

# RELAZIONE PROGETTO APL - Power Fitness

## Panoramica del Progetto

Il progetto si propone di sviluppare un gestionale per una palestra, offrendo un'interfaccia di accesso tramite form di login. Gli utenti registrati possono accedere alle funzionalità dell'applicazione, mentre i nuovi utenti possono registrarsi tramite un'apposita schermata.

## Ruoli Utente

Il sistema gestisce due categorie di utenti:

- **Amministratori:** Vengono aggiunti manualmente nel database e non possono essere aggiunti tramite la schermata di registrazione. Hanno l'autorità di approvare, tramite un'area dedicata visibile solo a loro, le registrazioni dei dipendenti avvenute attraverso la schermata di registrazione.
- **Dipendenti:** Possono essere classificati come Dipendenti semplici e Responsabili, entrambi con funzionalità equivalenti per semplificare il sistema. Dopo essersi registrati tramite la schermata dedicata, l'accesso all'applicazione è consentito solo previa conferma da parte dell'amministratore.

## Interfaccia e Funzionalità

Dopo l'autenticazione, l'interfaccia presenta le varie funzionalità dell'applicazione, mentre il server Crow genera un token JWT, essenziale per garantire richieste autenticate al server.

Tra le funzionalità disponibili:

- **Gestione Corsi:** Schermata per visualizzare, creare, modificare ed eliminare i corsi, insieme ad un sistema intuitivo all'interno del quale una volta scelto un corso si può accedere ad una schermata per la **Gestione delle Lezioni** relative a quel corso.
- **Gestione Utenti:** Schermata che offre strumenti per la visualizzazione, la creazione, la modifica e l'eliminazione degli utenti. Quest'area include, inoltre, due funzionalità specifiche: **Gestione degli Abbonamenti** e **Gestione delle Prenotazioni** relative all'utente selezionato.
- **Statistica Abbonamenti:** Accesso a statistiche riguardanti il numero di abbonamenti emessi, selezionando una data specifica, rappresentate tramite un istogramma generato dal server Flask Python.
- **Info e Contatti:** Schermata informativa che fornisce dettagli sull'applicazione e i contatti degli sviluppatori.
- **Logout:** Opzione per il logout degli utenti.

## Funzionalità Specifiche

Oltre a quanto sopra, gli amministratori hanno il potere di approvare i dipendenti appena registrati attraverso la funzionalità:

- **Gestione Dipendenti:** Schermata che permette di gestire le approvazioni dei dipendenti.

## Funzionalità nel dettaglio

- Nell'area **Gestione Corsi**, i dipendenti possono gestire le lezioni relative a ciascun corso accedendo all'apposita voce **Gestione Lezioni**.
  - Nella sezione relativa alla **Gestione delle Lezioni**, ogni lezione è etichettata da un giorno specifico della settimana, ad esempio "Mercoledì", e da una fascia oraria specifica.  
È possibile pianificare più lezioni per un corso nello stesso giorno della settimana, assicurandosi che gli orari non si sovrappongano.
- All'interno dell'area **Gestione Utenti**, sono disponibili due schermate distinte accessibili selezionando un utente: una per la **Gestione degli Abbonamenti** e l'altra per la **Gestione delle Prenotazioni**.
  - Nella sezione relativa alla **Gestione degli Abbonamenti**, è possibile visualizzare gli abbonamenti di un utente, distinguendoli tra attivi, scaduti ed eventuali rinnovi già effettuati.  
I dipendenti possono creare nuovi abbonamenti per un utente o procedere con il rinnovo.  
Possono scegliere tra abbonamenti della durata di 1, 2, 3, 6 o 12 mesi, a partire dalla data odierna o successiva.
  - All'interno della sezione **Gestione delle Prenotazioni**, è consentito prenotare un utente per le lezioni dei corsi solo se dispone di un abbonamento attivo per la data della lezione e non ha già altre prenotazioni in conflitto di orario.

Nota: le lezioni di ciascun corso sono etichettate dal giorno della settimana (ad esempio "Mercoledì"), mentre le prenotazioni di ogni lezione sono etichettate dalla data (ad esempio "04/10/2023").

## Moduli Utilizzati

- **Modulo C#:** Client con interfaccia grafica che interagisce con gli altri moduli per eseguire azioni specifiche.
- **Modulo C++:** Server Crow che si connette al database e mette a disposizione del client le route per effettuare le richieste al server.
- **Modulo Python:** Server Flask che calcola le statistiche sugli abbonamenti, accedendo al database e fornendo i dati al client.

## Suddivisione del Lavoro

- **Modulo C#:** Sviluppato da Damiano Leone.
- **Modulo C++:** Sviluppato da Luca Cirrone. Damiano Leone ha collaborato per allineare e rettificare le funzionalità del server, garantendo una migliore integrazione con l'interfaccia grafica. Inoltre, ha sviluppato nuove route e apportato modifiche ad alcuni dei servizi riguardanti gli abbonamenti e le prenotazioni.
- **Modulo Python:** Sviluppato da Luca Cirrone.

## Link al repository Git:

[Damiano-Leone/Power-Fitness \(github.com\)](https://github.com/Damiano-Leone/Power-Fitness)