Università degli Studi di Modena e Reggio Emilia

Dipartimento di Ingegneria "Enzo Ferrari"

Corso di Laurea in Ingegneria Informatica

Progettazione e sviluppo di una piattaforma riutilizzabile in contesto aziendale

Relatore:
Prof. Francesco Guerra

Presentata da:
Matteo Sirri

Anno Accademico 2020-2021

Abstract

Indice

Capitolo 1

Introduzione

1.1 Obiettivo

1.2 Campo di applicazione

1.3 Panoramica

Capitolo 2

Descrizione generale

2.1 Inquadramento

2.2 Macrofunzionalità del sistema

2.3 Caratteristiche degli utenti

2.4 Vincoli generali

2.5 Analisi future

Capitolo 3

Tecnologie

3.1 Implementazione

In questa sezione verranno descritti gli strumenti utilizzati per implementare i componenti che permettono alla piattaforma di erogare i propri servizi.

3.1.1 Ambiente di sviluppo

La scelta dell'ambiente di sviluppo, ovvero l'insieme di strumenti e tecnologie che permettono di sviluppare codice sorgente, è stata imposta dal know-how aziendale.

Node.js

Node.js è un ambiente runtime JavaScript open-source e multipiattaforma.

Le caratteristiche fondamentali sono: l'esecuzione dell'engine V8, sviluppato da Google, che permette di compilare ed eseguire codice JavaScript al di fuori di browser web, l'uso di un insieme di primitive I/O asincrone di tipo non bloccante e l'esecuzione di applicazioni su un solo processo, senza generazione di nuovi thread per ogni richiesta. Pertanto quando si deve eseguire una operazione I/O, come una richiesta ad un web server, Node.js non blocca il thread, mettendo in attesa la CPU, ma, al contrario, la lascia libera di portare avanti altri compiti e si occuperà di ripristinare l'operazione non appena arriverà una risposta.

Grazie a queste peculiarità è possibile realizzare applicazioni performanti in grado di gestire connessioni concorrenti con un singolo server, senza introdurre la complessità logica legata alla gestione della concorrenza fra thread.

In questo ambiente è poi possibile utilizzare lo standard ECMAScript nelle sue varie versioni in modo flessibile in quanto è possibile modificare il set di funzionalità abilitate, potendo così adattarsi al meglio nei vari contesti di utilizzo.

Infine, Node.js permette anche di aumentare la produttività di un team di sviluppo perchè fornisce agli sviluppatori front-end, che conoscono il linguaggio JavaScript, la possibilità di sviluppare codice *server-side*; senza dover imparare un linguaggio del tutto nuovo.

Grazie alle sue caratteristiche Node.js risulta essere un'ottimo strumento per lo sviluppo di servizi web.

3.1.2 Linguaggio

Typescript

3.1.3 Framework

Nest.js

3.1.4 Testing

Jest

3.2 Gestione dati

3.2.1 Database

MongoDB

3.3 Servizi esterni

3.3.1 AWS SES

3.4 Protocolli comunicazione

3.4.1 HTTP

3.4.2 RabbitMQ

3.5 Sicurezza

3.5.1 Autenticazione

3.5.2 Autorizzazione

JWT

OAuth2.0

3.6 Gestione codice condiviso

3.6.1 Git

3.6.2 Npm

Capitolo 4

Architettura

4.1 Descrizione generale

4.2 API Server

4.2.1 Descrizione generale

4.2.2 Principi di design

4.2.3 Auth module

4.2.4 Demo module

4.2.5 User module

4.2.6 Mail module

4.3 Mailer microservice

4.3.1 Descrizione generale

4.3.2 Principi di design

4.3.3 Template Service

4.3.4 Transport Service

4.4 Database Server

Capitolo 5

Distribuzione

5.1 Descrizione generale

5.2 CI/CD pipeline

5.2.1 Descrizione generale

5.2.2 Motivazioni

5.2.3 Integrazione continua

5.2.4 Distribuzione continua

5.2.5 Deployment continuo

Capitolo 6

Conclusioni

6.1 Valutazioni complessive

6.2 Sviluppi futuri

Bibliografia