UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE E INFORMATICHE
Corso di Laurea in Informatica

Tesi di Laurea

REALIZZAZIONE DI UN PLUGIN PER L'API GATEWAY KONG PER L'INTEGRAZIONE CON UN MICROSERVIZIO PER IL CONTROLLO DEGLI ACCESSI TRAMITE TOKEN

Candidato:
Valerio Desiati

Relatore:
Prof. Roberto Alfieri

Anno Accademico 2022/2023

Dedica

Indice

Elenco delle figure

Elenco degli algoritmi

Elenco delle tabelle

Introduzione

L'introduzione deve contenere un riassunto del lavoro di Tesi. In particolare bisogna esprimere chiaramente e molto sinteticamente: contesto dello studio, motivazioni, contributo e conclusioni. Bisogna quindi fare un sommario dello studio ad alto livello, fornendo le intuizioni senza ricadere in dettagli tecnici. Anche lo stile dovrebbe essere più discorsivo rispetto alle parti tecniche della tesi.

Capitolo 1

Descrizione dello stage

In questo capitolo saranno esposte brevemente tutte le caratteristiche e le dinamiche dello stage svolto, che saranno approfondite nei capitoli successivi.

1.1 Azienda

AESYS S.r.l. è un'azienda fondata nel 2013 a Pescara (PE) che si occupa di sviluppo software e consulting con sedi a Pescara e Torino. Ad oggi AESYS conta più di 240 dipendenti.
Nel corso degli anni l'azienda ha vissuto una grande crescita e ora dispone di figure specializzate in molteplici campi.

1.1.1 Servizi offerti

AESYS è attiva in molti campi dell'ambito dell'*Information Technology* fornendo servizi per piattaforme Web e Mobile, in ambito DevOps, Cloud, Machine Learning e sviluppo UX/UI.

1.1.2 Organizzazione

AESYS è suddivisa in *Business Unit*, ognuna per campo di sviluppo.
In particolare, nella realizzazione di questo progetto sono state coinvolte due Business Unit: la RED, che si focalizza su tutte le tecnologie legate al linguaggio Java e alla JVM e la ORANGE, unità in ambito DevOps che fornisce supporto per manutenzione e monitoraggio dei sistemi su grandi infrastrutture.

1.1.3 Metodologie aziendali

AESYS adotta la metodologia di sviluppo Agile da diverso tempo, applicandone i principi nel quotidiano.
La metodologia di sviluppo Agile si basa su dei valori fondamentali:

- Il personale e le interazioni sono più importanti dei processi e degli strumenti.
- È meglio avere un software funzionante che una documentazione esaustiva.
- La collaborazione con il cliente è più importante della stipula di un contratto.
- Essere pronti al cambiamento.

Strumenti di supporto

In AESYS lo strumento più utilizzato per comunicare è Microsoft Teams.
Il software in questione permette di creare al proprio interno una vera e propria gerarchia aziendale, programmare meeting, avere agende condivise, instant messaging e tante altre risorse utili per lavorare in team. All'interno dell'azienda viene utilizzato *Git* come sistema di versionamento del software (argomento approfondito nel capitolo *Tecnologie utilizzate*).

Figura 1.1: Logo di AESYS S.r.l.

1.2 Progetto di stage

Il progetto di stage prevede la realizzazione di un plugin per *Kong Gateway* scritto con il linguaggio *Lua*.
Il plugin prevede l'integrazione del gateway con un microservizio scritto in Java per verificare l'avvenuto acquisto di un modulo applicativo da parte di un utente identificato tramite token.
Il progetto comprende la realizzazione del microservizio e l'ottimizzazione del plugin tramite l'utilizzo di una cache locale a Kong.

Ovviamente è stato necessario acquisire delle competenze di base prima della realizzazione del progetto, come ad esempio: Acquisire una conoscenza sufficiente del sistema di Version Control *Git*.

Le competenze necessarie per lo svolgimento del progetto sono:

- Conoscenza della metodologia di sviluppo Agile (acquisita durante lo stage).
- Conoscenza del sistema di Version Control *Git* (acquisita durante lo stage).
- Conoscenza del linguaggio Java e della JVM (Java Virtual Machine) per la realizzazione del microservizio (acquisita durante gli studi accademici).
- Conoscenza del framework *Spring* da utilizzare nello sviluppo del microservizio.
- Conoscenza della struttura e del funzionamento di Kong Gateway (acquisita durante lo stage).
- Conoscenza del linguaggio Lua per lo sviluppo del plugin (acquisita durante lo stage).
- Conoscenza degli strumenti per effettuare testing sul prodotto finito (acquisita durante lo stage).

1.2.1 Ripartizione del lavoro svolto

Lo stage aziendale ha avuto una durata di 225 ore, come previsto dal piano di studio Universitario per la Facoltà di Informatica. All'interno dell'azienda le ore sono state suddivise in cinque settimane lavorative full time, dal lunedì al venerdì dalle 09:00 alle 18:00.
Di seguito due tabelle che riassumono la ripartizione settimanale e oraria dello sviluppo del progetto.

Settimana	Attività svolte
Prima settimana	Esposizione delle specifiche del progetto. Acquisizione delle competenze preliminari necessarie quali *Git* e metodologia di sviluppo *Agile*.
Seconda settimana	Studio del framework *Spring*. Sviluppo del microservizio.
Terza settimana	Sviluppo del microservizio.
Quarta settimana	Testing del microservizio. Studio del funzionamento di Kong Gateway. Configurazione Kong Gateway.
Quinta settimana	Studio delle specifiche del plugin. Studio della sintassi e della semantica del linguaggio *Lua*. Sviluppo del plugin per Kong Gateway. Testing del prodotto finito.

—c—c—c—

Attività svolte
Esposizione delle specifiche del progetto
Acquisizione delle competenze preliminari necessarie quali *Git* e metodologia d
Studio del framework *Spring*
Sviluppo del microservizio
Testing del microservizio
Studio del funzionamento di Kong Gateway
Configurazione Kong gateway
Studio delle specifiche del plugin
Studio della sintassi e della semantica del linguaggio Lua
Sviluppo plugin per Kong gateway
Testing del prodotto finito

1.3 Obiettivi

Capitolo 2

Tecnologie utilizzate

In questo capitolo saranno descritte le tecnologie utilizzate nella realizzazione del progetto.

2.1 Git

Git è un DVCS (Distributed Version Control Systems) gratuito, open source e distribuito, utilizzabile da riga di comando, che consente di effettuare il controllo versione per un progetto.
Data la sua natura "distribuita" *Git* è basato su flussi di lavoro simultanei con i quali diversi sviluppatori possono collaborare ad un progetto, ognuno con il proprio workflow.

Figura 2.1: Logo di Git

2.2 Java

Java è un linguaggio di programmazione ad alto livello orientato agli oggetti. Il suo scopo è quello di essere multipiattaforma, tutte le piattaforme che supportano il linguaggio devono essere in grado di eseguire un codice *Java* compilato senza effettuare nuovamente la compilazione.
Compilando un codice *Java* si ottiene un file *Java bytecode* (con estensione `.class`) che sarà eseguito sulla *JVM* (Java Virtual Machine).

È proprio la *JVM* ad essere utilizzabile sulla maggior parte delle piattaforme.

Figura 2.2: Logo di Java

2.3 Eclipse

Eclipse è un IDE (Integrated Development Environment) per lo sviluppo software realizzato in Java.
Eclipse unisce in un'unica interfaccia grafica:

- Scrittura del codice sorgente
- Compilazione
- Debugging

Figura 2.3: Logo di Eclipse

2.4 Framework Spring

Spring è un framework open source per lo sviluppo di applicazioni in Java.
Spring è strutturato in diversi moduli ma consente l'utilizzo solo di quelli di cui effettivamente si necessita.
Nello specifico, per la realizzazione del progetto sono stati utilizzati i moduli descritti in seguito.

2.4.1 Spring Boot

L'utilizzo di questo modulo consente di creare applicazioni Java standalone, pronte all'esecuzione.
Spring Boot consente di scegliere quale tool utilizzare per effettuare la build, in questo progetto è stato utilizzato Maven.
Alla base di ogni build con *Spring Boot* e *Maven* c'è il file *pom.xml* (acronimo di Project Object Model) in cui sono descritte tutte le impostazioni e le dipendenze necessarie alla build in un linguaggio *simil-XML*.

2.4.2 Spring Data

Spring Data fornisce un modello di programmazione per l'accesso ai dati indipendentemente dal tipo di database utilizzato.
Nello specifico, per la realizzazione del progetto è stata utilizzata la specifica di *Spring Data* chiamata *JPA* (Java Persistence API) per:

- Gestione del database
- Creazione di tabelle
- Esecuzione di query

Figura 2.4: Logo di Spring

2.5 Azure DevOps

Azure DevOps è una piattaforma fornita da Microsoft™che consente di pianificare il lavoro, creare e distribuire applicazioni.
Nello specifico, per la realizzazione del progetto sono state utilizzate le applicazioni descritte in seguito:

- **Azure Repos** per la creazione e la gestione di repository *Git* per il controllo e il versionamento del codice sorgente.
- **Azure Pipelines** per l'automatizzazione di build e deploy dell'intero progetto.
- **Azure Artifacts** per la condivisione degli artefatti *Maven*.

Figura 2.5: Logo di Azure

2.6 Docker

Docker è un progetto open source per la creazione di container portabili e multipiattaforma.
Docker utilizza il kernel Linux per isolare i processi in modo da poterli eseguire in maniera indipendente.
Ogni container è basato su un'immagine, solitamente un intero Sistema Operativo, a scelta tra quelle fornite all'interno di Docker Hub (raccolta ufficiale di tutte le immagini disponibili) o un'immagine "custom", realizzata appositamente dal singolo sviluppatore per un determinato scopo.
Grazie all'organizzazione in container si ha un alto livello di sicurezza, come se i sistemi in esecuzione fossero fisicamente separati.
Nella realizzazione del progetto è stata utilizzata l'immagine ufficiale di *Kong Gateway* (argomento approfondito nel paragrafo successivo) per la creazione del container.

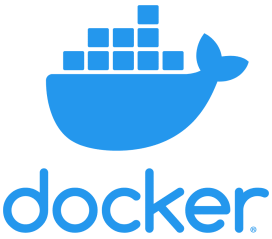

Figura 2.6: Logo di Docker

2.7 Kong Gateway

Un API gateway è uno strumento che si interpone tra un client e un back end per la gestione delle API (Application Programming Interface) che si comporta come un proxy inverso accettando tutte le richieste indirizzate alle API gestite, consentendo di configurare *services* e *routes*.
Kong Gateway è un API gateway cloud-native che fornisce tutte le caratte-

ristiche descritte sopra e, inoltre, consente l'utilizzo di plugin.
Una volta installato è possibile configurarlo accedendo alle seguenti pagine:

- **Kong Manager**, porta 8000, consente di utilizzare un'interfaccia grafica per configurare *services*, *routes* e *plugins*.
- **Pagina delle configurazioni**, porta 8002, raccoglie tutte le configurazioni del gateway in formato JSON.

Figura 2.7: Logo di Kong Gateway

2.7.1 Service

Un *service* in Kong Gateway è un'astrazione di tutti i servizi upstream custom che si aggiungono alla configurazione. Con *servizio upstream custom* si intende un microservizio custom che prende dati dalla richiesta inoltrata al gateway e ne restituisce altri al gateway stesso, che si occuperà di comunicarli al client.
Solitamente ad ogni *service* è associata una o più *routes*.

2.7.2 Route

Una *route* è una regola definita per indirizzare correttamente le richieste del client.
L'associazione di una (o più) route ad un servizio consente di realizzare un meccanismo di routing molto potente, dato che è possibile configurare nel dettaglio il percorso che si vuole realizzare (protocolli da utilizzare, livello di sicurezza ecc.).

2.7.3 Plugin

Un *plugin* è un'entità che sarà eseguita durante tutto il ciclo di vita di una richiesta o risposta HTTP/S (HyperText Trasfer Protocol / Secure).
È il modo in cui Kong Gateway fornisce la possibilità di ottenere funzionalità

aggiuntive per un *service* o una *route*.
I plugin possono configurabili possono essere sia proprietari (attivabili da Kong Manager) sia custom. Per la realizzazione di un plugin custom si ha la possibilità di scegliere tra vari linguaggi di programmazione per lo sviluppo quali Go, Python, JavaScript e Lua (linguaggio utilizzato per lo sviluppo del plugin custom utilizzato nel progetto).

2.7.4 Consumer

Un *consumer* in Kong Gateway può essere inteso come un utente di uno specifico servizio e può essere identificato tramite un `id` univoco.

2.8 PostgreSQL

PostgreSQL è un DBMS (Database Management System) open source relazionale a oggetti che supporta la gran parte delle istruzioni del linguaggio SQL standard alle quali aggiunge diverse feature quali:

- Query complesse
- Foreign keys
- Triggers
- Views aggiornabili
- Integrità dei dati nelle transazioni
- Controllo concorrente del versionamento

Inoltre fornisce la possibilità di aggiungere tipi di dato, funzioni, operatori ecc.

Figura 2.8: Logo di PostgreSQL

2.9 Lua

Lua è un linguaggio di scripting open source che combina la sintassi procedurale a costrutti di dati basati su array associativi.
È un linguaggio tipizzato dinamicamente, viene eseguito interpretando un bytecode e gestisce la memoria in modo automatico tramite un *garbage collector*.
È stato scelto per la realizzazione del plugin per le sue caratteristiche quali:

- Velocità
- Portabilità
- Leggerezza
- Embed-easy

Figura 2.9: Logo di Lua

2.10 Postman

Postman è una piattaforma API per la creazione, sviluppo e testing di APIs. Nello sviluppo del progetto è stato utilizzato per la fase di testing dato che permette di effettuare delle richieste HTTP/S, offrendo la possibilità di configurare il body della stessa e di ricevere la risposta.

Figura 2.10: Logo di Postman

Capitolo 3

Analisi dei requisiti

3.1 Casi d'uso

Capitolo 4

Progettazione e sviluppo

4.1 Introduzione ai microservizi

4.2 Introduzione a Kong Gateway

4.3 Architettura del progetto

4.4 Architettura del microservizio

4.4.1 Sviluppo del microservizio

4.5 Architettura del plugin

4.5.1 Sviluppo del plugin

Capitolo 5

Testing

Conclusione

Appendice A

Appendice