

Piano di Sviluppo in 48 Ore

Progetto Cloud Monitoring CoffeeMek

Data: 06/06/2025

Autore: Manus AI

Versione: 1.0

Indice

- [1. Introduzione](#)
 - [2. Analisi dei Requisiti Essenziali](#)
 - [3. Architettura Minima](#)
 - [4. Pianificazione Frontend \(Blazor\)](#)
 - [5. Pianificazione Backend \(Node.js\)](#)
 - [6. Piano Orario Dettagliato](#)
 - [7. Gestione Rischi e Dipendenze](#)
 - [8. Deliverable Finali](#)
 - [9. Conclusioni](#)
-

Introduzione

Questo documento presenta un piano dettagliato per lo sviluppo del sistema di Cloud Monitoring per CoffeeMek nell'arco di 48 ore, con un team composto da 3 persone: - 1 sviluppatore frontend (Blazor) - 2 sviluppatori backend (Node.js)

Il piano è stato strutturato per massimizzare l'efficienza del team, rispettare le dipendenze tra componenti e garantire la consegna di un MVP (Minimum Viable Product) funzionante entro il termine stabilito.

Analisi dei Requisiti Essenziali

Vincoli di Progetto

- **Tempo disponibile:** 48 ore
- **Team:** 3 persone
- 1 sviluppatore frontend (Blazor)
- 2 sviluppatori backend (Node.js)

Funzionalità Essenziali (MVP)

Data la limitazione di tempo (48 ore), sono state identificate le seguenti funzionalità essenziali:

1. Frontend Essenziale (Blazor)

- **Dashboard semplificata:**
 - Visualizzazione base dello stato dei macchinari
 - Filtro per sede produttiva
 - Visualizzazione dei dati di telemetria più recenti
- **Gestione lotti semplificata:**
 - Visualizzazione dei lotti attivi
 - Form base per creazione nuovo lotto
- **Autenticazione base:**
 - Login semplice senza gestione ruoli complessa

2. Backend Essenziale (Node.js)

- **API Core:**
 - Endpoint per autenticazione
 - Endpoint per recupero dati macchinari
 - Endpoint per gestione base dei lotti
- **Database:**
 - Schema semplificato con tabelle essenziali
 - Connessione a Azure SQL

- **Gestione dati in tempo reale:**
- Ricezione dati dai macchinari (o dal simulatore)
- Elaborazione base e salvataggio

3. Simulatore Semplificato

- **Generatore dati base:**
- Simulazione di un numero limitato di macchinari
- Generazione dati con frequenza fissa
- Invio dati al backend

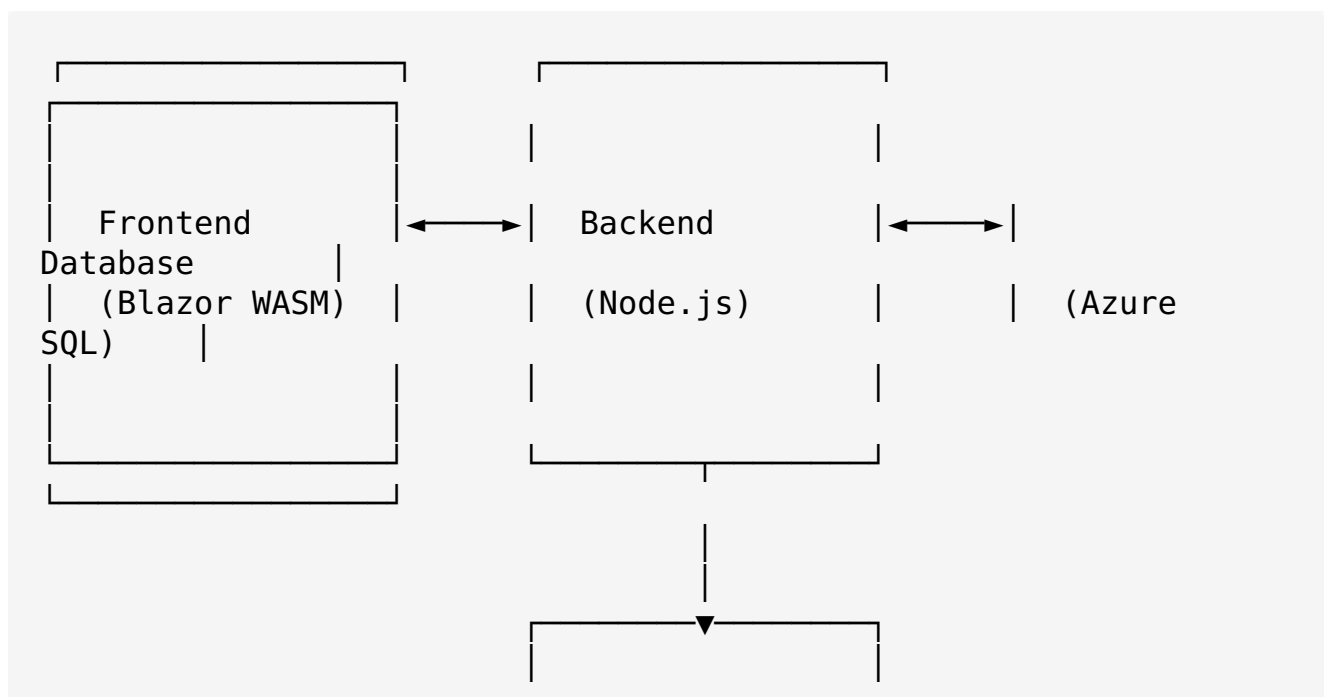
Funzionalità Escluse dall'MVP

Per rispettare il vincolo di 48 ore, le seguenti funzionalità sono state escluse:

1. **CRUD completo** per clienti e commesse
2. **Schedulazione avanzata** dei lotti
3. **Dashboard avanzata** con grafici complessi
4. **Gestione completa** dei fusi orari
5. **Reportistica storica** dettagliata
6. **Simulazione avanzata** con scenari complessi
7. **Gestione ruoli e permessi** avanzata

Architettura Minima

Architettura Complessiva



Componenti Principali

1. Frontend (Blazor WebAssembly)

- 2. Layout e navigazione base
- 3. Autenticazione
- 4. Dashboard

5. Gestione lotti

6. Backend (Node.js)

- 7. Server Express
- 8. Autenticazione JWT
- 9. API RESTful
- 10. Comunicazione real-time

11. Connessione database

12. Database (Azure SQL)

13. Schema minimo con tabelle per:

- Utenti
- Sedi produttive
- Macchinari
- Lotti
- Telemetria

14. Simulatore (Node.js)

- 15. Generatore dati
- 16. Comunicazione con backend

Schema Database Minimo

```
-- Utenti
CREATE TABLE Users (
  Id INT IDENTITY(1,1) PRIMARY KEY,
  Username NVARCHAR(50) NOT NULL UNIQUE,
  PasswordHash NVARCHAR(100) NOT NULL,
```

```

    Role NVARCHAR(20) DEFAULT 'User'
);

-- Sedi Produttive
CREATE TABLE Facilities (
    Id INT IDENTITY(1,1) PRIMARY KEY,
    Name NVARCHAR(100) NOT NULL,
    Location NVARCHAR(100) NOT NULL,
    TimeZone NVARCHAR(50) NOT NULL
);

-- Macchinari
CREATE TABLE Machines (
    Id INT IDENTITY(1,1) PRIMARY KEY,
    Name NVARCHAR(100) NOT NULL,
    Type NVARCHAR(50) NOT NULL,
    FacilityId INT FOREIGN KEY REFERENCES Facilities(Id),
    Status NVARCHAR(20) DEFAULT 'Idle'
);

-- Lotti
CREATE TABLE Lots (
    Id INT IDENTITY(1,1) PRIMARY KEY,
    Code NVARCHAR(50) NOT NULL UNIQUE,
    Description NVARCHAR(200),
    CreatedAt DATETIME DEFAULT GETUTCDATE(),
    Status NVARCHAR(20) DEFAULT 'Created',
    CurrentMachineId INT NULL FOREIGN KEY REFERENCES
Machines(Id)
);

-- Telemetria
CREATE TABLE Telemetry (
    Id INT IDENTITY(1,1) PRIMARY KEY,
    MachineId INT FOREIGN KEY REFERENCES Machines(Id),
    LotId INT NULL FOREIGN KEY REFERENCES Lots(Id),
    Timestamp DATETIME DEFAULT GETUTCDATE(),
    DataType NVARCHAR(50) NOT NULL,
    DataValue NVARCHAR(MAX) NOT NULL
);

```

Pianificazione Frontend (Blazor)

Attività e Tempi

| Fase | Descrizione | Ore |
|--------|--------------------------------|-----|
| 1 | Setup e Configurazione | 4 |
| 2 | Implementazione Autenticazione | 4 |
| 3 | Layout e Navigazione | 4 |
| 4 | Dashboard | 8 |
| 5 | Gestione Lotti | 8 |
| 6 | Comunicazione Real-time | 4 |
| 7 | Testing e Ottimizzazione | 4 |
| 8 | Deployment | 4 |
| Totale | | 40 |

Tecnologie e Librerie

- **Blazor WebAssembly**: Framework principale
 - **Dapper**: Per accesso ai dati
 - **Radzen Blazor Components**: Per componenti UI avanzati
 - **BlazorStrap**: Per componenti Bootstrap in Blazor
 - **SignalR Client**: Per comunicazione real-time
-

Pianificazione Backend (Node.js)

Suddivisione Sviluppatori

- **Sviluppatore Backend 1 (BE1)**: Responsabile dell'API core, autenticazione e database
- **Sviluppatore Backend 2 (BE2)**: Responsabile della comunicazione real-time, simulatore e deployment

Attività e Tempi per Sviluppatore Backend 1 (BE1)

| Fase | Descrizione | Ore |
|------------|------------------------------|-----|
| 1.1 | Creazione Progetto Node.js | 2 |
| 1.2 | Configurazione Database | 2 |
| 2.1 | Implementazione JWT | 2 |
| 2.2 | Middleware di Autorizzazione | 2 |
| 3.1 | API Utenti | 2 |
| 3.2 | API Macchinari | 2 |
| 3.3 | API Lotti | 3 |
| 6.1 | Test Unitari | 2 |
| 6.2 | Test di Integrazione | 2 |
| 6.3 | Ottimizzazione Performance | 2 |
| 6.4 | Documentazione API | 2 |
| Totale BE1 | | 23 |

Attività e Tempi per Sviluppatore Backend 2 (BE2)

| Fase | Descrizione | Ore |
|------|--------------------------------|-----|
| 1.3 | Setup Ambiente di Sviluppo | 2 |
| 2.3 | Sicurezza API | 2 |
| 3.4 | API Telemetria | 3 |
| 4.1 | Setup Server WebSocket | 2 |
| 4.2 | Broadcast Dati Telemetria | 3 |
| 4.3 | Integrazione con API | 3 |
| 5.1 | Setup Progetto Simulatore | 2 |
| 5.2 | Generatore Dati Fresa e Tornio | 3 |

| Fase | Descrizione | Ore |
|-------------------|---------------------------|-----------|
| 5.3 | Generatore Dati Linee | 3 |
| 5.4 | Comunicazione con Backend | 4 |
| 7.1 | Configurazione Azure | 2 |
| 7.2 | Deployment Backend | 2 |
| 7.3 | Deployment Simulatore | 2 |
| 7.4 | Test Post-Deployment | 2 |
| Totale BE2 | | 35 |

Tecnologie e Librerie

- **Express.js**: Framework web
 - **tedious**: Driver per SQL Server
 - **jsonwebtoken**: Per gestione JWT
 - **socket.io**: Per comunicazione real-time
 - **dotenv**: Per gestione variabili d'ambiente
 - **cors**: Per gestione CORS
-

Piano Orario Dettagliato

Giorno 1 (Prime 24 ore)

Ore 1-4: Setup e Configurazione Iniziale

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|---------------------------------------|----------------------------|-------------------------------------|
| 1 | Creazione progetto Blazor WebAssembly | Creazione progetto Node.js | Setup ambiente di sviluppo |
| 2 | Integrazione librerie UI | Configurazione Express.js | Configurazione variabili d'ambiente |
| 3 | Configurazione autenticazione | Configurazione database | Setup sistema di build |

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|----------------------------------|---------------------------|------------------------------------|
| 4 | Configurazione comunicazione API | Creazione schema database | Implementazione sistema di logging |

Ore 5-8: Autenticazione e Struttura Base

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|------------------------------|--------------------------------|---------------------------|
| 5 | Implementazione pagina login | Implementazione JWT | Sicurezza API |
| 6 | Implementazione form login | Generazione token | Rate limiting e HTTPS |
| 7 | Gestione sessione | Middleware autorizzazione | API Telemetria (inizio) |
| 8 | Protezione rotte | Gestione errori autorizzazione | API Telemetria (continua) |

Ore 9-12: Componenti Core

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|-------------------------|---------------------------|---------------------------|
| 9 | Layout principale | API Utenti | API Telemetria (completa) |
| 10 | Barra di navigazione | API Macchinari (inizio) | Setup Server WebSocket |
| 11 | Menu di navigazione | API Macchinari (completa) | Configurazione namespace |
| 12 | Componenti condivisi | API Lotti (inizio) | Gestione connessioni |

Ore 13-16: Implementazione Dashboard e API

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|-------------------------|---------------------------|------------------------------------|
| 13 | Layout dashboard | API Lotti (continua) | Broadcast dati telemetria (inizio) |

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|-------------------------|---------------------------|--------------------------------------|
| 14 | Sistema di grid | API Lotti (completa) | Broadcast dati telemetria (continua) |
| 15 | Filtri e controlli | Test unitari API | Filtri per tipologia dati |
| 16 | Widget stato macchinari | Test integrazione API | Ottimizzazione performance |

Ore 17-20: Gestione Lotti e Simulatore

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|-------------------------|-------------------------------|------------------------------|
| 17 | Widget dati telemetria | Ottimizzazione query | Integrazione API e WebSocket |
| 18 | Lista lotti | Implementazione caching | Setup progetto simulatore |
| 19 | Tabella con paginazione | Documentazione API (inizio) | Generatore dati Fresa CNC |
| 20 | Dettaglio lotto | Documentazione API (completa) | Generatore dati Tornio |

Ore 21-24: Integrazione Componenti

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|-------------------------|---------------------------------|---|
| 21 | Creazione lotto | Supporto integrazione frontend | Generatore Linea Assemblaggio |
| 22 | Validazione dati | Supporto integrazione WebSocket | Generatore Linea Test |
| 23 | Aggiornamento stato | Test integrazione completa | Comunicazione simulatore-backend (inizio) |
| 24 | Componente avanzamento | Correzione bug identificati | Comunicazione simulatore-backend (continua) |

Giorno 2 (Ultime 24 ore)

Ore 25-28: Comunicazione Real-time e Simulatore

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|----------------------------|----------------------------------|---|
| 25 | Configurazione SignalR | Supporto integrazione simulatore | Comunicazione simulatore-backend (completa) |
| 26 | Connessione al server | Test integrazione simulatore | Gestione errori e ritentativi |
| 27 | Handler per messaggi | Correzione bug | Simulazione scenari errore |
| 28 | Aggiornamento UI real-time | Miglioramento performance | Test integrazione simulatore |

Ore 29-32: Testing e Ottimizzazione

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|----------------------------|---------------------------|------------------------------------|
| 29 | Testing manuale | Supporto frontend | Configurazione Azure |
| 30 | Verifica responsività | Supporto frontend | Setup risorse cloud |
| 31 | Test casi d'errore | Supporto frontend | Configurazione database produzione |
| 32 | Ottimizzazione performance | Supporto frontend | Variabili d'ambiente produzione |

Ore 33-36: Preparazione Deployment

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|---------------------------------|-----------------------------|-------------------------------|
| 33 | Lazy loading | Preparazione per deployment | Deployment backend (inizio) |
| 34 | Miglioramento tempi caricamento | Supporto deployment | Deployment backend (completa) |

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|-------------------------|---------------------------|---------------------------|
| 35 | Correzione bug | Supporto deployment | Deployment simulatore |
| 36 | Pulizia codice | Supporto deployment | Avvio simulazione |

Ore 37-40: Deployment e Test Finale

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|--------------------------|------------------------------|---------------------------------|
| 37 | Preparazione build | Test post-deployment backend | Test post-deployment simulatore |
| 38 | Ottimizzazione build | Correzione problemi | Monitoraggio simulatore |
| 39 | Generazione file statici | Supporto deployment frontend | Supporto deployment frontend |
| 40 | Deployment su Azure | Supporto deployment frontend | Supporto deployment frontend |

Ore 41-44: Integrazione e Test Sistema Completo

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|-------------------------------------|----------------------------|----------------------------|
| 41 | Test applicazione in produzione | Test integrazione completa | Test integrazione completa |
| 42 | Verifica connessione backend | Correzione problemi | Correzione problemi |
| 43 | Controllo funzionalità real-time | Ottimizzazione finale | Ottimizzazione finale |
| 44 | Correzione problemi post-deployment | Documentazione | Documentazione |

Ore 45-48: Finalizzazione e Buffer

| Ora | Frontend Developer (FE) | Backend Developer 1 (BE1) | Backend Developer 2 (BE2) |
|-----|-------------------------|---------------------------|---------------------------|
| 45 | Buffer per imprevisti | Buffer per imprevisti | Buffer per imprevisti |
| 46 | Buffer per imprevisti | Buffer per imprevisti | Buffer per imprevisti |
| 47 | Documentazione finale | Documentazione finale | Documentazione finale |
| 48 | Consegna finale | Consegna finale | Consegna finale |

Gestione Rischi e Dipendenze

Punti di Sincronizzazione Critici

Per garantire una corretta integrazione tra i componenti, sono stati identificati i seguenti punti di sincronizzazione critici:

- 1. **Fine Ora 8:** Allineamento su autenticazione (FE, BE1)
- 2. **Fine Ora 16:** Integrazione dashboard e API (FE, BE1, BE2)
- 3. **Fine Ora 24:** Verifica integrazione componenti (FE, BE1, BE2)
- 4. **Fine Ora 32:** Preparazione deployment (FE, BE1, BE2)
- 5. **Fine Ora 40:** Verifica deployment (FE, BE1, BE2)
- 6. **Fine Ora 44:** Test sistema completo (FE, BE1, BE2)

Gestione delle Dipendenze

Il piano è stato strutturato per gestire le seguenti dipendenze critiche:

- 1. **Frontend ↔ Backend API:**
- 2. L'implementazione dell'autenticazione frontend dipende dall'API di autenticazione
- 3. La dashboard frontend dipende dalle API per macchinari e telemetria
- 4. La gestione lotti frontend dipende dalle API per lotti
- 5. **Frontend ↔ Comunicazione Real-time:**
- 6. L'aggiornamento in tempo reale della UI dipende dal server WebSocket
- 7. **Backend API ↔ Simulatore:**
- 8. Le API di telemetria dipendono dai dati generati dal simulatore

9. Deployment:

10. Il deployment frontend dipende dal deployment backend
11. Il test finale dipende dal deployment di tutti i componenti

Strategie di Mitigazione dei Rischi

1. Problemi di integrazione:

2. Punti di sincronizzazione frequenti
3. Sviluppatori backend disponibili a supportare il frontend quando necessario

4. Ritardi nello sviluppo:

5. Prioritizzazione delle funzionalità essenziali
6. Possibilità di semplificare ulteriormente alcune componenti

7. Problemi di deployment:

8. Test di deployment anticipato (ora 32)
9. Supporto congiunto durante la fase di deployment

10. Problemi tecnici imprevisti:

11. Buffer dedicato nelle ultime ore
 12. Possibilità di riallocare risorse in base alle necessità
-

Deliverable Finali

Al termine delle 48 ore, il team consegnerà:

1. Frontend Blazor

- Applicazione deployata su Azure
- Funzionalità di autenticazione
- Dashboard con visualizzazione stato macchinari
- Gestione base dei lotti
- Aggiornamenti in tempo reale

2. Backend Node.js

- API RESTful deployate su Azure

- Autenticazione JWT
- Endpoint per macchinari, lotti e telemetria
- Server WebSocket per comunicazione real-time

3. Simulatore

- Applicazione Node.js deployata
- Generazione dati per i quattro tipi di macchinari
- Invio dati al backend

4. Documentazione

- Documentazione API
 - Istruzioni di utilizzo base
 - Note di deployment
-

Conclusioni

Questo piano dettagliato per 48 ore è stato progettato per massimizzare l'efficienza del team di 3 persone e garantire la consegna di un MVP funzionante del sistema di Cloud Monitoring per CoffeeMek. La suddivisione dei compiti, i punti di sincronizzazione e il buffer per imprevisti mirano a ridurre i rischi e aumentare le probabilità di successo del progetto.

Il piano si concentra sulle funzionalità essenziali che forniscono il massimo valore nel breve termine, rimandando le funzionalità avanzate a sviluppi futuri. L'architettura proposta, sebbene semplificata, è progettata per essere scalabile e permettere l'aggiunta di nuove funzionalità in futuro.

Documento preparato da: Manus AI

Data: 06/06/2025