

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



**Analisi, progettazione e sviluppo del backend
di un'applicazione web per la gestione di
eventi**

Tesi di laurea

Relatore

Prof.Davide Bresolin

Laureando

Alberto Lazari

ANNO ACCADEMICO 2021-2022

Sommario

La tesi descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, presso la sede di Treviso di Moku S.r.l., il cui obiettivo era la reimplementazione del backend di una piattaforma di gestione di eventi, sfruttando gli strumenti tipicamente utilizzati nei progetti dell'azienda.

In particolare, i seguenti capitoli tratteranno del contesto lavorativo dell'azienda, dell'analisi svolta sullo stato della piattaforma ad inizio stage, della progettazione e successiva implementazione iniziale del nuovo backend, focalizzando l'attenzione sulle scelte stilistiche e architetturelle perseguite.

Ringraziamenti

Voglio ringraziare il Prof. Davide Bresolin, per l'interesse, il supporto e l'aiuto fornito durante il periodo di stage e di stesura di questa tesi.

Ringrazio i miei genitori e tutti i miei famigliari per il supporto e l'affetto che mi hanno donato durante questi anni di studio.

Ringrazio i colleghi di Moku che mi hanno accolto calorosamente tra loro durante la mia esperienza di stage, in particolare Riccardo e Nicolò, per avermi sempre fornito l'aiuto che cercavo durante il mio lavoro.

Ringrazio la Comunità Capi del Mestre 2, per avermi accompagnato fin qui, infondendo in me una maturità e una consapevolezza che mi hanno permesso di raggiungere questo traguardo.

Ringrazio i miei colleghi studenti, stagisti e gli amici dei gruppi di progetto, con cui ho condiviso le difficoltà e le soddisfazioni di quest'anno.

Infine ringrazio i miei amici, che mi sono sempre stati vicini e con cui ho condiviso esperienze indimenticabili.

Padova, Luglio 2022

Alberto Lazari

Indice

1	L'azienda	1
1.1	Descrizione generale	1
1.2	Modello di sviluppo	1
2	Descrizione dello stage	3
2.1	Introduzione al progetto	3
2.2	Requisiti	3
2.3	Pianificazione	3
2.4	Tecnologie utilizzate	3
3	Progettazione	5
3.1	Analisi e refactor dei modelli	5
3.2	API	5
3.3	Gestione dei permessi	5
4	Codifica	7
4.1	Modelli	7
4.1.1	Migrazioni del database	7
4.1.2	Associazioni a modelli e file	7
4.1.3	Validazioni	7
4.2	Controller	7
4.2.1	APIController	7
4.2.2	Implementazione delle action	7
4.3	Gestione dei permessi	7
4.4	Test di unità	7
5	Conclusioni	9
5.1	Raggiungimento dei requisiti	9
5.2	Valutazione personale	9
	Bibliografia	11

Elenco delle figure

Elenco delle tabelle

Capitolo 1

L'azienda

1.1 Descrizione generale



Moku S.r.l. è una start-up nata nel 2013 all'interno di un progetto supportato da H-Farm. Dopo aver abbandonato il progetto si è dedicata allo sviluppo software su commissione e consulenza, per poi allontanarsi definitivamente da H-Farm a settembre 2021, muovendo la sua sede dalla *farm* a Roncade a quella attuale di Treviso.

L'azienda è in continua espansione e conta circa 20 dipendenti, la maggior parte con età inferiore ai 30 anni. Questo contribuisce a mantenere l'ambiente di lavoro stimolante e accogliente per tutti, permettendo di includere i diversi studenti che ogni anno svolgono il loro stage presso l'azienda. Per questi vengono attivate proposte di progetto per i ruoli di sviluppatore backend, sviluppatore frontend e sviluppatore mobile, all'interno di team interni che lavorano a progetti reali commissionati all'azienda.

1.2 Modello di sviluppo

I progetti di Moku seguono un modello di sviluppo *agile*, con metodologie basate su *Scrum*¹, un framework pensato per team di sviluppo software di piccole dimensioni (non più di dieci membri). La metodologia adottata prevede le seguenti caratteristiche:

- il lavoro viene suddiviso in *sprint*, intervalli temporali della durata di due settimane;

¹Jeff Sutherland Ken Schwaber. *The 2020 Scrum Guide*. URL: <https://scrumguides.org/scrum-guide.html>.

- ogni *sprint* è preceduto da una riunione di pianificazione degli obiettivi, espressi sotto forma di *user stories*, che esprimono le funzionalità del software da implementare, scritte in un linguaggio naturale dal punto di vista dell'utente;
- uno *sprint* termina con la relativa *sprint review*, una riunione con il cliente che ha l'obiettivo di mostrare l'incremento prodotto nel software, attraverso dimostrazioni del funzionamento del software stesso;
- il team a cui viene affidato lo sviluppo di un progetto è composto da diverse figure professionali, tra cui un *project manager*, il cui compito è coordinare il lavoro tra gli altri componenti e definire le *user stories* da inserire nel *backlog* degli *sprint*, oltre a gestire la pianificazione dello *sprint* stesso;
- all'inizio di ogni giornata lavorativa, il team si riunisce nello *stand-up meeting*, una riunione della durata di circa 15 minuti, per condividere lo stato del lavoro di ogni componente, descrivere gli obiettivi del giorno e far emergere eventuali problemi sorti durante lo sviluppo.

La metodologia adottata permette di avere una comunicazione regolare ed efficace tra il team di sviluppo e il cliente, che porta a una definizione più semplice e precisa dei requisiti che il prodotto deve rispettare e a una comprensione immediata e chiara dell'avanzamento dello sviluppo da parte del cliente, attraverso le dimostrazioni pratiche effettuate nel contesto delle *sprint review*.

Capitolo 2

Descrizione dello stage

2.1 Introduzione al progetto

Storia del progetto prima del mio arrivo, azienda che ha commissionato il progetto, descrizione dello scopo della piattaforma e del suo funzionamento, motivazioni alla base della scelta di riscrittura del backend.

2.2 Requisiti

Requisiti obbligatori, desiderabili e opzionali previsti.

2.3 Pianificazione

Divisione settimanale del lavoro dal piano di lavoro (con la somma delle ore corretta: 320 invece di 300).

2.4 Tecnologie utilizzate

Descrizione della configurazione del framework Ruby on Rails utilizzata: librerie utilizzate, Postgres, AWS, API REST .

Capitolo 3

Progettazione

3.1 Analisi e refactor dei modelli

Decisioni significative prese durante l'attività di refactor dei modelli, diagramma ER del nuovo backend.

3.2 API

Forma generale dei controller (operazioni CRUD) con casi particolari, autenticazione.

3.3 Gestione dei permessi

Permessi per le categorie di utenti per ogni controller.

Capitolo 4

Codifica

4.1 Modelli

4.1.1 Migrazioni del database

Comando rails generate e migrazione prodotta.

4.1.2 Associazioni a modelli e file

Associazioni di Active Record e Active Storage.

4.1.3 Validazioni

Validazioni sugli attributi del modello e le associazioni.

4.2 Controller

4.2.1 APIController

Descrizione dei metodi di utilità ereditati dai controller dell'API.

4.2.2 Implementazione delle action

Descrizione ed esempio di action tipiche dei controller.

4.3 Gestione dei permessi

Funzionamento e uso della gemma “Pundit” per la gestione dei permessi relativi agli endpoint dell'API, scope e metodi relativi alle action, esempio di gestione della gerarchia che andrà rivisto

4.4 Test di unità

Descrizione della gemma “RSpec”, che fornisce strumenti per lo sviluppo guidato dal comportamento (behaviour-driven development), esempi di modelli testati.

Capitolo 5

Conclusioni

5.1 Raggiungimento dei requisiti

Tabella con stato di completamento dei requisiti, con commento (dove necessario)

5.2 Valutazione personale

Messe alla prova le competenze fornite dal corso di laurea, verificata l'efficacia dei corsi e dei progetti svolti, imparato un nuovo linguaggio e framework con filosofia di sviluppo a me nuova, scoperto ambiente lavorativo aziendale con i ruoli e le dinamiche interne.

Bibliografia

Siti web consultati

Ken Schwaber, Jeff Sutherland. *The 2020 Scrum Guide*. URL: <https://scrumguides.org/scrum-guide.html>.