

RELAZIONE TECNICA

FliBrutti Backend API

Panoramica del Progetto e Tecnologie Utilizzate

1. Introduzione e Panoramica del Progetto

Il progetto FliBrutti Backend rappresenta un sistema gestionale completo sviluppato con tecnologie Microsoft .NET 8. L'applicazione è progettata per gestire le operazioni aziendali di un'impresa nel settore dei trasporti NCC (Noleggio Con Conducente), offrendo funzionalità di gestione utenti, autenticazione sicura, tracciamento delle presenze dei dipendenti e gestione dei preventivi.

L'architettura del sistema segue i principi della Clean Architecture, garantendo separazione delle responsabilità, testabilità del codice e facilità di manutenzione nel tempo. Il backend espone API RESTful protette da autenticazione JWT, consentendo l'integrazione con qualsiasi client frontend.

2. Architettura del Sistema

Il progetto adotta una struttura a layer ben definita, organizzata in cinque progetti distinti che seguono il pattern della Clean Architecture:

Layer	Descrizione e Responsabilità
API	Presentation layer contenente i Controller ASP.NET Core, la configurazione dei servizi e il punto di ingresso dell'applicazione.
Application	Business logic layer con servizi, interfacce, DTO, mappers e logica di validazione.
Core	Domain layer contenente le entità del dominio, enumerazioni e modelli di base.
Infrastructure	Data access layer con DbContext di Entity Framework e configurazioni del database.
Test	Progetto di testing con unit test basati su xUnit per la verifica delle funzionalità.

Questa organizzazione permette una chiara separazione delle dipendenze.

3. Stack Tecnologico

3.1 Framework e Runtime

- **.NET 8.0** - Framework di sviluppo principale, ultima versione LTS di Microsoft
- **ASP.NET Core 8** - Framework per la creazione di Web API RESTful
- **Entity Framework Core 8.0.22** - ORM per l'accesso ai dati con approccio Code-First
- **Pomelo.EntityFrameworkCore.MySql 8.0.3** - Provider MySQL per EF Core

3.2 Database

Il sistema utilizza MySQL 8.0 come database relazionale.

3.3 Sicurezza e Autenticazione

- **JWT Bearer Authentication** - Autenticazione stateless tramite token JSON Web Token
- **Refresh Token Rotation** - Sistema di rinnovo token con invalidazione automatica
- **Argon2id** - Algoritmo di hashing password
- **Role-Based Authorization** - Controllo accessi basato su ruoli (Admin, Dipendente)

3.4 Logging e Monitoraggio

Serilog è utilizzato come framework di logging strutturato. La configurazione prevede output su console e file organizzati per data e contesto sorgente, facilitando il debug e l'analisi dei problemi in produzione.

4. Funzionalità Principali

4.1 Gestione Utenti

Il sistema supporta due tipologie di utenti: utenti autenticati (dipendenti e amministratori) e utenti non autenticati (clienti che richiedono preventivi). Gli amministratori possono creare nuovi account, modificare i ruoli e gestire le informazioni personali.

4.2 Sistema di Autenticazione

L'autenticazione implementa un flusso completo con Access Token (validità 2 ore) e Refresh Token (validità 15 giorni). Il sistema include protezione contro l'accumulo di token tramite eliminazione fisica dei token precedenti al login.

4.3 Gestione Presenze (Firme)

I dipendenti possono registrare entrata e uscita dal lavoro. Il sistema verifica automaticamente che non esistano firme aperte precedenti e traccia tutti i movimenti con timestamp precisi.

4.4 Preventivi NCC

La gestione dei preventivi permette ai clienti di richiedere quotazioni per servizi di trasporto NCC. Gli amministratori possono esaminare le richieste, calcolare i costi (inclusi eventuali extra) e aggiornare lo stato da "da elaborare" a "completato".

5. Modello dei Dati

Il database è strutturato secondo il seguente schema entità-relazioni:

Entità	Descrizione
Person	Dati anagrafici base (nome, cognome, data di nascita)
User : Person	Utenti autenticati con email, password hashata e ruolo
UserNotAuthenticated : Person	Clienti che richiedono preventivi senza registrazione
Firma	Registrazioni di entrata/uscita dei dipendenti
PreventivoNCC	Richieste di preventivo con partenza, arrivo, costo
PreventivoExtra	Voci extra aggiuntive ai preventivi
RefreshToken	Token di refresh per la gestione delle sessioni

6. Endpoints API

Le API sono organizzate per controller e seguono le convenzioni REST. Il prefisso base è /v1/api/:

- **LoginController** - POST /login, POST /refresh, POST /revoke, POST /logout
- **UserController** - POST (crea utente), PATCH /UpdatePassword, PATCH /UpdateType
- **PersonController** - GET (ottieni persona), PATCH (aggiorna persona)
- **FirmaController** - GET (lista firme), POST /Entry, POST /Exit
- **PreventivoNCCController** - GET, POST, GET /ToExamine, PATCH /AddPreventivoCalculated