

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

ED ELETTRICA E MATEMATICA APPLICATA (DIEM)

CLASSE DELLE LAUREE MAGISTRALI IN INGEGNERIA

INFORMATICA (LM-32)



SOFTWARE ENGINEERING (A-H)

ARCHITETTURA DEL SOFTWARE

Gruppo10:

Alessandra Boccuzzi

Alessia Di Francesco

Marco Della Rocca

Rosario Frontino

Matricole:

0622701567

0622701573

0622701484

Anno Accademico 2020– 2021

Il sistema sviluppato implementa un'architettura di rete di tipo Client-Server.

I Client, rappresentati dai Web browser degli utilizzatori finali, interagiscono con il Server tramite una Web Application, operando indirettamente su dati reali memorizzati in un database dedicato.

L'applicazione implementa il pattern architetturale Model-View-Controller (MVC), separato in Frontend (View) e Backend (Controller).

L'implementazione del Frontend ha richiesto l'utilizzo del linguaggio di markup HTML, il linguaggio di stile CSS e i linguaggi di scripting JAVASCRIPT e PHP.

Il Backend implementa una REST API in Python sfruttando le funzionalità fornite dal Micro Web Framework «Flask» e dall'Object Relational Mapper (ORM) SQLAlchemy.

La base dati, infine, è stata implementata con il popolare Relational DataBase Management System (RDBMS) PostgreSQL.

Il Server in questione è una Droplet Digital Ocean, ovvero un'istanza in Cloud sulla quale è installato il sistema operativo Ubuntu.

Accedendo all'indirizzo web collegato al Server il Web Browser esegue la risoluzione del DNS sfruttando i servizi offerti da No-IP, un fornitore pubblico gratuito, ottenendo il reale indirizzo IP del Server.

A questo punto invia una richiesta HTTP al Web Server sulla porta 80 che viene ricevuta dal processo Nginx, il quale analizza l'indirizzo della richiesta per comprendere la locazione del file desiderato.

Individuato il file in questione, Nginx comunica con PHP-FPM, un Process Manager, che si occupa della gestione di processi PHP per l'interpretazione e la risoluzione del codice di Frontend.

Quando necessario PHP invia richieste, costruite tramite la libreria cURL, ad un processo Python.

Tale processo, gestito tramite il Process Manager PM2, provvede alla reindirizzazione delle richieste ricevute agli endpoint destinatari ed esegue

scambi di informazioni con il Database Relazionale sfruttando l'ORM per la generazione del codice SQL .

Alla fine del processo di interpretazione, PHP avrà generato un output che consiste in codice costituito esclusivamente da HTML, CSS e Javascript che sarà nuovamente inoltrato, mediante PHP-FPM ed Nginx al Web browser sotto forma di risposta HTTP.

L'accesso al server è garantito dal protocollo SSH configurato senza password, permettendo dunque di loggare solo se nel file apposito sono presenti le chiavi pubbliche RSA degli sviluppatori addetti alla manutenzione. Per questioni di sicurezza è stato inoltre utilizzato un utente linux separato al quale sono garantiti permessi limitati ad un ristretto set di file e cartelle presenti sul Server.

Il codice del Backend implementa test automatici eseguiti su di un'istanza di Database separata, configurabile indicandone l'indirizzo in una specifica variabile d'ambiente.

La codebase è hostata su Github ed è sincronizzata con il Server tramite una Pipeline di Continuous Integration avviabile grazie a script di deploy (in BAT o bash) che automatizzano il processo di pull delle repository.

Il codice è diviso in tre progetti separati: uno per il Backend, uno per il Frontend dell'applicazione di amministrazione ed uno per il Frontend dell'applicazione di pianificazione.

All'avvio dell'applicazione, ad accogliere l'utente, sarà presente una schermata di login che ricaverà informazioni circa il suo ruolo, reindirizzandolo correttamente all'applicazione Admin o Planner.

Gli endpoint sono protetti da autenticazione e quindi dal ruolo dell'utente: per accedere a tali richieste è necessario inviare, nell'Header apposito della richiesta HTTP, il JSON Web Token (JWT) ottenuto in fase di autenticazione con scadenza configurabile tramite variabili d'ambiente.

La revoca dei token (e quindi il logout) è gestita con un set memorizzato in memoria centrale.

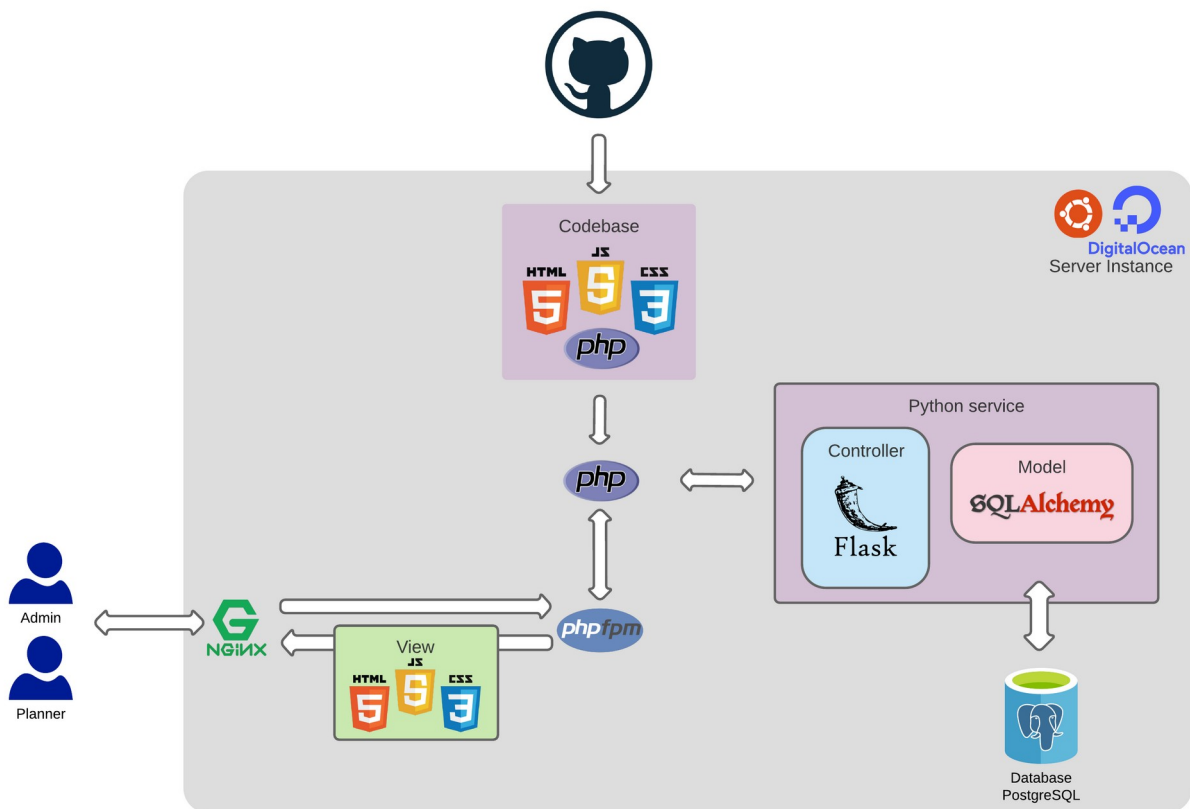


Figura - Struttura dell'Architettura Software