Per riavviare completamente:

1. Terminare server (Ctrl + C)
2. Chiudere Prompt dei Comandi
3. Riaprire e navigare nella cartella progetto
4. Eseguire `python manage.py runserver`

## Contatti e Supporto

In caso di problemi tecnici non risolti:

- Verificare file di log per errori specifici
- Consultare documentazione Django ufficiale
- Contattare supporto tecnico fornendo dettagli errore

---

**Importante:** Mantenere sempre backup aggiornati dei dati e testare modifiche in ambiente di sviluppo prima dell'implementazione.