Stalker

corso di **INGEGNERIA del SOFTWARE**
prof. Tullio Vardanega

Università degli Studi di Padova

capitolato per il progetto
didattico 2019/2020 proposto da

Indice

Chi è Imola Informatica

Imola informatica si occupa, da più di 35 anni, di consulenza IT con l'obiettivo di aiutare i propri clienti a fare dell'innovazione tecnologica uno strumento strategico nei processi organizzativi e nella progettazione di sistemi informativi complessi. I nostri clienti sono prevalentemente gruppi finanziari e assicurativi ma anche aziende e startup. La nostra mission è favorire il continuo cambiamento tecnologico e culturale per essere sempre allineati con gli ultimi aggiornamenti in ambito di information technology.

La nostra esperienza ci ha fatto comprendere come la chiave per produrre soluzioni di successo sia la collaborazione. In questo senso riteniamo importante valorizzare il mondo e la tecnologia open source, incoraggiandone l'impiego (personale, aziendale, da parte dei nostri assistiti) e partecipando attivamente allo sviluppo[1].

Il progetto proposto dall'Università di Padova (Corso di Ingegneria del Software) rappresenta per Imola Informatica una opportunità per confrontarsi con le nuove generazioni di programmatori, facendosi conoscere e creando nuovi legami e network di interesse.

Il problema

La normativa vigente in materia di sicurezza regolamenta la gestione delle presenze nei locali pubblici ed aperti al pubblico prevedendo una serie di precauzioni ed adempimenti volti a garantire uno sfollamento sicuro in caso di emergenze e bisogno (capienza massima, uscite di sicurezza con determinate caratteristiche, via di fuga libere e controllate, etc etc…). In questo senso diventa onere non eludibile tracciare il numero di persone presenti all'interno dei locali. Questo dato non è tuttavia così immediato da ottenere e,

[1] * Coerentemente con ciò è nostra volontà rendere disponibile il software prodotto in questo progetto attraverso un repository pubblico accompagnato da una licenza di tipologia MIT License, Apache License o GNU General Public License (GPL) 3.

ancora più, da monitorare.

Nello specifico, in questo progetto si prendono in considerazione due case-study particolari rispetto ai quali si vuole individuare una possibile risposta a tale obbligo:

- Tracciamento delle singole presenze dei dipendenti nei differenti luoghi di lavoro di Imola informatica (ad oggi effettuato tramite firma cartacea o apposito flag nel gestionale aziendale Trattandosi di una gestione attiva è facile che i dipendenti si scordino di effettuarla nei momenti corretti facendo risultare ingressi in tarda mattina o presenze praticamente perenni nelle sedi aziendali);

- Tracciamento delle persone presenti in una importante struttura fieristica (Fiera di Verona), sia globalmente che nei singoli padiglioni (ad oggi i tornelli permettono il solo tracciamento globale ma non quello puntuale).

Di seguito proponiamo alcune definizioni fondamentali per lavorare su questo progetto:

- **Organizzazione**: soggetto che ha interesse a tracciare le presenze delle persone all'interno dei propri luoghi, in maniera anonima o autenticata.
- **Luogo**: spazio fisico indentificato da un insieme di coordinate geografiche (padiglione di una fiera/ufficio di una azienda/aula universitaria/piazza). Ciascun luogo è riconducibile ad una organizzazione.
- **Tracciatura**: rilevamento della presenza all'interno di un luogo. Può essere:
    - anonima: quando il soggetto non è autenticato nell'organizzazione di riferimento;
    - autenticata: quando il soggetto è autenticato nell'organizzazione di riferimento.

La soluzione

Per la progettazione della soluzione abbiamo previsto, come assunto di base non derogabile, il fatto che le persone da tracciare siano in possesso di uno smartphone (Android o Ios) e nelle condizioni di installare un'applicazione. L'obiettivo è quello di sviluppare un'applicazione in grado di segnalare ad un server dedicato sia l'ingresso che l'uscita dell'utilizzatore dalle aree d'interesse (in modalità anonima o meno a seconda delle esigenze).

Nel server deve essere possibile gestire più organizzazioni. Per ogni organizzazione:

- deve essere possibile gestire molteplici luoghi;
- è necessario definire se prevedere una tracciatura autenticata e l'eventuale procedura di autenticazione.

Al momento della registrazione dell'applicazione viene effettuata una richiesta al server per recuperare la lista delle organizzazioni alle quali è possibile registrarsi.

In riferimento ai casi d'uso presentati nel paragrafo precedente:

- per il caso "fiera" è sufficiente definire una organizzazione che non preveda autenticazione;
- per il caso "Imola Informatica" è invece necessario definire una organizzazione che supporti una autenticazione tramite LDAP[2] per tracciare la presenza dei singoli dipendenti, non solo in modalità anonima ma anche esplicita.

Il server deve essere correlato di una UI con una procedura di autenticazione che permetta di gestire le autorizzazioni puntualmente. In base al livello di autorizzazione dell'utente deve essere possibile effettuare operazioni di ricerca ed analisi di dati sulle diverse organizzazioni. L'analisi base consiste nel monitoraggio del numero di persone presenti nei luoghi in un determinato lasso temporale ed in tempo reale. Per gli utenti con l'autorizzazione specifica dev'essere anche possibile effettuare query di monitoraggio per singolo utente all'interno delle organizzazioni (es: quante volte l'utente "x" si è presentato a lavoro questo mese e in quali fasce orarie).

L'applicazione cellulare (su Android o IOS) deve permettere le seguenti operazioni:

- recupero lista organizzazioni (Refresh manuale e/o temporizzato);
- login nell'organizzazione con eventuale autenticazione;
- storico degli accessi;
- visualizzazione in tempo reale della propria presenza o meno all'interno di un luogo monitorato e cronometro del tempo trascorso al suo interno;
- predisposizione di un pulsante "anonimo" che permetta, all'interno di una organizzazione che prevede l'autenticazione, di risultare presente in maniera anonima.

Le comunicazioni tra applicazione cellulare e server avvengono solo al momento d'ingresso ed uscita dai luoghi designati, altre informazioni relative alla posizione nei luoghi non designati non sono tracciate al fine di garantire la privacy degli utilizzatori.

Al fine di supportare entrambe le tipologie di funzionamento il server dev'essere in grado di scalare in base al numero di utilizzatori in modo dinamico sia in aggiunta che in riduzione. *Esempio: se un server che offre l'argomento di una fiera come quella di Verona, nella fattispecie la lista dei padiglioni della fiera con le rispettive coordinate geografiche, dev'essere in grado di scalare nel weekend del Vinitaly al fine di poter supportare i milioni di visitatori ed in seguito dev'essere in grado di ri-scalare a ritroso nei giorni successivi al picco in modo da non sprecare risorse.* Le risorse messe a disposizione per il server non sono illimitate, è pertanto necessario identificare anche una soluzione (rallentamenti, code lato server, cash locali o altre

[2] https://it.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol
https://it.wikipedia.org/wiki/OpenLDAP

possibilità individuate dagli studenti) che permetta il proseguimento del servizio anche nei casi di richieste che vanno oltre la capacità a pieno regime.

Quando si parla di scalabilità di carico le alternative da prendere in considerazione sono due: scalabilità verticale e scalabilità orizzontale.

- La scalabilità verticale è una scalabilità hardware in cui si aumentano le risorse fisiche a disposizione (più Ram, più HD, più CPU, etc.);
- La scalabilità orizzontale consiste, di fatto, nell'aggiunta di più istanze dell'applicazione con un load balancer sopra per la ridistribuzione del carico fra di esse.

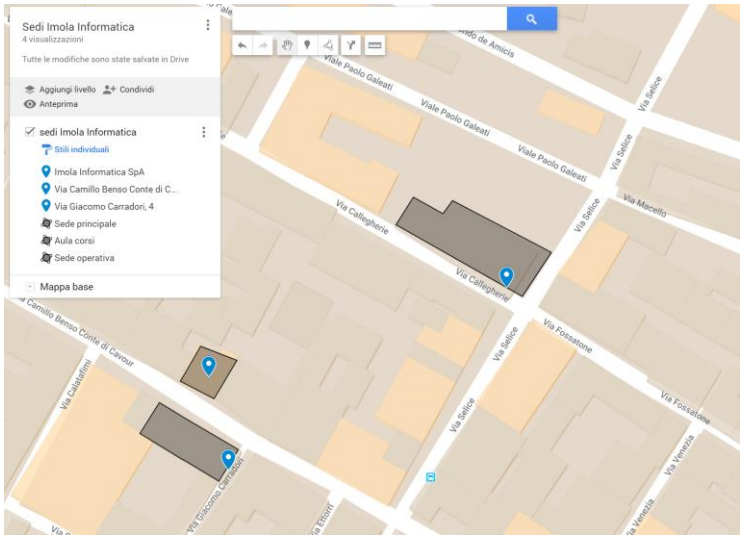
Nonostante possa sembrare più conveniente una scalabilità di tipo verticale, all'aumentare delle risorse richieste questa, data la natura del mercato per le componenti hardware, comporta un aumento di costo esponenziale. La scalabilità orizzontale, invece, richiede sempre le stesse risorse mantenendo un aumento dei costi lineare che in base al numero di istanze aumenta in maniera costante e controllata.

In generale le due tipologie di scalabilità presentano vantaggi e svantaggi in base al caso d'uso da prendere in considerazione, per il caso specifico di questo progetto si è ritenuto di optare per una scalabilità di tipo orizzontale.

I casi d'uso

I casi d'uso del progetto presentano requisiti minimi, di seguito evidenziati in grassetto, e requisiti opzionali desiderati, evidenziati attraverso una sottolineatura.

Caso d'uso A: Imola Informatica – Monitoraggio autenticato



In questo caso la necessità primaria è monitorare la presenza dei dipendenti all'interno delle sedi aziendali sparse per l'Italia. E' prevista la presenza di due "macro" tipologie di attori:

- Amministratore:

  attraverso l'interfaccia offerta dal server centrale deve essere in grado di:

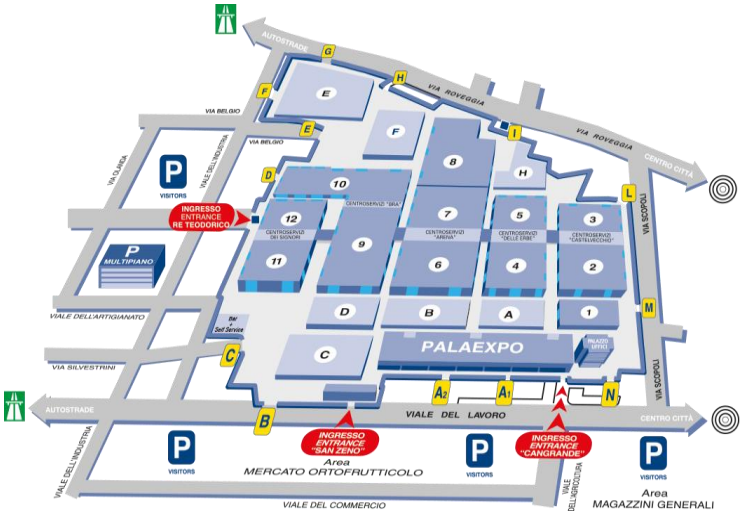
  - **loggarsi all'interno del server con un'utenza personale;**
  - **creare, modificare o eliminare organizzazioni;**
    - **l'organizzazione deve consentire l'autenticazione tramite LDAP;**
    - **la visualizzazione deve mostrare solo l'organizzazione di cui si è amministratori;**
    - **la modifica deve permettere di modificare i luoghi (vedi sotto) e gli utenti amministratori possibili:** il creatore è l'owner dell'organizzazione ma ha la possibilità di identificare altri utenti amministratori come gestori dell'organizzazione (aventi la possibilità di gestire i luoghi a loro volta) o come visualizzatori (aventi solo le abilitazioni di visualizzazione ma non di modifica).
  - **aggiungere, modificare e rimuovere luoghi inserendo le coordinate geografiche** o, opzionalmente, evidenziando le aree in una mappa;
  - **configurare i dettagli del server LDAP che le applicazioni dovranno utilizzare per registrarsi all'organizzazione Imola Informatica;**
  - **inviare una richiesta di aggiornamento a tutte le applicazioni in modo tale da fargli scaricare la lista aggiornata delle organizzazione e dei rispettivi luoghi;**
  - **monitorare in ogni momento il numero di dipendenti presenti in tutti i singoli luoghi e nell'organizzazione nel suo complesso;**
  - **effettuare ricerche sugli accessi di uno specifico dipendente**;
  - estrapolare un report, sotto forma tabellare, che evidenzi gli accessi, le ore trascorse all'interno dei luoghi, e quali luoghi sono più frequentati.

- Dipendente:

  attraverso l'applicazione cellulare deve poter :

  - **scaricare la lista di tutte le organizzazioni;**
  - **autenticarsi nell'organizzazione, qualora previsto;**
  - aver accesso allo storico degli accessi;
  - visualizzare in tempo reale la propria presenza o meno all'interno di un luogo monitorato e il cronometro del tempo trascorso al suo interno (da visualizzare solo in caso di luogo d'interesse);
  - **usufruire di un pulsante "anonimo" che permetta, all'interno di una organizzazione che prevede l'autenticazione, di risultare presente in maniera anonima.**

Caso d'uso B: Fiera di Verona – Monitoraggio anonimo



Il caso d'uso per la fiera di Verona è sostanzialmente identico a quello descritto per Imola informatica, l'unica differenza è che non è necessario configurare il server LDAP in quanto l'argomento dev'essere segnato come argomento pubblico.

Scelte e preferenze tecniche

Imola Informatica da sempre è interessata e costantemente impegnata ad esplorare nuove soluzioni tecnologiche all'avanguardia ~~e~~, pertanto, predilige non imporre tecnologie specifiche per lo sviluppo del server o della UI.

Al fine di poter utilizzare al meglio le professionalità dei referenti aziendali, la loro esperienza e il loro background, in riferimento ad alcune scelte tecniche ci si sente di **consigliare** alcune opzioni:

- utilizzo di Java (versione 8 o superiori), python[i] o nodejs[ii] per lo sviluppo del server back-end;
- utilizzo di protocolli asincroni per le comunicazioni app mobile-server;
- utilizzo del pattern di Publisher/Subscriber[iii];
- utilizzo dell'IAAS Kubernetes o di un PAAS, Openshift o Rancher, per il rilascio delle componenti del Server nonché per la gestione della scalabilità orizzontale.

Al fine di garantire il raggiungimento degli obiettivi minimi del progetto **si richiede**:

- che il server esponga, in aggiunta ad eventuali altri protocolli richiesti per l'interazione con il servizio specifico, delle API Rest[iv] attraverso le quali sia possibile utilizzare l'applicativo. In alternativa è possibile utilizzare gRPC[3] come soluzione alternativa al Rest;
- una precisione sufficiente a certificare la presenza della persona all'interno degli edifici. L'utilizzo del GPS riduce in modo rilevante l'autonomia dei cellulari, l'applicazione è da sviluppare in maniera

[3] https://grpc.io : open-source RPC framework sviluppato inizialmente da Google che sta prendendo una fetta di mercato sempre più importante per quanto riguarda comunicazioni in architetture di microservizi, mobile e siti web.

tale da bilanciare nel miglior modo possibile batteria e posizione (si consiglia un soluzione ibrida network-GPS). E' richiesto un resoconto delle scelte fatte e dei test effettuati per garantire il miglior rapporto raggiunto;

- che tutte le componenti applicative siano correlate da test unitari e d'integrazione. Inoltre, è richiesto che il sistema venga testato nella sua interezza tramite *test end-to-end*. I punteggi minimi verranno concordati una volta individuate, con l'aiuto dei referenti aziendali, le metriche software più adeguate.

Eventuali alternative potranno venire discusse con i gruppi aderenti al capitolato ed i referenti aziendali durante lo svolgimento del progetto attraverso i canali di comunicazione descritti nella sezione Riferimenti aziendali.

Obiettivi del progetto

Affinché il progetto possa dirsi concluso con esito positivo è necessario che siano raggiunti i seguenti obiettivi:

- server, completo di UI, in grado di soddisfare i requisiti obbligatori evidenziati nel caso d'uso "Imola Informatica – Monitoraggio autenticato";
- applicazione mobile, (IOS o Android) che permetta di soddisfare i requisiti obbligatori evidenziati nel caso D'uso "Imola Informatica – Monitoraggio autenticato";
- test di carico che dimostrino il corretto funzionamento in situazioni normali, di carico e di sovraccarico;
- copertura di test >= 80% correlata di report;
- report dei test effettuati relativamente all'ottimizzazione della precisione dell'applicazione rispetto al consumo della batteria dei cellulari;
- documentazione su:
    - scelte implementative e progettuali effettuate e relative motivazioni;
    - problemi aperti e eventuali soluzioni proposte da esplorare.

Si ritiene di interesse citare qui altri due risultati auspicabili seppur non strettamente necessari al fine del completamento del progetto:

- cifrare tutte le comunicazioni fra App e Server in modo tale da garantire la validità delle informazioni;
- fornire un'analisi rispetto al carico massimo supportato in numero di utenti e di quale sarebbe il servizio cloud più adatto per supportarlo analizzando prezzo, stabilità del servizio ed assistenza. (supponendo di di-

sporre di massimo 2 CPU e 1Gi[4] per istanza del server). Identificare inoltre in che modo il costo varierebbe in caso di aumento delle istanze e da quale momento la scalabilità orizzontale risulti effettivamente più conveniente di una scalabilità verticale.

Riferimenti aziendali

L'azienda, per il progetto, mette a disposizione figure di diverso livello in modo da supportare al meglio tutte le esigenze degli studenti.

In particolare, seguiranno il progetto:

- un professionista con più di 3 anni d'esperienza in azienda per fornire il supporto dal punto di vista tecnico. Fungerà da interfaccia principale con i gruppi.
- un professionista con oltre 20 anni d'esperienza, che fungerà da responsabile del progetto lato azienda e fornirà ulteriore supporto architetturale e tecnico in caso di bisogno.

Inoltre l'azienda mette a disposizione, in caso di bisogno, server nei quali gli studenti potranno effettuare le installazioni dei componenti applicativi sviluppati.

A causa della distanza tra l'università e l'azienda, le comunicazioni fra i gruppi e i referenti aziendali avverranno, principalmente, tramite chat (preferibilmente Telegram) o tramite videochiamate (Hangout o Skype). In caso di necessità sarà comunque possibile organizzare incontri di persona e/o definire ulteriori strumenti di comunicazione.

Links Utili

Di seguito una serie di link che potrebbero aiutare nello sviluppo del progetto:

- ✓ https://github.com/gvellut/MapAlarmist
  applicazione Android opensource che permette di far suonare una sveglia all'accesso in un'area geografica specifica;
- ✓ https://developer.apple.com/documentation/corelocation/getting_the_user_s_location
  documentazione ufficiale Apple dei meccanismi per ottenere la posizione dell'utente;
- ✓ https://medium.com/faun/scaling-applications-in-the-cloud-52bb6dfbac4e
  interessante panoramica di possibili meccanismi di scalabilità;
- ✓ https://owntracks.org/
  un progetto open source, disponibile sia per Android che per IOS, di un'applicazione per monitora-

[4] https://en.wikipedia.org/wiki/Kibibyte

re e condividere la propria posizione. Si consiglia una lettura approfondita della sua documentazione in quanto presenta molteplici aspetti in comune con il progetto da noi proposto;

- ✓ https://steelkiwi.com/blog/mobile-application-security-best-practices-for-app-developers
  una panoramica ad alto livello di alcune delle best practices per il garantire la sicurezza della propria applicazione mobile.

---

[i] https://www.python.org/

[ii] https://nodejs.org/it/

[iii] In questo pattern, mittenti e destinatari di messaggi dialogano attraverso un tramite, che può essere detto *dispatcher* o *broker*. Il mittente di un messaggio (detto *publisher*) non deve essere consapevole dell'identità dei destinatari (detti *subscriber*); esso si limita a "pubblicare" (in inglese *to publish*) il proprio messaggio al *dispatcher*. I destinatari si rivolgono a loro volta al *dispatcher* "abbonandosi" (in inglese *to subscribe*) alla ricezione di messaggi. Il *dispatcher* quindi inoltra ogni messaggio inviato da un *publisher* a tutti i *subscriber* interessati a quel messaggio.

In genere, il meccanismo di sottoscrizione consente ai *subscriber* di precisare nel modo più specifico possibile a quali messaggi sono interessati. Per esempio, un *subscriber* potrebbe "abbonarsi" solo alla ricezione di messaggi da determinati *publisher*, oppure aventi certe caratteristiche.

Questo schema implica che ai *publisher* non sia noto quanti e quali siano i *subscriber* e viceversa. Questo può contribuire alla scalabilità del sistema. (fonti: https://it.wikipedia.org/wiki/Publish/subscribe)

[iv] https://www.restapitutorial.com/