

# Università degli Studi di Modena e Reggio Emilia

Dipartimento di Scienze Fisiche, Informatiche e Matematiche

Corso di laurea in Informatica

## Engim report service

Relatore:

Prof Claudia Canali

Candidato:

Dumitru Frunza

Anno accademico 2021/2022



# Contents

<b>1</b>	<b>Engim Srl</b>	<b>1</b>
1.1	L'azienda . . . . .	1
1.2	Infrastruttura . . . . .	2
1.3	Microservizi . . . . .	2
1.4	Sistema automatico di generazione di report . . . . .	3
<b>2</b>	<b>Requisiti del progetto</b>	<b>4</b>
2.1	Descrizione . . . . .	4
2.2	Scelte del linguaggio . . . . .	4
<b>3</b>	<b>Sviluppo</b>	<b>5</b>
3.1	Workflow . . . . .	5
3.2	Creazione di un file e salvataggio su S3 . . . . .	5
3.3	Stampa di un file pdf da un json . . . . .	5

# 1 Engim Srl

## 1.1 L'azienda

Engim è una società che si occupa di creare soluzioni tecnologiche in ambito ICT, telecomunicazioni, sistemi di gestione e mobilità. Da oltre 10 anni operano nel mercato della tracciabilità di flotte e attività e della sicurezza dei lavoratori in solitario.

ServizioGPS è il noleggio di tracker gps per veicoli lavoratori. Una sostanziosa parte dei clienti sono comuni, che tramite i prodotti Engim, tracciano il percorso delle macchine pulisci neve e spargi sale. I tracker possono essere prodotti fisici oppure un app per smartphone. A loro volta i prodotti fisici si dividono in tracker fissi o mobili. Il servizio include anche un gestionale per poter visualizzare, modificare o archiviare i propri dati.

Twicetouch è noleggio di dispositivi di sicurezza individuale. Il prodotto tutela i lavoratori in solitario mandando una segnalazione in caso di emergenza. Esistono due tipi di rilevazione:

- caduta: l'accelerometro del dispositivo rileva un urto pericoloso
- assenza: il dispositivo non si è mosso per un lasso prolungato di tempo, quindi si presume che il lavoratore possa essere incosciente

Similmente a servizioGPS è possibile noleggiare un dispositivo fisico (badge) oppure l'app per android. In entrambi i casi è possibile impostare i numeri in caso di emergenza, che riceveranno una chiamata e un messaggio SMS.

## 1.2 Infrastruttura

Le tecnologie usate per servizioGPS sono le seguenti:

- Ruby on rails full stack
- Mariadb e Redis come database
- Python come back end di supporto
- Java per il prodotti app

Ruby on rails sviluppa tutta la parte grafica e gran parte del backend di servizioGPS. Il fetcher invece, si dedica esclusivamente alla elaborazione di coordinate gps, dando supporto soprattutto nei momenti di punta. Allo stesso scopo il servizio si appoggia su molteplici server. Uno di questi server è Amazon Web Services, che ha permesso l'introduzione dei microservizi.

## 1.3 Microservizi

Un servizio è un processo che esegue specifiche operazioni autonomamente, risponde a eventi oppure rimane in attesa di una richiesta. Nel caso in cui queste operazioni vengano svolte continuamente, il servizio è estremamente vantaggioso. Non è però necessario che il servizio sia sempre in esecuzione se viene usato in maniera discontinua o per brevi periodi di tempo. I microservizi coprono questo ruolo, hanno le stesse caratteristiche di un servizio ma eseguono solo su richiesta.

L'ambiente di esecuzione è completamente gestito da AWS, l'unica necessità per creare un microservizio è caricare il proprio codice. L'ambiente viene

creato al momento della richiesta, esegue il codice e cessa di esistere. Naturalmente non esiste uno spazio di archiviazione, quindi nel caso sia necessario salvare un file deve essere caricato su un servizio di cloud computing come S3. Un limite dei microservizi è il fatto che si è costretti a una scelta di tecnologie compatibili con l'infrastruttura.

Tenendo a mente le caratteristiche della tecnologia, un caso d'uso ottimo per i microservizi è un API che viene richiamato per brevi periodi oppure in maniera arbitraria.

## **1.4 Sistema automatico di generazione di report**

Le informazioni di ogni cliente vengono archiviate al termine di ogni anno. Nel caso in cui il cliente abbia necessità o interesse a salvare i dati del loro lavoro possono stampare o scaricare una serie di informazioni dal sito.

Le più importanti stampe sono i punti che arrivano e le loro specifiche. Al momento questo è gestito internamente tramite una gemma di ruby.

Il sistema attuale crea un istanza di chrome, l'istanza contiene un HTML che si desidera convertire in pdf e infine avviene il parsing di del documento. Questo crea una serie di gravi problemi:

- La necessità di avviare un istanza di chrome e il parsing di un HTML è estremamente costoso dal punto di vista delle risorse
- Il parsing di un HTML è anche estremamente costoso in termini di tempo, aggravato dalle lunghe query dovute alla grande mole di dati
- Usare HTML rende semplice l'organizzazione del documento, ma si diventa dipendenti dalla gemma per il corretto parsing

La natura del nostro problema rende molto facile la scelta di un microservizio. L'operazione è ripetitiva, ben definita e usata per brevi periodi. Altri vantaggi importanti sono il risparmio di risorse del server, che evita di gravare sulle operazioni più critiche, e la possibilità di usare il microservizio per qualsiasi altro prodotto Engim.

## **2 Requisiti del progetto**

### **2.1 Descrizione**

Il progetto deve essere una lambda su AWS. La lambda deve essere in grado di ricevere un json, autenticare tramite token e whitelist, salvare il progetto su un bucket S3 e ritornare il link di download al pdf. Se qualcosa va storto deve ritornare 400 se è un problema di input oppure 500 se è un problema di connessione al bucket. Il programma inoltre deve essere modulare per permettere di espandere in futuro le stampe ad altri servizi oltre che servizioGPS e creare altri tipi di stampa come kml e exel.

### **2.2 Scelte del linguaggio**

Le lambda su AWS hanno il supporto per i linguaggi più popolari del momento, in particolare ho preso in considerazione python, javascript e ruby. La libreria che faceva di più al mio caso era pdfKit di javascript, che è un port di pdfkit libreria di php per scrivere pdf senza dover far parsing di altri file. Ruby aveva più libreria con parsing intermedi e python aveva libreria non mantenute.

## **3 Sviluppo**

### **3.1 Workflow**

Il mio senior ha deciso di fare un implementazione a grossi step, ogni volta facendo un piccolo pezzo per poi controllare che tutto funzioni correttamente. Ogni issue è uno step e a ogni step il progetto viene testato sia manualmente su aws tramite simulazioni di richiesta sia con test automatizzati tramite mocha.

### **3.2 Creazione di un file e salvataggio su S3**

La lambda è una funzione asincrona che viene eseguita ogni volta che riceve una chiamata https. Bisogna prima di tutto creare una connessione con il bucket di s3, una volta creata questa connessione si può salvare un file passando il filepath oppure uno stream di data e avremo salvato il file su S3

### **3.3 Stampa di un file pdf da un json**

Le operazioni utili al mio scopo text per scrivere una stringa, stroke per disegnare una linea e infine è possibile manipolare colori e font. Inizialmente avevo deciso di programmare in maniera funzionale perché risultava una logica più semplice essendo un breve script, con l'aumentare della complessità e con il modo di esportare di javascript ho pensato che una classe fosse più adeguata. Quindi c'è un costruttore che definisce la grandezza del documento, questo mantiene la relazione di un foglio A4 (circa 1.41) in modo che stampa e visualizzazione sono sempre molto comodi. Dopo l'inizializzazione



si inizia la scrittura dei blocchi. Ogni json potrebbe avere un header, body e table. Se uno dei 3 manca, non verrà scritto e gli altri si organizzeranno di conseguenza. Ogni metodo di scrittura controlla che non ci siano dati mancanti che possano nuocere alla stampa come ad esempio se manca un titolo non è una stampa valida, e ritornano 400. Se tutto va bene e la stampa va a buon fine, è possibile mandare l'oggetto con get document. Ogni possibile caso è testato con mocha, la connessione con AWS è mochata tramite aws-sdk e gran parte dei dati è generata casualmente tramite faker. Se tutto va a buon fine il programma ritorna 200 e il link del download.