

ACQUEDOTTO

DOCUMENTAZIONE TECNICA

Sistema di Pianificazione

Gestione Turni Acquedotto

Versione: 1.0

Data: 01/02/2026

Deployment: <https://planner.danielesturla.it>

Stack: React + FastAPI + Baserow + Docker

INDICE

1. Panoramica del Progetto	3
2. Architettura Tecnica	4
3. Funzionalità Implementate	6
4. Configurazione Baserow	11
5. File di Configurazione Completi	13
6. Deployment e Manutenzione	16
7. Guida Utente	18
8. Guida Amministratore	21
9. Troubleshooting	23
10. Cronologia Modifiche	25

1. PANORAMICA DEL PROGETTO

1.1 Obiettivo

Sistema web-based per la gestione della pianificazione settimanale dei turni degli operatori dell'acquedotto. Permette ai responsabili di team di organizzare le attività giornaliere (mattina/pomeriggio) per ciascun operatore, con funzionalità avanzate di collaborazione e visualizzazione.

1.2 Caratteristiche Principali

- Pianificazione settimanale con vista tabellare
- Gestione multi-responsabile con permessi granulari
- Note giornaliere condivise con attributi di authorship
- Modalità superuser per amministratori
- Export PDF per reportistica
- Integrazione meteo in tempo reale
- Responsive design per desktop e mobile
- Filtri personalizzati (team, weekend)

1.3 Utenti del Sistema

Ruolo	Descrizione	Permessi
Manager	Responsabile di team	Gestisce il proprio team
Superuser	Amministratore	Gestisce tutti i team
Operatore	Dipendente	Visualizza solo (no accesso diretto)

2. ARCHITETTURA TECNICA

2.1 Stack Tecnologico

Componente	Tecnologia	Versione
Frontend	React 18.2	Single-page HTML
Backend	Python FastAPI	Latest
Database	Baserow Cloud	API-based
Containerizzazione	Docker + Portainer	Latest
Web Server	Uvicorn	Built-in FastAPI
Styling	Tailwind CSS	CDN 3.x
Export	jsPDF + html2canvas	2.5.1 / 1.4.1
Weather API	Open-Meteo	Free tier

2.2 Struttura Applicazione

```
Struttura Directory:  
/opt/pianificazione-app/  
  
backend/  
    main.py (API FastAPI)  
    .env (configurazione)  
    Dockerfile  
    requirements.txt  
  
frontend/  
    index.html (React single-file)  
  
docker-compose.yml
```

2.3 Deployment

Server: VPS con Docker e Portainer
URL Pubblico: <https://planner.danielesturla.it>
SSL: Certificato gestito automaticamente
Backup: Database Baserow in cloud (automatico)

2.4 Flusso Dati

1. **Utente** → Interagisce con interfaccia React
2. **React** → Effettua chiamate API a FastAPI backend
3. **FastAPI** → Comunica con Baserow API (CRUD operations)
4. **Baserow** → Memorizza dati persistenti in cloud
5. **FastAPI** → Ritorna dati a React in formato JSON
6. **React** → Aggiorna interfaccia utente

3. FUNZIONALITÀ IMPLEMENTATE

3.1 Pianificazione Base

- Vista settimanale con 7 colonne (Lun-Dom)
- Due righe per operatore (Mattina/Pomeriggio)
- Salvataggio automatico onChange/onBlur
- Cancellazione automatica celle vuote
- Celle readonly per operatori non del proprio team
- Evidenziazione weekend e festività italiane 2026
- Auto-evidenziazione celle FERIE/ASSENTE

3.2 Note Giornaliere Multiple

Sistema di note condivise per giorno con supporto multi-autore:

- Ogni manager può aggiungere proprie note per ogni giorno
- Visualizzazione a lista con badge autore
- Modifica/cancellazione solo delle proprie note
- Indicatore visivo: emoji [*] + conteggio + lista autori
- Popup modale per gestione completa
- Persistenza su tabella Baserow dedicata

3.3 Gestione Testi Lunghi

Per celle con testo > 50 caratteri:

- Indicatore visivo '...' in basso a destra
- Desktop: Tooltip al hover con testo completo
- Mobile/Desktop: Click apre popup di modifica
- Popup modale con textarea espansa
- Salvataggio/Annulla nel popup

3.4 Superuser (Amministratore)

Modalità speciale per amministratori con campo **is_superuser=true** in tabella managers:

- Badge 'ADMIN' rosso visibile nell'header
- Può modificare TUTTE le celle di TUTTI i team
- Bypassaggio controlli di permessi team-based
- Utile per supervisione e correzioni urgenti

3.5 Filtri e Visualizzazione

- **Solo mio team:** Mostra solo operatori del manager loggato
- **Mostra weekend:** Nasconde/mostra colonne Sabato e Domenica
- Navigazione settimane: Frecce ← → per scorrere
- Colonne weekend più strette (70px vs 143px desktop)
- Layout responsive con larghezze ottimizzate mobile

3.6 Integrazione Meteo

Integrazione in tempo reale con API Open-Meteo per coordinate di Pavia (45.18N, 9.15E):

- Icône meteo per ogni giorno: [SOLE] [NUVOLE] [PIOGGIA] [NEVE] [TEMPORALE]
- Visualizzazione precipitazioni: mm e probabilità %
- Dati mostrati solo se pioggia prevista > 0mm
- Aggiornamento automatico per settimana visualizzata

3.7 Export PDF

Funzionalità di export per reportistica:

- Conversione tabella completa in PDF landscape A4
- Header con: Titolo, Date, Nome generatore, Timestamp
- Gestione multi-pagina automatica
- Librerie: jsPDF + html2canvas
- Nome file: pianificazione_[data-inizio].pdf

3.8 Responsive Design

Ottimizzazioni specifiche per dispositivi mobile (< 768px):

- Colonne giorni normali: 75px (uniforme)
- Colonne weekend: 50px (compatte)
- Font ridotti: 10-11px
- Touch scrolling ottimizzato
- Tooltip disabilitati (solo popup)
- Header wrappato per schermi piccoli

4. CONFIGURAZIONE BASEROW

4.1 Tabelle Database

ID	Nome Tabella	Descrizione	Campi Principali
609	managers	Responsabili team	id, name, password, team[], is_superuser
2692	planning	Pianificazione turni	person, date, period, activity, manager_id
2693	externals	Aziende esterne	name
2694	team_assignments	Assegnamenti team	manager_id, person
610	daily_notes	Note giornaliere	date, note, manager_id, manager_name

4.2 Variabili Ambiente (.env)

```
BASEROW_URL=https://api.baserow.io BASEROW_TOKEN=tu_token_api_qui  
PLANNING_TABLE_ID=2692 EXTERNALS_TABLE_ID=2693 TEAM_ASSIGNMENTS_TABLE_ID=2694  
MANAGERS_TABLE_ID=609 DAILY_NOTES_TABLE_ID=610
```

[ATTENZIONE] IMPORTANTE: Il token API Baserow deve avere permessi di lettura/scrittura su tutte le tabelle elencate. Si ottiene da: Baserow → Settings → API Tokens.

4.3 Schema Campi Dettagliato

Tabella: managers (ID 609)

Campo	Tipo	Descrizione
id	Text	Identificativo univoco manager (es: 'daniele')
name	Text	Nome completo (es: 'Daniele')
password	Text	Password in chiaro (per semplicità)
team	Multiple Select	Lista operatori del team
is_superuser	Boolean	Flag amministratore (true/false)

Tabella: planning (ID 2692)

Campo	Tipo	Descrizione
person	Text	Nome operatore
date	Date	Data turno (YYYY-MM-DD)
period	Text	Periodo: 'am' o 'pm'
activity	Long Text	Descrizione attività
manager_id	Text	ID manager che ha inserito

Tabella: daily_notes (ID 610)

Campo	Tipo	Descrizione
date	Date	Data nota (YYYY-MM-DD)
note	Long Text	Testo nota
manager_id	Text	ID manager autore
manager_name	Text	Nome manager (auto-popolato)

5. FILE DI CONFIGURAZIONE COMPLETI

[ATTENZIONE] Questa sezione contiene TUTTI i file necessari per ricreare il progetto da zero. Copiare e salvare questi file esattamente come mostrato.

5.1 File: backend/.env

```
# Configurazione Baserow
BASEROW_URL=https://api.baserow.io
BASEROW_TOKEN=IL_TUO_TOKEN QUI # ID Tabelle Baserow
PLANNING_TABLE_ID=2692
EXTERNALS_TABLE_ID=2693 TEAM_ASSIGNMENTS_TABLE_ID=2694 MANAGERS_TABLE_ID=609
DAILY_NOTES_TABLE_ID=610
```

IMPORTANTE: Sostituire 'IL_TUO_TOKEN QUI' con il token API reale ottenuto da:
Baserow - Settings - API Tokens - Create Token

5.2 File: backend/requirements.txt

```
fastapi==0.104.1 uvicorn==0.24.0 httpx==0.25.1 python-dotenv==1.0.0
pydantic==2.5.0
```

5.3 File: backend/Dockerfile

```
FROM python:3.11-slim WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY main.py .
COPY .env .
CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

5.4 File: docker-compose.yml

```
version: '3.8' services: planner-api: build: ./backend container_name: planner-api ports: - "8000:8000" restart: unless-stopped environment: - TZ=Europe/Rome planner-frontend: image: nginx:alpine container_name: planner-frontend ports: - "80:80" volumes: - ./frontend:/usr/share/nginx/html:ro restart: unless-stopped
```

5.5 Struttura Completa Directory

Creare la seguente struttura sul server:

/opt/pianificazione-app/

backend/
main.py (API FastAPI - codice completo nel repository)
.env (file configurazione - vedere 5.1)
Dockerfile (vedere 5.3)
requirements.txt (vedere 5.2)

frontend/
index.html (applicazione React - codice completo nel repository)

docker-compose.yml (vedere 5.4)

5.6 Passi per Deploy Iniziale

1. Creare directory: mkdir -p /opt/pianificazione-app/backend
/opt/pianificazione-app/frontend
2. Copiare main.py nella cartella backend/
3. Creare file .env in backend/ con il contenuto della sezione 5.1
4. Creare requirements.txt in backend/ con contenuto sezione 5.2
5. Creare Dockerfile in backend/ con contenuto sezione 5.3
6. Copiare index.html nella cartella frontend/
7. Creare docker-compose.yml nella root con contenuto sezione 5.4
8. Eseguire: cd /opt/pianificazione-app
9. Eseguire: docker compose up -d --build
10. Verificare log: docker logs planner-api
11. Testare: curl http://localhost:8000/api/managers

[!] NOTA: I file main.py (backend) e index.html (frontend) contengono centinaia di righe di codice. Per il codice completo, fare riferimento al repository del progetto o ai file sorgente esistenti sul server.

6. DEPLOYMENT E MANUTENZIONE

5.1 Comandi Docker Essenziali

```
# Riavvio completo (dopo modifiche) cd /opt/pianificazione-app docker compose down docker compose up -d --build # Visualizzare i log docker logs planner-api docker logs -f planner-api # modalità follow # Riavvio singolo container docker compose restart planner-api # Vedere container attivi docker ps # Accesso shell nel container docker exec -it planner-api bash
```

5.2 Struttura docker-compose.yml

Due servizi principali:

- **planner-api**: Backend FastAPI (porta 8000)
- **planner-frontend**: Nginx che serve index.html (porta 80)

Il reverse proxy esterno (Nginx/Traefik) gestisce SSL e routing.

5.3 Aggiornamento Codice

1. Modificare file sorgenti (main.py o index.html)
2. Eseguire: docker compose down
3. Eseguire: docker compose up -d --build
4. Verificare log per errori: docker logs planner-api
5. Testare funzionalità su browser (Ctrl+Shift+R per svuotare cache)

5.4 Backup e Restore

Dati: Baserow effettua backup automatici in cloud. Nessuna azione richiesta.

Codice: Mantenere copia locale di /opt/pianificazione-app/

File .env: Backup critico del file .env con token API

5.5 Monitoraggio Salute Sistema

```
# Verificare container in esecuzione docker ps | grep planner # Controllare uso risorse docker stats planner-api planner-frontend # Verificare spazio disco df -h # Test connessione API curl https://planner.danielesturla.it/api/managers
```

7. GUIDA UTENTE

7.1 Login

1. Aprire <https://planner.danielesturla.it> nel browser
2. Selezionare il proprio nome dal menu a tendina
3. Inserire la password
4. Click su 'Accedi'

7.2 Inserimento Attività

Per inserire un'attività per un operatore del proprio team:

1. Individuare la riga dell'operatore
2. Click sulla cella del giorno desiderato (Mattina o Pomeriggio)
3. Digitare l'attività (es: 'Cantiere via Roma')
4. Click fuori dalla cella o premere Tab → salvataggio automatico

[!] Se il testo è lungo (>50 caratteri), apparirà un indicatore '...'. Su desktop: posizionare il mouse per vedere tooltip. Su tutti i dispositivi: click sulla cella per aprire popup di modifica.

7.3 Cancellazione Attività

Per cancellare un'attività:

1. Aprire la cella
2. Cancellare tutto il testo
3. Click fuori o Tab
4. La cella vuota viene automaticamente eliminata dal database

7.4 Gestione Note Giornaliera

Le note giornaliere sono visibili a tutti i manager e possono contenere informazioni condivise (es: 'Riunione ore 10', 'Chiusura zona Nord').

- Click sull'icona [*] nella riga 'Note Giorno' per aprire popup
- Visualizzare note esistenti con badge autore
- Aggiungere nuova nota nella sezione '+ Aggiungi nuova nota'
- Modificare/Cancellare solo le proprie note (pulsanti disponibili)
- Le note di altri manager sono in sola lettura

7.5 Navigazione Settimane

Usare le frecce < e > nell'header per scorrere tra le settimane:

- < = Settimana precedente
- > = Settimana successiva
- L'indicatore mostra "Settimana del [data lunedì]"

7.6 Filtri Disponibili

Nell'header sono disponibili due checkbox:

- **Solo mio team:** Mostra solo gli operatori del proprio team
- **Mostra weekend:** Nasconde/mostra le colonne Sabato e Domenica

7.7 Export PDF

Per esportare la pianificazione corrente:

1. Click sul pulsante '[PDF] PDF' nell'header
2. Il browser scaricherà automaticamente un file PDF
3. Il PDF contiene l'intera pianificazione visibile sullo schermo
4. Nome file: pianificazione_[data-lunedì].pdf

7.8 Logout

Click su 'Esci' sotto il proprio nome nell'header in alto a destra.

8. GUIDA AMMINISTRATORE

8.1 Abilitazione Superuser

Per trasformare un manager in amministratore:

1. Accedere a Baserow
2. Aprire la tabella 'managers' (ID 609)
3. Localizzare la riga del manager
4. Impostare il campo 'is_superuser' su 'true'
5. Il manager vedrà badge 'ADMIN' al prossimo login

8.2 Privilegi Superuser

- Modifica di TUTTE le celle, anche di altri team
- Nessuna cella in modalità readonly
- Utile per correzioni urgenti o supervisione
- Badge 'ADMIN' rosso sempre visibile

8.3 Gestione Utenti

Aggiungere nuovo manager:

1. Baserow → Tabella 'managers'
2. Nuova riga con: id, name, password, team[], is_superuser
3. Salvare → disponibile immediatamente nel menu login

Modificare team:

1. Editare campo 'team' (Multiple Select)
2. Aggiungere/rimuovere nomi operatori
3. Salvare → il manager vedrà le modifiche al prossimo refresh

8.4 Gestione Operatori

Gli operatori non hanno accesso diretto al sistema, ma devono essere presenti nei campi 'team' dei manager per apparire nella pianificazione.

Aggiungere nuovo operatore:

1. Aggiungere il nome al campo 'team' di un manager
2. L'operatore appare immediatamente nella tabella

Trasferire operatore tra team:

1. Rimuovere dal team vecchio
2. Aggiungere al team nuovo

8.5 Pulizia Dati

Eliminare pianificazioni vecchie:

- Accedere a Baserow → Tabella 'planning' (ID 2692)
- Filtrare per date vecchie
- Selezionare righe e cancellare

Eliminare note vecchie:

- Tabella 'daily_notes' (ID 610)
- Stesso procedimento

9. TROUBLESHOOTING

9.1 Non riesco a fare login

- Verificare credenziali in tabella Baserow 'managers'
- Controllare che l'utente esista e password sia corretta
- Verificare log backend: docker logs planner-api
- Verificare connessione a Baserow (token API valido)

9.2 Le modifiche non si salvano

- Verificare di aver cliccato fuori dalla cella (trigger onBlur)
- Controllare log backend per errori API
- Verificare permessi: sei nel team corretto?
- Verificare che non sei superuser per quella cella
- Ricaricare pagina e riprovare

9.3 Celle bloccate (readonly)

- Verifica: l'operatore è nel tuo team?
- Se no, è normale (solo superuser può modificare tutti)
- Verifica campo 'team' del tuo manager in Baserow

9.4 Note non visibili

- Verificare tabella 'daily_notes' su Baserow
- Controllare che il campo 'date' sia corretto (YYYY-MM-DD)
- Ricaricare pagina con Ctrl+Shift+R
- Verificare console browser per errori JavaScript

9.5 Layout rotto su mobile

- Svuotare cache browser/app mobile
- Riavviare app/browser
- Verificare che file index.html sia aggiornato
- Controllare media query CSS nel codice

9.6 Export PDF fallisce

- Verificare che librerie jsPDF e html2canvas siano caricate
- Controllare console browser per errori JavaScript
- Provare a ridurre contenuto visibile (disabilitare weekend)
- Verificare permessi download browser

9.7 Comandi Debug Utili

```
# Log backend in tempo reale docker logs -f planner-api | grep ERROR #
Verificare connessione Baserow curl -H "Authorization: Token TUO_TOKEN" \
https://api.baserow.io/api/database/rows/table/609/?user_field_names=true #
Riavvio forzato docker compose down && docker compose up -d --build # Verificare
stato container docker ps -a | grep planner
```

10. CRONOLOGIA MODIFICHE

Data	Versione	Modifiche Principali
Gen 2026	0.1	<ul style="list-style-type: none">- Setup iniziale progetto- Tabella pianificazione base- Login manager
Gen 2026	0.2	<ul style="list-style-type: none">- Integrazione Baserow API- CRUD completo pianificazione- Gestione duplicati
Gen 2026	0.3	<ul style="list-style-type: none">- Separazione onChange/onBlur- Fix salvataggio celle- DELETE endpoint
Gen 2026	0.4	<ul style="list-style-type: none">- Export PDF funzionante- Deploy Docker/Portainer- SSL attivo
Gen 2026	0.5	<ul style="list-style-type: none">- Note giornaliera singole- Manager attribution- Popup note
Gen 2026	0.6	<ul style="list-style-type: none">- Note multiple per giorno- Gestione array note- Edit/Delete note proprie
Gen 2026	0.7	<ul style="list-style-type: none">- Superuser implementation- Badge ADMIN- Permessi cross-team
Gen 2026	0.8	<ul style="list-style-type: none">- UI compattazione (-15% altezza)- Weekend -50% larghezza- Rimozione placeholder
Gen 2026	0.9	<ul style="list-style-type: none">- Integrazione meteo Open-Meteo- Precipitazioni display- Icône weather dinamiche
Gen 2026	0.10	<ul style="list-style-type: none">- Festività italiane 2026- FERIE/ASSENTE auto-coloring- Emoji [*] per note
Gen 2026	0.11	<ul style="list-style-type: none">- Filtro 'Solo mio team'- Ottimizzazioni mobile- Touch scrolling
Gen 2026	0.12	<ul style="list-style-type: none">- Celle lunghe con tooltip- Popup edit testi lunghi- Indicatore ...

Feb 2026	1.0	- Filtro 'Mostra weekend' - Fix responsive mobile - Larghezze unificate 75/50px
----------	-----	---

10.1 Roadmap Futura (Opzionale)

Possibili sviluppi futuri da considerare:

- Notifiche push per modifiche importanti
- Sistema di approvazione turni
- Esportazione Excel/CSV oltre a PDF
- Dashboard statistiche (ore lavorate, presenze)
- Integrazione calendario esterno (Google Calendar, iCal)
- Sistema di template per attività ricorrenti
- Storico modifiche con audit trail
- API mobile dedicata per app nativa

--- FINE DOCUMENTAZIONE ---

Documento generato automaticamente

Per supporto tecnico: consultare sezione Troubleshooting