

# ANALISI GESTIONE SCADENZE

MyGest - Data: 19/11/2025

## 1. INTRODUZIONE

La gestione delle scadenze rappresenta un obiettivo primario del progetto MyGest. Questo documento analizza lo stato attuale dell'implementazione, confrontandola con i requisiti desiderati, e propone un piano di miglioramento dettagliato.

## 2. REQUISITI DESIDERATI

### 2.1 Modello Scadenza

La scadenza è un promemoria di un flusso di lavoro con i seguenti attributi:

- titolo - identificativo della scadenza
- descrizione - dettagli estesi
- stato - stato corrente della scadenza
- priorità - livello di urgenza
- categoria - classificazione tematica
- note\_interne - annotazioni riservate
- periodicità - unità di ripetizione (giornaliera, mensile, settimanale)
- periodicità\_intervallo - numero di unità tra occorrenze (es: ogni 3 mesi)
- num\_occorrenze - numero totale di occorrenze da generare
- data\_scadenza - data della prima o unica occorrenza
- google\_calendar\_calendar\_id - ID calendario Google per sincronizzazione
- Collegamenti: pratica, documento, fascicolo (opzionali)
- Assegnatari: uno o più utenti responsabili

### 2.2 Modello Occorrenza

Le occorrenze sono la proiezione temporale della scadenza. Ogni scadenza ha almeno 1 occorrenza. Gli attributi previsti sono:

- titolo - può differire dalla scadenza padre
- descrizione - dettagli specifici dell'occorrenza
- data - giorno dell'occorrenza
- ora\_inizio - orario di inizio
- ora\_fine - orario di fine

- giornaliera - flag per evento all-day
- stato - stato specifico dell'occorrenza
- google\_calendar\_event\_id - ID evento Google Calendar
- google\_calendar\_synced\_at - timestamp sincronizzazione

## 2.3 Sistema Alert

Ogni occorrenza deve supportare MULTIPLI alert con:

- metodo\_alert - email, webhook o altro
- offset\_alert\_periodo - unità di tempo (minuti, ore, giorni, settimane)
- offset\_alert - numero di unità prima dell'occorrenza
- alert\_programmata\_il - timestamp calcolato per l'invio
- Esempio: primo alert 1 settimana prima, secondo 1 giorno prima, terzo 1 ora prima

## 3. STATO ATTUALE IMPLEMENTAZIONE

### 3.1 Aspetti Soddisfatti ✓

L'implementazione attuale copre correttamente molti aspetti:

Componente	Attributi Implementati
Scadenza	titolo, descrizione, stato, priorità, categoria, note_interne
	periodicità, periodicità_intervallo, google_calendar_calendar_id
	Collegamenti M2M: pratiche, fascicoli, documenti
	Assegnatari M2M con User
Occorrenza	titolo, descrizione, stato, metodo_alert
	google_calendar_event_id, google_calendar_synced_at
	Collegamento a Comunicazione
Servizi	OccurrenceGenerator per generazione occorrenze
	AlertDispatcher per invio notifiche
	GoogleCalendarSync per sincronizzazione

### 3.2 Problemi e Discrepanze Critiche ✗

#### 1. Attributi mancanti in Scadenza:

- num\_occorrenze: non esiste come campo del modello
- data\_scadenza: manca il campo per la data della prima occorrenza

#### 2. Problemi in ScadenzaOccorrenza:

- Mancano campi separati: data, ora\_inizio, ora\_fine
- Esiste solo 'inizio' e 'fine' come DateTimeField completi
- giornaliera: manca il flag per eventi all-day
- offset\_alert\_periodo: non esiste come campo
- Si usa solo offset\_alert\_minuti invece di unità configurabili

#### 3. LIMITAZIONE CRITICA - Alert Singolo:

- Ogni occorrenza supporta UN SOLO alert
- Requisito: MULTIPLI alert per occorrenza
- Impossibile configurare: 1 settimana + 1 giorno + 1 ora prima

#### 4. Interfaccia utente:

- Vista lista non evidenzia l'accoppiata Scadenza/Occorrenza
- Manca una vista Scadenziario dedicata
- Alert multipli non gestibili tramite form

## 4. SOLUZIONI PROPOSTE

### 4.1 Soluzione Alert Multipli (PRIORITÀ ALTA)

**Problema:** Attualmente ogni occorrenza ha un solo alert.

**Soluzione:** Creare un nuovo modello **ScadenzaAlert** con relazione ForeignKey verso ScadenzaOccorrenza.

```
class ScadenzaAlert(models.Model):
    occorrenza = ForeignKey(ScadenzaOccorrenza, related_name='alerts')
    offset_alert_periodo = CharField(choices=[minuti, ore, giorni, settimane])
    offset_alert = PositiveIntegerField() # numero di unità
    metodo_alert = CharField(choices=[EMAIL, WEBHOOK])
    alert_config = JSONField()
    alert_programmata_il = DateTimeField() # calcolato automaticamente
    alert_inviata_il = DateTimeField(null=True)
    stato = CharField(choices=[pending, sent, failed])
```

Vantaggi:

- Supporto illimitato di alert per occorrenza
- Configurazione granulare (1 settimana, 1 giorno, 1 ora prima)
- Diversi metodi per diversi alert (email + webhook)
- Tracciabilità stato per ogni singolo alert

### 4.2 Campi Mancanti (PRIORITÀ ALTA)

**In Scadenza aggiungere:**

- num\_occorrenze: PositiveIntegerField(null=True) - numero massimo di occorrenze
- data\_scadenza: DateField(null=True) - data prima occorrenza

**In ScadenzaOccorrenza aggiungere:**

- giornaliera: BooleanField(default=False) - flag per evento all-day

**Rimuovere da ScadenzaOccorrenza:**

- offset\_alert\_minuti, alert\_config → spostati in ScadenzaAlert

### 4.3 Vista Scadenziario (PRIORITÀ MEDIA)

Creare una vista dedicata che mostri tutte le occorrenze future raggruppate per giorno, evidenziando l'accoppiata Scadenza/Occorrenza con:

- Filtro per range temporale (oggi, questa settimana, questo mese)
- Visualizzazione per giorno con tutte le occorrenze
- Indicazione degli alert programmati per ogni occorrenza
- Link rapidi a scadenza, pratica, fascicolo collegati
- Azioni: completa, rinvia, invia alert manualmente

### 4.4 Miglioramento Template Lista (PRIORITÀ MEDIA)

Modificare la home delle scadenze per evidenziare meglio:

- Colonna 'Prossima Occorrenza' con data/ora ben visibile
- Colonna 'Alert Programmati' con conteggio
- Badge colorati per priorità e stato
- Filtri rapidi per stato, priorità, assegnatario
- Indicatore visivo per scadenze senza occorrenze future

## 5. PIANO DI IMPLEMENTAZIONE

Fase	Attività	Priorità	Tempo
1	Creare modello ScadenzaAlert	ALTA	2h
1	Creare migration per nuovi campi	ALTA	1h
1	Migration dati esistenti (offset_alert_minuti → ScadenzaAlert)	ALTA	2h
1	Aggiornare forms per gestire alert multipli	ALTA	3h
1	Aggiornare admin per inline alerts	ALTA	1h
2	Modificare AlertDispatcher per gestire alert multipli	ALTA	2h
2	Aggiornare services OccurrenceGenerator	ALTA	1h
2	Test unitari nuova logica alert	ALTA	2h
3	Creare vista Scadenziario	MEDIA	4h
3	Creare template scadenziario.html	MEDIA	3h
3	Aggiornare template home.html	MEDIA	2h
3	Aggiornare template detail.html per alert multipli	MEDIA	2h
4	Aggiornare API serializers per ScadenzaAlert	MEDIA	2h
4	Aggiornare endpoints API	MEDIA	2h
4	Documentazione modifiche	MEDIA	1h
5	Test integrazione completi	ALTA	3h
5	Test con utenti reali	MEDIA	2h

**Stima Totale: ~35 ore di sviluppo**

### 5.1 Riepilogo per Priorità

Priorità	Attività	Motivazione
ALTA	Modello ScadenzaAlert + Migration	Risolve limitazione critica degli alert singoli
ALTA	Nuovi campi (num_occorrenze, data_scadenza, modello)	Adempimento requisiti desiderati
ALTA	Aggiornamento logica alert	Necessario per supportare alert multipli
MEDIA	Vista Scadenziario	Migliora UX e visibilità occorrenze
MEDIA	Miglioramento template	Evidenzia accoppiata Scadenza/Occorrenza
MEDIA	Aggiornamento API	Supporto frontend per nuove funzionalità
BASSA	Integrazione calendario visuale	Enhancement UI opzionale
BASSA	Export PDF/Excel	Funzionalità extra per reporting

## 6. CONCLUSIONI

L'implementazione attuale della gestione scadenze in MyGest copre una buona parte dei requisiti desiderati, ma presenta alcune **limitazioni critiche** che ne riducono l'efficacia:

### Punti di forza:

- Architettura solida con separazione Scadenza/Occorrenza
- Integrazione con Google Calendar funzionante
- Collegamenti M2M con pratiche, fascicoli, documenti
- Sistema di logging e audit trail completo
- Supporto periodicità multiple con generazione automatica

### Criticità da risolvere:

- **Alert singolo per occorrenza:** limita drasticamente l'utilità del sistema
- Campi mancanti che limitano la configurazione
- UI non ottimizzata per l'uso quotidiano (manca scadenziario)

### Raccomandazione:

Si consiglia di procedere con l'implementazione delle modifiche **PRIORITÀ ALTA** nel più breve tempo possibile, in particolare la creazione del modello ScadenzaAlert che risolve la limitazione più critica del sistema attuale.

Le modifiche di priorità MEDIA possono essere implementate in una seconda fase, ma sono comunque importanti per rendere il sistema realmente utilizzabile nel quotidiano.

Con queste implementazioni, MyGest disporrà di un sistema di gestione scadenze completo, flessibile e allineato ai requisiti originali del progetto.