

Progetto PAC

Lorenzo Leoni, Davide Salvetti, Matteo Verzeroli

20 dicembre 2021

1 Iterazione 0

1.1 Requisiti

Si vuole realizzare una applicazione che permetta la gestione delle squadre della Protezione Civile. Gli attori in gioco sono i seguenti:

- **volontari**: coloro che operano durante gli interventi;
- **caposquadra**: responsabile di un gruppo di volontari durante gli interventi;
- **coordinatore**: amministratore del sistema, si occupa dell'inserimento di nuovi volontari e della creazione e modifica delle squadre.

I volontari sono organizzati in squadre ed ogni squadra ha un caposquadra che coordina le attività. Ogni squadra si occupa degli interventi all'interno di una specifica zona.

Per ogni attore devono essere memorizzate le informazioni anagrafiche di base (nome, cognome, CF, residenza). In particolare, per i volontari dovrà essere memorizzato anche il loro stato operativo: un volontario può dichiararsi operativo solo quando sta partecipando ad un intervento. Quando un volontario è operativo, il suo caposquadra potrà tracciare in tempo reale la posizione del volontario.

Gli interventi possono essere di due tipi: programmati oppure emergenze. È compito del caposquadra creare un nuovo intervento. Per ogni intervento bisogna memorizzare alcune informazioni (tipo di intervento, descrizione dell'intervento, materiali utilizzati durante l'intervento, partecipanti, ora, luogo) e poi, al termine, generare un report (in formato PDF) con un riassunto dei dati ed eventuali foto scattate. Il caposquadra potrà poi visualizzare la lista degli interventi e i rispettivi report. Il caposquadra potrà anche visualizzare le giacenze del magazzino ed eventualmente modificarle.

I volontari dovranno inserire delle ore di reperibilità. Nel momento in cui viene creato un intervento programmato o un intervento di emergenza, l'applicazione, sulla base delle reperibilità e della difficoltà dell'intervento, creerà una squadra temporanea formata da un certo numero di volontari e gli invierà una notifica. A questo punto i volontari parteciperanno all'intervento e potranno documentare tutto ciò che è stato fatto, caricare foto ed inserire i materiali utilizzati.

Il coordinatore ha il compito di inserire nuovi volontari tesserati nel sistema ed assegnarli ad una squadra esistente, oppure di creare nuove squadre. Nel

Qui ho inserito il caso d'uso Gestione magazzino, verificare sia corretto o ampliarlo

momento in cui il coordinatore crea una nuova squadra, dovrà inserire anche la zona a cui la squadra verrà assegnata.

L'applicazione fornirà anche la possibilità agli operatori di essere notificati con avvisi riguardanti bollettini meteo, terremoti, allerte della protezione civile nazionali ed eventi particolari provenienti da stazioni installate sul territorio. Le informazioni per generare gli avvisi vengono recuperate tramite le seguenti API:

- APRS.FI: sito utilizzato dai radioamatori per la condivisione di dati open. Il sito fornisce alcune API REST da cui è possibile richiedere informazioni su stazioni meteorologiche pubbliche;
- OpenWeatherMap: il sito permette di scaricare informazioni meteorologiche, previsioni e dati sulla qualità dell'aria in tutta Italia tramite delle API REST;
- INGV Earthquake event: permette di ricevere informazioni sui terremoti che si sono verificati in una particolare area geografica.
- Protezione Civile POP: web app che fornisce delle API tramite cui è possibile scaricare i bollettini giornalieri della protezione civile (recuperati dalla repository Github della protezione civile) riguardo al rischio idraulico ed idrogeologico e temporali;
- Recupero di dati da stazioni personalizzate posizionate da una squadra. La stazione fornirà una API dalla quale poter recuperare dati di interesse (per esempio dati meteorologici o livello di torrenti).

Le stazioni personalizzate installate dalla squadra saranno realizzate tramite prototipi Arduino/Raspberry e sensori (reali o simulati) ed esporranno un API REST alla quale richiedere le informazioni mediante un protocollo da noi definito. Ad esempio, si potrà realizzare un sistema di monitoraggio del livello dell'acqua di valli, per essere avvisati in caso di straripamento, oppure dati meteorologici (per esempio temperatura, umidità, vento) per avere informazioni dettagliate su un'area di particolare interesse (per esempio aree protette o risorse naturali).

Il caposquadra potrà decidere a quali servizi di notifica, riguardanti la zona della sua squadra, vuole sottoscrivere. Sarà possibile, infatti, richiedere di essere informati sulle condizioni meteo ed altre allerte derivanti da un'analisi dei dati (che sarà effettuata dalla nostra applicazione) di una/più stazione/i accessibili su APRS.FI, specificando il codice della centralina desiderata. Oppure, potrà scegliere di ricevere i bollettini meteo basati sulle previsioni di OpenWeatherMap (che fornisce previsioni più precise ma riguardanti una zona più generica rispetto all'area operativa della squadra). Oppure, in base alla località assegnata alla squadra potrà richiedere di essere avvisato in caso di terremoti in quella zona. Infine, si può richiedere di essere avvisati sui bollettini giornalieri emessi dalla protezione civile nazionale. Indicando poi le API di riferimento delle stazioni personalizzate e la tipologia del dato richiesto (temperatura, livello acqua, etc.) sarà possibile essere avvisati su particolari eventi derivanti da un'analisi dei dati ricevuti.

L'applicazione analizzerà i dati meteorologici provenienti dalle stazioni meteorologiche (su APRS.FI o personali) per generare previsioni a breve termine

e avvisare le squadre in caso di particolari fenomeni (vento forte = attenzione alla possibilità di incidenti dovuti alla caduta di piante, temperatura sotto lo zero = possibile ghiaccio, livelli di pioggia elevati = allagamenti, livello acqua fiume elevato = possibilità esondazioni).

All'interno dell'applicazione verranno anche utilizzate delle mappe (fornite da OpenStreetMap) per visualizzare il luogo degli interventi o la posizione dei volontari operativi. Inoltre, in fase di registrazione verranno utilizzate delle API (fornite dal sito <https://comuni-ita.herokuapp.com/>) che permettono di ricevere alcune informazioni (posizione, cap, codice istat, etc.) sui comuni italiani.

L'applicazione potrà essere utilizzata su dispositivi mobili o computer, dopo che l'utente ha effettuato l'autenticazione.

1.2 Casi d'uso

Di seguito vengono riportati i casi d'uso raggruppati in base all'attore. Gli attori in gioco sono i seguenti:

- volontario;
- caposquadra;
- coordinatore.

1.2.1 Casi d'uso: Volontario

- **Gestione account:** come volontario, voglio poter visualizzare e modificare le informazioni relative al mio account.
- **Login:** come volontario, voglio poter accedere al mio profilo personale.
- **Logout:** come volontario, voglio poter disconnettere il mio profilo personale dall'applicazione.
- **Gestione reperibilità:** come volontario, voglio poter indicare e modificare le mie ore di reperibilità.
- **Visualizzazione informazioni squadra:** come volontario, voglio poter visualizzare le informazioni relative alla mia squadra.
- **Visualizza intervento programmato:** come volontario, voglio poter visualizzare le informazioni relative ad un intervento programmato a cui sono stato assegnato.
- **Visualizza emergenza:** come volontario, voglio poter visualizzare le informazioni relative ad un intervento d'emergenza a cui sono stato assegnato.
- **Visualizza informazioni zona:** come volontario, voglio poter consultare le informazioni riguardanti il meteo, terremoti, qualità dell'aria, stazioni personalizzate relative alla zona assegnata alla mia squadra.
- **Inserimento informazioni report:** come volontario, voglio poter arricchire la documentazione relativa ad un intervento, aggiungendo osservazioni e foto.

- **Segnalazione operatività:** come volontario, voglio poter segnalare quando sono operativo e mi trovo sul luogo dell'intervento.

1.2.2 Casi d'uso: Caposquadra

Il caposquadra include tutti i casi d'uso del volontario più i seguenti:

- **Gestione report intervento programmato:** come caposquadra, voglio poter visualizzare e modificare i report riguardanti gli interventi programmati creati da me.
- **Gestione report emergenza:** come caposquadra, voglio poter visualizzare e modificare i report riguardanti le emergenze create da me.
- **Gestione intervento programmato:** come caposquadra, voglio poter creare, modificare o eliminare un intervento programmato.
- **Gestione intervento emergenza:** come caposquadra, voglio poter creare, modificare o eliminare un intervento d'emergenza.
- **Gestione magazzino:** come caposquadra, voglio poter visualizzare e modificare le giacenze del magazzino.
- **Notifiche allarmi zona:** come caposquadra, voglio poter decidere su quali eventuali allarmi, relativi alla mia zona, essere notificato.
- **Gestione informazioni relative alla zona:** come caposquadra, voglio poter definire quali informazioni (meteorologiche, terremoti, stazioni personalizzate, bollettini protezione civile) ricevere.

1.2.3 Casi d'uso: Coordinatore

- **Gestione account:** c coordinatore, voglio poter visualizzare e modificare le informazioni relative al mio account.
- **Login:** come coordinatore, voglio poter accedere al mio profilo personale.
- **Logout:** come coordinatore, voglio poter disconnettere il mio profilo personale dall'applicazione.
- **Inserimento volontari:** come coordinatore, voglio poter inserire nuovi volontari.
- **Gestione squadre:** come coordinatore, voglio poter creare, modificare ed eliminare le squadre.

1.3 Diagramma dei casi d'uso

In figura 1 è stato riportato il diagramma dei casi d'uso. In questo diagramma sono stati inseriti i casi d'uso descritti in precedenza mettendoli in correlazione con il corrispettivo attore. È importante notare che la figura del *caposquadra* è una specializzazione del *volontario* e di conseguenza eredita tutti i suoi casi d'uso, oltre ad implementarne altri personalizzati.

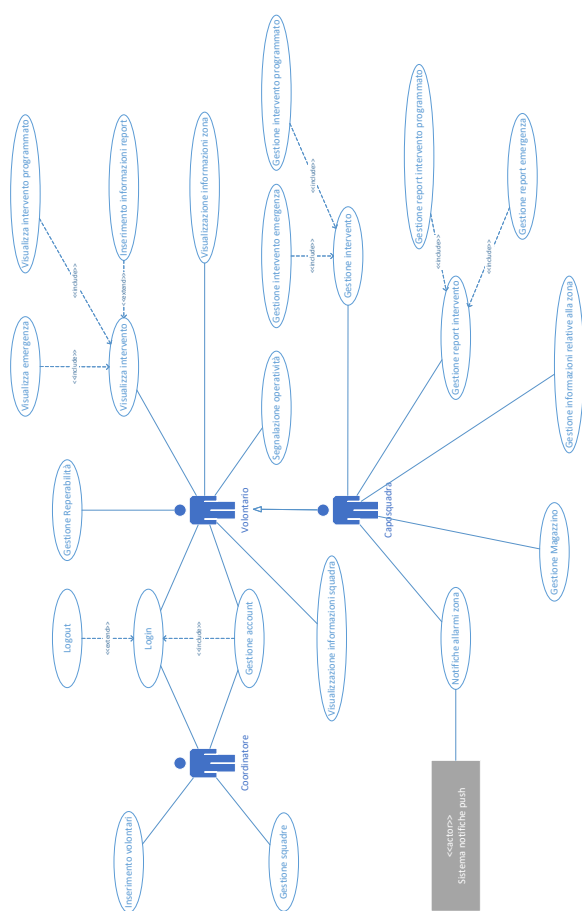


Figura 1: Diagramma dei casi d'uso.

1.4 Architettura del sistema

L'architettura del sistema è stata formalizzata attraverso due deployment diagram differenti: il primo, riportato in figura 2, rappresenta il sistema tramite una notazione a stile libero, mentre il secondo, riportato in figura 3, rappresenta il sistema tramite lo stile UML. In particolare, in entrambi i diagrammi è possibile identificare un pattern architetturale *three thier*. All'interno del *presentation layer* ci sono le applicazioni lato client che vengono utilizzate dagli attori. Nell'*application layer*, invece, ci sono due macchine distinte:

- su una macchina c'è un **Web Server Java**, che si occupa della gestione di richieste tramite protocollo *HTTP/Rest* da parte del client;
- sull'altra macchina ci sono due componenti autonome, un **Data Fetcher** che si occupa di recuperare informazioni dai data server tramite API e le inserisce all'interno del database, ed un **Data Analyzer** che si occupa di effettuare delle analisi dei dati acquisiti con lo scopo di identificare possibili situazioni di emergenza.

Nel *data layer* troviamo invece un database all'interno del quale sono inserite tutte le informazioni inerenti il sistema.

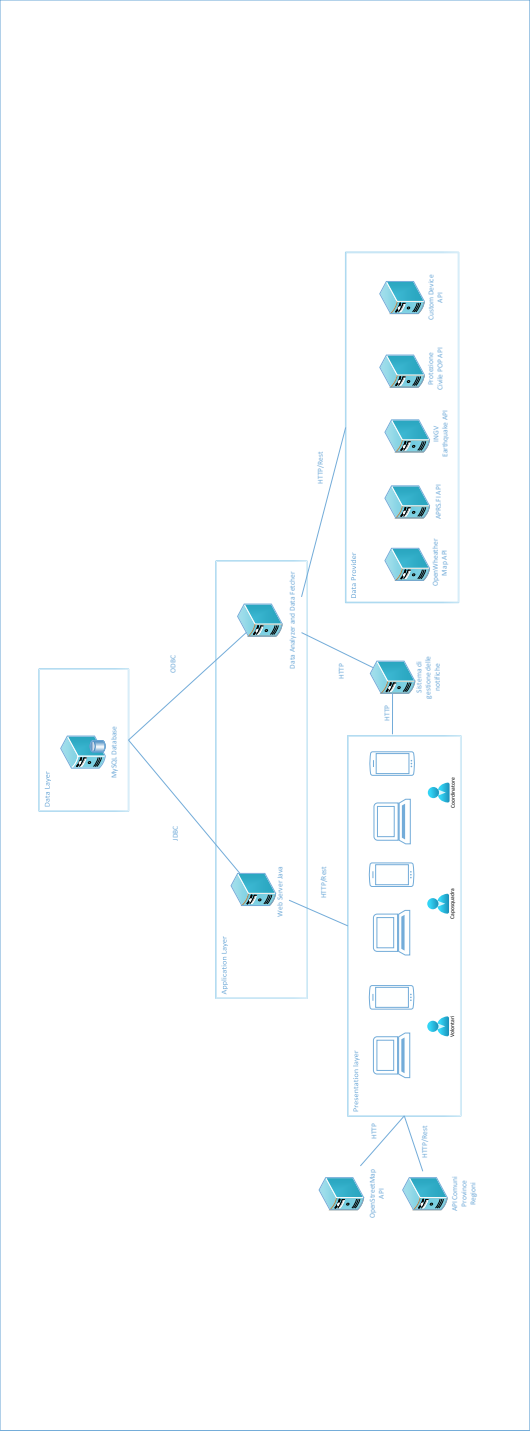


Figure 2: Deployment diagram in stile libero.

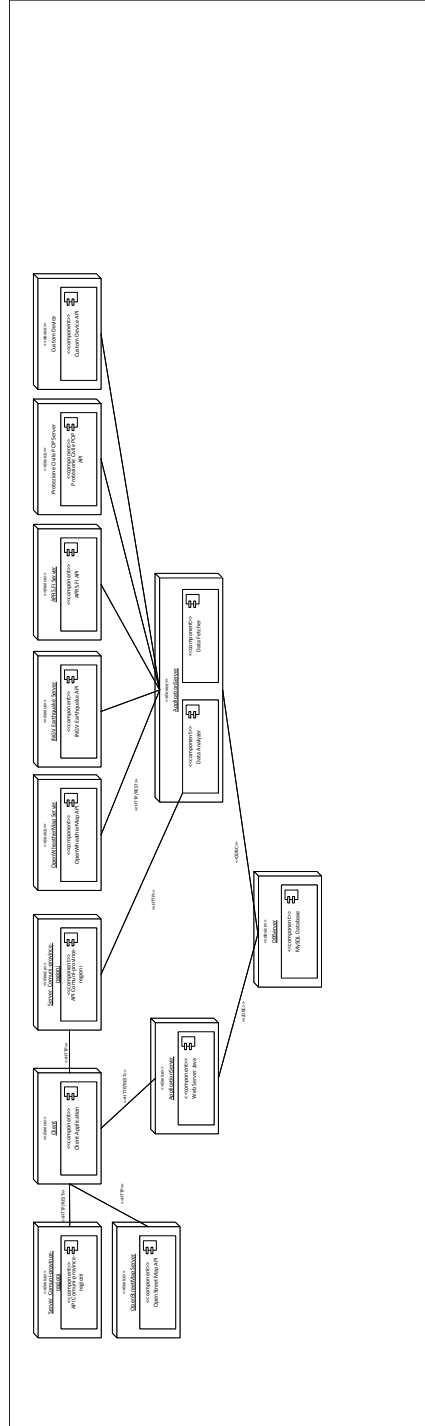


Figura 3: Deployment diagram in stile UML.

1.5 Toolchain e tecnologie utilizzate

Per la realizzazione del sistema verranno utilizzati i seguenti strumenti:

- **Modellazione:**
 - diagramma dei casi d'uso, deployment diagram, component diagram, class diagram, sequence diagram, diagramma entità-relazione: Visio.
- **Implementazione Applicazione Client:**
 - Linguaggio di programmazione: C++.
 - IDE: Qt.
 - Interfaccia grafica: QML.
- **Implementazione Web Server:**
 - Linguaggio di programmazione: Java.
 - IDE: Eclipse.
 - Framework: Spring.
 - Analisi statica: JUnit.
 - Analisi dinamica: Stan4J.
- **Implementazione Data Analyzer e Data Fetcher:**
 - Linguaggio di programmazione: DA DECIDERE.
 - IDE: DA DECIDERE.
- **Implementazione Database:**
 - Tipologia: relazionale.
 - Database: MySQL.
- **Documentazione, versioning e organizzazione del team:**
 - Documentazione: Latex, con scrittura tramite editor TexStudio.
 - Versioning: GitHub.
 - Organizzazione del Team: Microsoft Teams.