

Relazione del Progetto di Ingegneria del Software

Durante le prime riunioni volte ad analizzare le indicazioni abbiamo discusso sullo **scopo** del progetto che poi avremmo chiamato "MuniciPath" e siamo giunti a queste conclusioni:

- L'obiettivo è un sito di contribuzione nazionale, suddiviso per i singoli **comuni registrati**.
- Ogni comune e ogni utente devono essere inizialmente registrati dai **gestori della piattaforma**.
- Ogni comune deve avere un "proprietario", detto **curatore**. Un curatore deve essere un utente registrato. Un comune non può avere più di un curatore e un curatore non può gestire più di un comune.
- Un curatore può eleggere degli "aiutanti", detti **moderatori**, utenti interni alla gestione del comune che hanno lo scopo di amministrarlo insieme al curatore.
- Abbiamo deciso che un comune contiene un certo numero di **Punti**, che a sua volta contiene un certo numero di **Post**. Ogni post ha un titolo, un testo, delle immagini, un **autore** e un **tipo**. Solo il titolo, l'autore e il tipo sono obbligatori.
- I tipi di post sono: **Social**, **Authority**, **Health**, **Turistic**, **Event** o **Contest**. Gli ultimi due contengono altre due informazioni obbligatorie: il momento di inizio e quello di fine.
- I post di tipo "temporale" possono essere anche persistenti, che significa che dopo la loro fine questi non vengono automaticamente cancellati. Ciò è stato pensato per riproporli tramite una modifica.
- Un punto rappresenta semplicemente un raccoglimento di post. Indica dove si trovano in una mappa un certo numero di post e (tramite il colore) il tipo della maggior parte di loro. Dunque, non può esistere un "punto vuoto".
- Esiste un punto speciale, che rappresenta l'intero comune. Ogni post che è contenuto in esso quindi riguarda tutta l'area circoscritta.

Successivamente abbiamo revisionato la lista degli **attori** come segue:

- **Gestori della piattaforma** possono convalidare account in attesa di essere convalidati. Inoltre, possono gestire i comuni. Ciò significa che possono crearli, modificarli o eliminarli.
- **Curatore** gestisce i contenuti e le autorizzazioni rispetto al proprio comune. Può anche visionare le statistiche comunali. Quando crea contenuti nel proprio comune appaiono come "creati dal comune".
- **Moderatore** ha lo stesso controllo del curatore in fatto di contenuti ma può gestire l'autorizzazione solo di utenti diversi dal curatore o gli altri moderatori.
- **Contributor Autorizzato** può liberamente creare contenuti di ogni tipo, ma può modificare ed eliminare solo i propri.
- **Contributor Non Autorizzato** invece che crearli, questo tipo di contributor può solo proporre la creazione. Anche le modifiche devono passare sotto approvazione. Invece, le eliminazioni non lo richiedono. Infine, non possono creare contenuti di tipo "Contest". Ad accettare o rifiutare le richieste sono i moderatori e il curatore.
- **Turista Autenticato** possono seguire contributor di loro piacimento e salvare e recensire post. Ricevono segnalazioni di eventi che salvano e notifiche di nuovi contenuti relativi ai contributors che seguono. Possono fare richiesta ad un comune di diventare contributor. Perdono i privilegi, elencati precedentemente, nei confronti di un comune, se l'amministrazione li **limita**.
- **Turista** possono navigare e vedere ogni contenuto del sito. A loro piacimento possono anche condividerlo.
- **Meteo** servizio per le previsioni meteo dei luoghi di interesse.
- **OSM** servizio per l'ottenimento di mappe territoriali.
- **Social Esterni** social network atti alla condivisione di contenuti di MuniciPath.

Dopo un'attenta riflessione abbiamo deciso di aggiungere la facoltà di creare contest unicamente al Contributor Autorizzato, vista la dose di fiducia che questa carica rappresenta. Inoltre, abbiamo pensato che la possibilità di condividere su social esterni fosse più interessante se condivisa con tutti gli utenti.

Dopo varie riflessioni abbiamo deciso di accorpare il concetto di itinerari ed esperienze negli **insiemi**, distinguendo i due tipi

semplicemente come “ordinati e non ordinati”. Un insieme segue le stesse regole di creazione di un post, e contiene: titolo, autore, se è ordinato e i post contenuti. Alcuni insiemi sono a tempo e, come i post, hanno anche un momento di inizio ed uno di fine. Anche loro possono essere persistenti. Ogni post mostra un elenco di tutti gli insiemi di cui fa parte, elencati in ordine cronologico.

Abbiamo anche studiato un sistema di **notifiche**, diviso in quattro tipi: segnalazioni, esito pending, novità ed evento.

Le segnalazioni provengono dai turisti autenticati e possono essere lette dallo staff del comune (quindi dal curatore e dai suoi moderatori).

L'esito pending arriva quando un membro dello staff approva o rifiuta una richiesta in pending. Il messaggio arriva all'autore e generalmente riguarda il motivo del rifiuto.

Le novità arrivano quando un contributor (o un comune) con dei followers pubblica un nuovo post o insieme. Tutti i followers ricevono la notifica.

Infine, gli eventi arrivano agli utenti che salvano un post di tipo evento. Quando l'autore desidera ricordare l'evento, tutti ricevono una notifica. Rientrano in questa categoria anche le notifiche per l'esito di un contest.

Un **contest** ha come “momento di inizio” quello in cui è stato creato ed è sempre persistente. Chiunque può contribuire al contest inserendo un testo e/o contenuti multimediali, ma solo se autenticati e solo una volta per account. Una volta arrivato il “momento di fine” non è più possibile partecipare e l'autore del contest può decidere il vincitore.

La facoltà di creare contenuti è stata associata ai solo ai **contributor** ed allo **staff del comune** dopo un' attenta riflessione: **la piattaforma deve essere un luogo adeguato e sotto controllo**. Da ciò si evince che, qualsiasi sia il contributo che si vuole dare per valorizzare il territorio, debba provenire da fonti autorizzate e sicure, per evitare disinformazione e/o evitare scherzi di cattivo gusto. Ciò non toglie che anche le fonti autorizzate possano rappresentare un rischio, ma ,per lo meno, sono una “*minaccia*” limitata e sotto controllo da parte dell'amministrazione del comune.

Excursus del processo unificato applicato al progetto

Iterazione 1

Durante la prima iterazione ci siamo concentrati sul C.R.U.D. dei post da parte dei contributors autorizzati e sulla navigazione del sito. Abbiamo definito cosa deve fare il sistema, quindi essere in grado di fornire agli utenti informazioni sul territorio che vogliono visitare, tramite punti geolocalizzati e tipizzati, sfruttando OSM. Alla fine dell'iterazione abbiamo prodotto l'implementazione in java delle classi che si occupano di questi fondamentali compiti.

Iterazione 2

Durante la seconda iterazione abbiamo sviscerato i concetti di insieme, ruolo e pending, aggiungendo la dimensione temporale a tutto il progetto. Durante questo lasso di tempo abbiamo introdotto per la prima volta il concetto di **turista limitato**, un utente che ha perso la possibilità di partecipare alle attività comunali nei confronti di uno o più comuni.

Sempre parlando di ruoli, abbiamo definito precisamente le caratteristiche del moderatore e le logiche di autorizzazione. Si sono poi decisi dei limiti logici: il comune è un insieme finito di punti, che possono essere caratterizzati da post, scegliendo un punto specifico; un punto avrà almeno un post, anche senza foto e descrizione (basta avere almeno un carattere nel titolo).

A fine iterazione siamo arrivati a sviluppare un prototipo completo e funzionante, per cui abbiamo deciso di terminare la **fase di analisi** e di iniziare, con la successiva iterazione, la **fase di sviluppo**.

Iterazione 3

Durante la terza iterazione abbiamo continuato lo sviluppo, cominciando a definire degli elementi, non fondamentali, ma al contempo importanti per la fruibilità e l'utilizzo del software che avevamo in mente.

Diciamo che in questa iterazione ci siamo dedicati un po' di più alla UX, visto che la struttura principale è stata ultimata nell'iterazione precedente.

Abbiamo, quindi, implementato le seguenti caratteristiche:

- **Un sistema di follow/unfollow:** un *turista autentico* può seguire un *contributor*, autorizzato o non. Ci si può seguire anche fra contributors.
- **Lista dei preferiti:** un utente può salvare un contenuto nella sua lista dei preferiti.
- **Recensione dei post:** abbiamo implementato la possibilità di lasciare un feedback ai post (votando da 1 a 5 stelle)
- **Servizio di notifica:** abbiamo pensato a *come* e *quando*, un utente debba essere notificato, o possa notificare qualcuno.
- **Segnalazione di contenuti inappropriati:** un utente ha la facoltà di segnalare un contenuto da lui ritenuto inopportuno. (ovviamente abbiamo pensato a come questa cosa possa essere un'arma a doppio taglio per coloro che vogliono solo creare disagio, quindi abbiamo pensato anche a delle contromisure adeguate in caso di comportamenti scorretti)
- **Una migliore gestione dei post da parte dello "staff" del comune:** ora i membri dell'amministrazione comunale (curatore e moderatori) possono interagire coi contenuti ritenuti inopportuni modificandoli o eliminandoli, notificando l'autore del gesto compiuto.
- **Analitiche comunali:** caratteristica che forse è stata la più complessa da implementare durante l'iterazione, vista la mole di dati da manipolare. Sfruttando la statistica ([Wilson score](#)) e gestendo i dati con modularità siamo riusciti ad ottenere un ottimo risultato in termini di analytics utili al comune (e.g. Top 5 contenuti più visti negli ultimi n mesi, con $1 \leq n \leq 12$)

Definendo questi dettagli e raffinando le ultime cose, ci siamo infine dedicati al testing.

Iterazione 4

Durante quest'ultima iterazione ci siamo concentrati sul lato frontend del progetto e sulla comunicazione tra le due parti tramite chiamate REST.

Abbiamo studiato il funzionamento e la struttura delle chiamate http e ci siamo dedicati alla raffigurazione dei dati desiderati in base al contesto.

Dopo attente analisi, abbiamo deciso di implementare un sistema basato su dei json web token (JWT), per gestire direttamente l'autenticazione degli utenti, e un corrispettivo sistema di salvataggio nel lato front end basato sui cookie.

Come risultato abbiamo ottenuto una web application in grado di navigare su una mappa interattiva, fornendo delle funzionalità in base all'autorizzazione dell'utente loggato.

Infine, dopo aver inserito le funzionalità essenziali nel sito internet, ci siamo dedicati alla trasmissione e alla visualizzazione di contenuti multimediali tra il back end in spring e il front end in angular.