

Corso di Laurea Magistrale in

Ingegneria Informatica

Progetto

Sistemi Distribuiti e Cloud Computing

|  |  |  |
| --- | --- | --- |
| **Professori**  Prof. Domenico Talia  Prof. Loris Belcastro |  | **Candidato**  Pasquale Molinaro Matricola: 235309 |

Anno Accademico 2021/2022

Sommario

[1. Introduzione 3](#_Toc138800572)

[1.1 Traccia 3](#_Toc138800573)

[2. Analisi dei Requisiti 4](#_Toc138800574)

[2.1 Requisiti Funzionali 4](#_Toc138800575)

[2.2 Requisiti non Funzionali 4](#_Toc138800576)

[3. Progettazione e Sviluppo 5](#_Toc138800577)

[3.1 Architettura 5](#_Toc138800578)

[3.2 Database – PostgreSQL 7](#_Toc138800579)

[3.3 Backend – Spring 9](#_Toc138800580)

[3.3.1 DAO Layer 10](#_Toc138800581)

[1.1.1 Service Layer 12](#_Toc138800582)

[3.3.2 Controller Layer 14](#_Toc138800583)

[3.4 Frontend – Angular 17](#_Toc138800584)

[4. Deploy – Azure Cloud Services 18](#_Toc138800585)

[4.1 Configurazione del servizio Azure 19](#_Toc138800586)

[4.2 Accesso, trasferimento file ed Deploy sulla VM 19](#_Toc138800587)

[5. Risultato Finale 19](#_Toc138800588)

# **Introduzione**

Il progetto si propone di creare un sistema distribuito per la ricerca distribuita di informazioni all’interno di file presenti su più server connessi tra loro. Gli utenti hanno la possibilità di effettuare la ricerca tramite parole chiave per visualizzare i file presenti, mentre gli amministratori del sistema sono in grado di salvare i vari file sui server e, come gli utenti, effettuare la ricerca di quelli già presenti attraverso meccanismi di indicizzazione.

L'accesso a tale sistema avviene attraverso un'interfaccia web che consente di eseguire le operazioni menzionate in precedenza. Inoltre, il sistema deve sfruttare le risorse di calcolo, archiviazione e virtualizzazione offerte da Microsoft Azure.

Questa relazione si concentra inizialmente sull'analisi dei requisiti del sistema. Successivamente, vengono descritte le fasi che hanno portato allo sviluppo della piattaforma web, le scelte di progettazione principali, l'architettura del sistema e i servizi offerti agli utenti. Infine, vengono analizzate le diverse componenti utilizzate per la distribuzione su Azure.

## **Traccia**

Lo sviluppo segue le indicazioni della seguente traccia:

Progettare e implementare un sistema distribuito per la ricerca distribuita di informazioni all’interno di file presenti su più server connessi tra loro. Il sistema deve consentire:

* Agli utenti di inserire uno o più termini da ricercare e visualizzare i risultati della ricerca.
* Ai gestori di salvare file sui server ed effettuare le ricerche su tutti i file presenti sui server. A questo scopo i gestori potrebbero indicizzare i contenuti presenti nei file chiedi volta in volta vengono inseriti per effettuare le ricerche sui dati già indicizzati.

Per la realizzazione dell'applicazione, lo studente dovrà includere un'analisi dei requisiti funzionali e non funzionali del sistema da realizzare.

# **Analisi dei Requisiti**

Nella fase iniziale del progetto, si è dedicata particolare attenzione all'analisi dei requisiti. Questa fase rappresenta un passaggio di estrema importanza nel processo di sviluppo di un sistema software, in quanto mira a definire le funzionalità che il nuovo prodotto dovrà offrire. Sono stati individuati i requisiti funzionali, che comprendono le specifiche delle azioni che il sistema deve essere in grado di eseguire, quali il caricamento di file da parte di un amministratore e la ricerca di file per gli utenti. Inoltre, sono stati identificati i requisiti non funzionali, che riguardano le prestazioni del sistema, la sicurezza dei dati e l'usabilità dell'interfaccia web. L'analisi dei requisiti ha costituito una solida base per la fase successiva dello sviluppo del sistema.

## **Requisiti Funzionali**

Nel contesto del progetto, sono emersi i seguenti requisiti da soddisfare per il sistema:

* Utilizzare le risorse di calcolo, storage e virtualizzazione fornite da Microsoft Azure.
* Consentire ai gestori di caricare i file nel Cloud.
* Implementare una funzionalità di ricerca dei file indicizzati.
* Fornire un'interfaccia web (Front-end Web) per l'accesso al sistema, consentendo sia il caricamento che la ricerca dei file.

Considerando le specifiche fornite, è possibile identificare ulteriori requisiti funzionali che possono essere considerati per il sistema. Ecco alcuni esempi:

* Sistema di registrazione degli utenti: Permettere agli utenti di registrarsi creando un account personale, fornendo le informazioni necessarie come nome, indirizzo e-mail e password.
* Implementazione di filtri di ricerca avanzati: Oltre alla ricerca dei file basata su parole chiave, il sistema dovrebbe offrire opzioni di ricerca avanzate, come la possibilità di filtrare i risultati in base all'estensione o al titolo dei file.

## **Requisiti non Funzionali**

Nel contesto del progetto, sono stati individuati i seguenti requisiti non funzionali che influenzano il lavoro degli sviluppatori e descrivono le caratteristiche del sistema:

* + Usabilità: Il software dovrà essere facile da utilizzare per gli utenti, garantendo un'interfaccia grafica intuitiva e user-friendly. L'usabilità sarà un fattore determinante per la soddisfazione e l'efficacia dell'interazione con il sistema.
  + Prestazioni: Il sistema dovrà dimostrare efficienza nell'utilizzo delle risorse di calcolo disponibili. Sfruttando al meglio le potenzialità di calcolo e virtualizzazione offerte da Azure, si potranno ottenere prestazioni ottimali, consentendo di gestire il carico di lavoro richiesto in modo efficiente.
  + Affidabilità: Il software dovrà funzionare come previsto e mantenere le sue funzionalità nel tempo. L'utilizzo delle risorse di cloud computing contribuirà a garantire un'alta affidabilità del sistema, fornendo maggiore disponibilità e tolleranza ai guasti.
  + Robustezza: Il sistema dovrà essere in grado di gestire situazioni impreviste o eccezionali senza causare malfunzionamenti critici. Ad esempio, dovrebbe essere in grado di gestire correttamente il caricamento di dati non corretti o inattesi, garantendo la corretta esecuzione delle funzionalità.

# **Progettazione e Sviluppo**

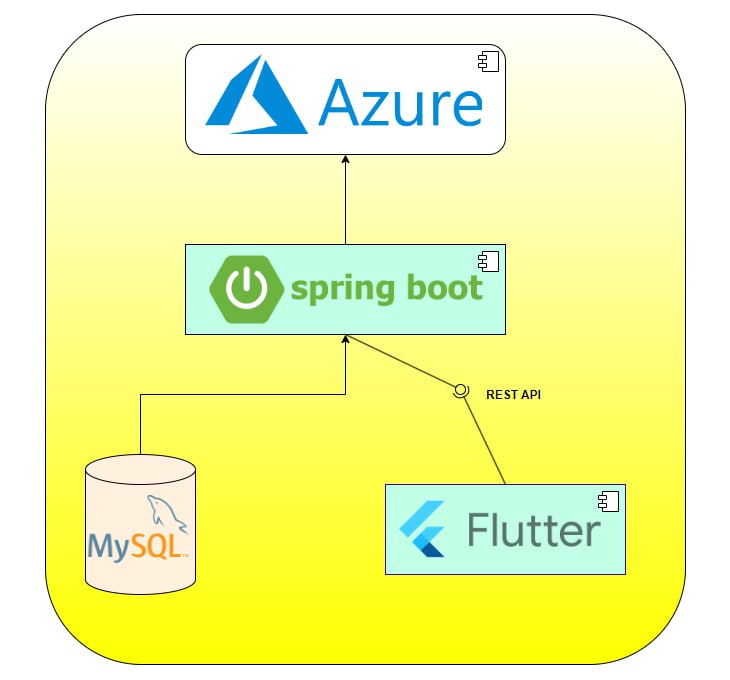
L’applicazione, considerandone lo scopo, è stata denominata **MyFileSharing.**

## **Architettura**

A seconda del tipo di applicazione da sviluppare, possono essere adottate diverse metodologie di sviluppo. Se l'applicazione è relativamente semplice e gestisce dati locali al dispositivo senza la necessità di condividerli con altri utenti in rete, è possibile svilupparla utilizzando metodi di archiviazione dati integrati all'interno dell'app stessa. Tuttavia, spesso si richiede lo sviluppo di applicazioni più complesse, come le applicazioni enterprise, che interagiscono con database esterni al dispositivo e memorizzano dati condivisi tra molti utenti, ad esempio nel caso di un'e-commerce. In questo contesto, è consigliabile separare il front-end dal back-end per gestirli in modo indipendente.

Complessivamente, l'architettura dell'applicazione può essere suddivisa in quattro macro-componenti:

1. Backend Server (Spring Framework): Questa componente gestisce la logica di business dell'applicazione e comunica con il database per l'accesso ai dati. Il framework Spring è spesso utilizzato per lo sviluppo del backend, fornendo un'ampia gamma di funzionalità e facilitando l'integrazione con altre componenti del sistema.
2. Frontend Server (Flutter Framework): Il frontend server si occupa dell'interfaccia utente dell'applicazione. Utilizzando il framework Flutter, è possibile creare un'interfaccia web interattiva e responsiva che consenta agli utenti di interagire con il sistema. Il frontend server comunica con il backend server per ottenere i dati e inviare le richieste dell'utente.
3. Database (MySQL): Questa componente gestisce la memorizzazione dei dati dell'applicazione. MySQL è un sistema di gestione di database relazionali ampiamente utilizzato che fornisce affidabilità, scalabilità e funzionalità avanzate per la gestione dei dati.
4. Servizi Cloud (Azure): I servizi cloud di Azure possono essere utilizzati per integrare funzionalità aggiuntive nell'applicazione, come l'archiviazione dei file, l'elaborazione dei dati o l'autenticazione degli utenti. Azure offre una vasta gamma di servizi scalabili e affidabili che possono essere integrati nell'architettura complessiva dell'applicazione.

Questa suddivisione in macro-componenti permette di separare le responsabilità e facilitare lo sviluppo, il testing e la manutenzione dell'applicazione. Inoltre, l'utilizzo di tecnologie consolidate come Spring Framework, Flutter, MySQL e servizi cloud come Azure contribuisce a garantire la robustezza, le prestazioni e l'affidabilità dell'intero sistema.

## **Database – MySQL**

La scelta del database per la creazione del progetto è ricaduta su MySQL. Esso è un sistema di gestione di database relazionale (RDBMS) open source che offre una vasta gamma di funzionalità per la gestione e l'elaborazione dei dati. È stato sviluppato originariamente da Michael Widenius e David Axmark presso la società MySQL AB, ed è attualmente di proprietà di Oracle Corporation. MySQL è ampiamente utilizzato in diversi settori, tra cui lo sviluppo di applicazioni web, l'e-commerce, i social media e molto altro ancora. La sua popolarità è dovuta alla sua affidabilità, prestazioni elevate e alla sua natura open source.

MySQL offre numerose caratteristiche che lo rendono un sistema di gestione di database potente ed efficiente. Alcune delle sue principali caratteristiche sono:

1. Linguaggio SQL: MySQL utilizza il linguaggio SQL (Structured Query Language) per interrogare, manipolare e gestire i dati nel database. Il linguaggio SQL di MySQL è facile da imparare e offre una sintassi intuitiva per eseguire operazioni come la creazione di tabelle, l'inserimento di dati, l'aggiornamento e la cancellazione di record.
2. Scalabilità: MySQL è noto per la sua scalabilità. Può gestire grandi quantità di dati e supporta l'archiviazione distribuita, consentendo di espandere facilmente il database in base alle esigenze aziendali. Inoltre, MySQL supporta la replicazione, che permette di creare copie dei dati su server multipli per migliorare la disponibilità e le prestazioni del sistema.
3. Prestazioni elevate: MySQL è progettato per offrire prestazioni elevate. Ottimizza le query in modo efficiente, utilizza indici per accelerare la ricerca dei dati e offre cache di query per ridurre i tempi di risposta. Inoltre, supporta l'ottimizzazione delle tabelle e delle query per migliorare ulteriormente le prestazioni del database.
4. Sicurezza dei dati: MySQL offre diverse funzionalità di sicurezza per proteggere i dati sensibili. Supporta l'autenticazione basata su password, il controllo degli accessi basato sui ruoli e la crittografia dei dati per garantire la privacy e la protezione delle informazioni sensibili.
5. Funzionalità avanzate: MySQL include molte funzionalità avanzate che consentono di gestire i dati in modo efficace. Queste includono stored procedure, funzioni definite dall'utente, trigger per automatizzare le azioni del database e la gestione degli eventi schedulati.

MySQL trova applicazione in una vasta gamma di scenari. Alcuni esempi di applicazioni pratiche di MySQL includono:

1. Sviluppo di applicazioni web: MySQL è ampiamente utilizzato come database backend per le applicazioni web.
2. Analisi dei dati: MySQL può essere utilizzato per l'archiviazione e l'elaborazione dei dati utilizzati nell'analisi dei dati.
3. Sistemi di gestione delle informazioni: MySQL può essere utilizzato come sistema di gestione delle informazioni per organizzare e gestire dati complessi, come ad esempio dati di inventario, informazioni sugli utenti e registrazioni di transazioni.

MySQL è un sistema di gestione di database relazionale potente ed estensibile che offre numerose funzionalità per la gestione efficace dei dati. Le sue caratteristiche di scalabilità, prestazioni elevate, sicurezza e facilità d'uso lo rendono una scelta popolare per gli sviluppatori e gli amministratori di database. L'utilizzo di MySQL può migliorare l'efficienza operativa, ottimizzare le prestazioni e consentire l'elaborazione dei dati in modo sicuro.

## Immagine che contiene Elementi grafici, Carattere, grafica, logo Descrizione generata automaticamente**Backend – Spring**

Spring Boot Web MVC fornisce un'architettura Model-View-Controller (MVC) preconfigurata che consente di separare gli aspetti distinti dell'applicazione, come la logica dei dati, la logica di business e la logica di input. La seguente è una breve descrizione di ciascun componente all'interno di questa architettura:

1. Model: Il modello rappresenta la logica dei dati dell'applicazione. Include la gestione e la manipolazione dei dati, nonché la definizione delle regole di validazione. È responsabile di interagire con il database e fornire le informazioni necessarie alla vista e al controller.
2. View: La vista è responsabile della presentazione dei dati all'utente finale. Può essere un'interfaccia utente grafica (GUI) o una rappresentazione dei dati in formato testuale. La vista si occupa di visualizzare i dati in modo appropriato e di acquisire l'input dell'utente.
3. Controller: Il controller funge da intermediario tra il modello e la vista. Riceve le richieste dell'utente dalla vista, elabora la logica di business necessaria per soddisfare la richiesta e coordina il modello per restituire i dati appropriati alla vista. Inoltre, il controller gestisce anche gli eventi e le azioni utente, come i click dei pulsanti o le richieste di input.

Oltre all'architettura MVC, il framework Spring Boot fornisce anche due ulteriori componenti:

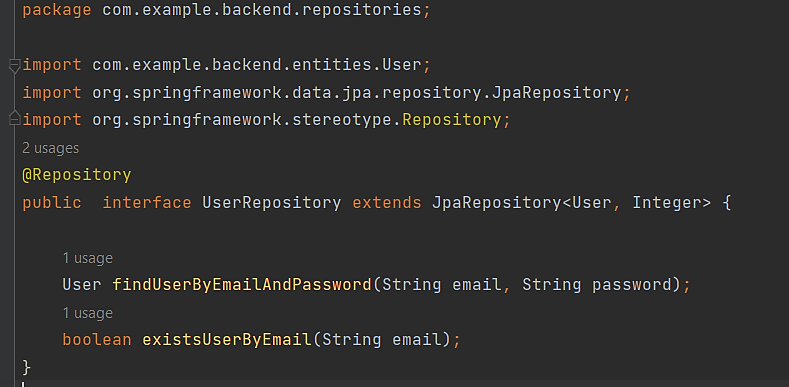
* Strato DAO (Data Access Object): Il DAO layer si interfaccia con una risorsa dati, come un database, e fornisce un'interfaccia generica per accedervi. Questo strato separa la logica di accesso ai dati dal resto dell'applicazione, consentendo di modificare i meccanismi di accesso senza influire sul codice che li utilizza. Questo favorisce la modularità e la manutenibilità del codice.
* Strato di servizio: Il service layer incapsula la logica di business dell'applicazione e definisce le funzionalità offerte indipendentemente dalla modalità di accesso ai dati. Questo strato consente di raggruppare la logica di business correlata e di renderla riutilizzabile in diversi contesti. Il servizio può coordinare le operazioni tra più oggetti DAO e implementare regole di business complesse.

### **DAO Layer**

In questo livello vengono definiti gli oggetti con i quali ci si interfaccia a livello fisico, si tratta delle tabelle del database ed anche di come queste vengono accedute. Spring, infatti, offre delle librerie per automatizzare l’esecuzione di query sul database che prendono il nome di JPA Repository.

Gli oggetti vengono rappresentati attraverso una classe di tipo Entity per ciascuna tabella:

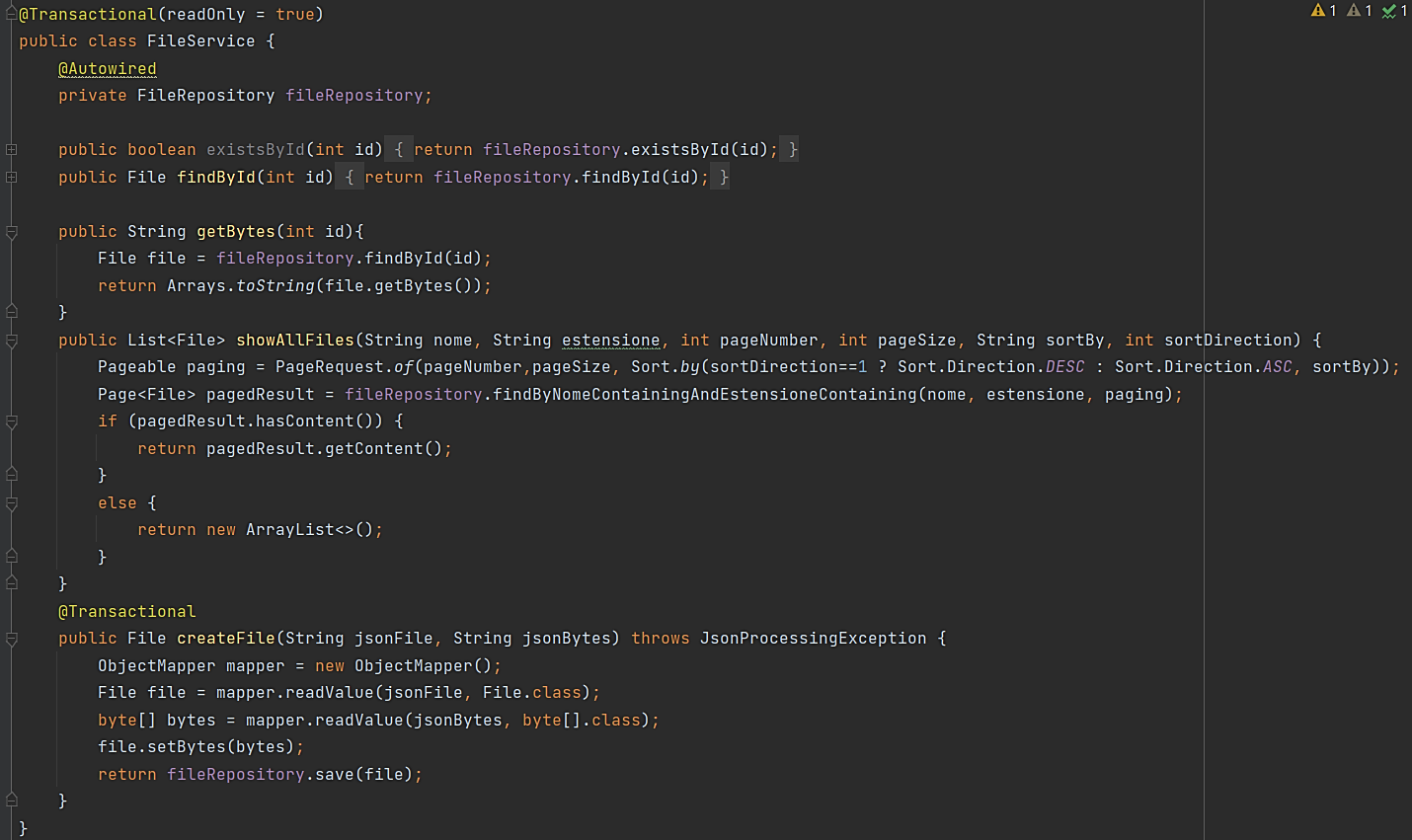
La struttura è la stessa per le altre tabelle. E poi sono presenti le classi relative alle Repository, che è dove vengono dichiarate le query che si vogliono poter eseguire sul Database.

User Repository:

File Repository:

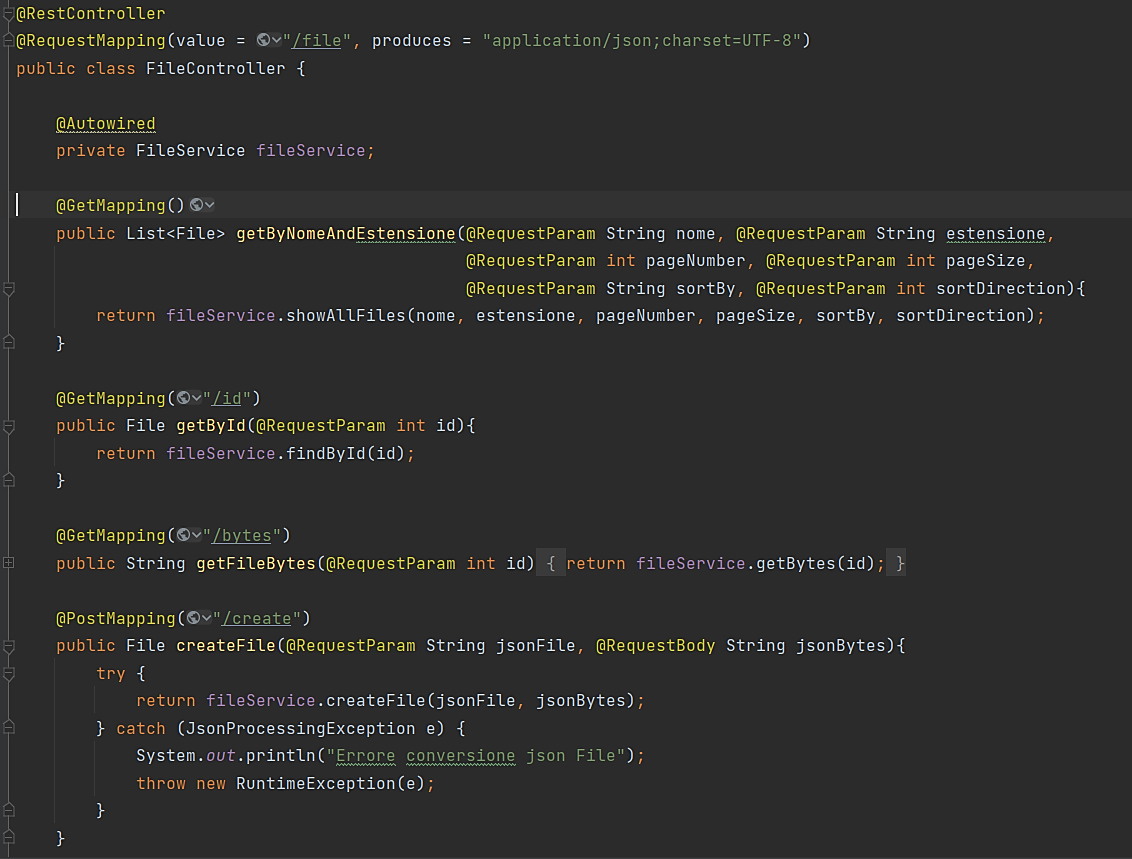
### **Service Layer**

Il service layer, responsabile dell'implementazione della logica di business dell'applicazione, può essere suddiviso in diversi componenti per garantire una migliore modularità e coesione interna. Di seguito sono elencati alcuni dei principali componenti che possono essere presenti all'interno del service layer:

* UserService: Questo componente gestisce le operazioni relative agli utenti dell'applicazione, come la registrazione, l'autenticazione, la gestione dei ruoli e dei permessi, e la gestione del profilo utente.
* File Service: gestisce la logica per il caricamento dei file all'interno del sistema, la gestione dell'archiviazione dei file su disco o su un servizio di storage cloud, e la generazione di identificatori univoci per i file. Questa è probabilmente la classe più complessa in quanto al suo interno è realizzato il salvataggio e indicizzazione dei file.

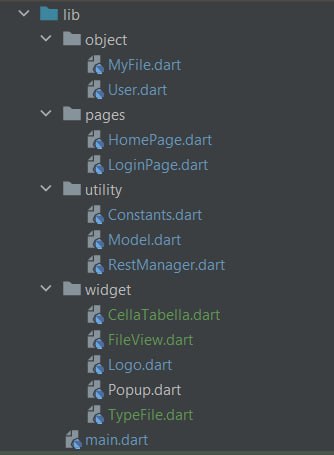
### **Controller Layer**

In questa classe vengono implementati gli strumenti di comunicazione con il Backend, fornendo le interfacce che espongono le API ai client, utilizzando il modello architetturale REST. Sono presenti due diverse classi di Controller, ciascuna dedicata a un oggetto Entity specifico:

* UserController: Questa classe contiene i metodi necessari per gestire le operazioni di login e registrazione degli utenti. I metodi implementati consentono di autenticarsi nel sistema utilizzando le credenziali dell'utente o di registrarsi come nuovo utente fornendo le informazioni richieste.
  + FileController: Questa classe contiene diversi metodi per gestire le operazioni legate ai file. I metodi implementati consentono la ricerca di un file in base al nome o al tipo di file (o entrambi), permettendo agli utenti di trovare facilmente i file desiderati. Inoltre, è presente un metodo POST che riceve i dati relativi ai nuovi file creati, consentendo l'inserimento di nuovi file nel sistema.

## **Frontend – Flutter**

Per la creazione del front-end dell’applicazione, quindi della parte visibile e interattiva per il client la scelta è ricaduta su Flutter. Esso è un framework open source sviluppato da Google per la creazione di applicazioni native multipiattaforma. Le principali caratteristiche che rendono Flutter un’ottima scelta per la creazione del front-end sono diverse:

1. Hot Reload: Una delle caratteristiche distintive di Flutter è l'hot reload, che consente agli sviluppatori di apportare modifiche al codice in tempo reale e visualizzare immediatamente i risultati nell'applicazione in esecuzione.
2. Widget personalizzabili: Flutter utilizza un approccio basato sui widget per costruire le interfacce utente. Offre una vasta gamma di widget predefiniti altamente personalizzabili e flessibili
3. Prestazioni elevate: Flutter utilizza il motore di rendering Skia per offrire prestazioni fluide e veloci.
4. Supporto: Essendo sviluppato e utilizzato da Google, Flutter gode di una vasta community a livello globale che fornisce supporto e risorse.

Il progetto in Flutter è strutturato come segue:

* Object: Questa sezione del codice definisce i modelli di dati utilizzati nell'applicazione. I modelli di dati sono utilizzati per mappare gli oggetti JSON ricevuti dal back-end in oggetti Dart che possono essere manipolati e utilizzati nell'applicazione.
* Utility: In questa parte del codice si trovano le classi di utilità. Queste classi forniscono funzionalità aggiuntive che facilitano lo sviluppo dell'applicazione. Ad esempio, l'interfaccia per l'inoltro delle richieste al back-end Spring è implementata qui. Inoltre, il RestManager, che si occupa di raccogliere le richieste provenienti dall'ApiController, le incapsula in richieste HTTP e le inoltra al back-end, è anche incluso in questa sezione.
* Pages: Questa sezione definisce l'interfaccia utente dell'applicazione. Qui sono presenti le diverse schermate principali che vengono visualizzate nell'applicazione. Le pagine sono responsabili di mostrare i dati agli utenti e di interagire con essi.
* Widget: Questa parte del codice contiene delle componenti grafiche che possono essere riutilizzate in più pagine. I widget sono delle strutture grafiche che consentono di creare interfacce utente più complesse e modulari. Essi possono essere utilizzati per creare bottoni, caselle di testo, icone e altre componenti visive presenti nelle pagine dell'applicazione.

# **Deploy – Azure Cloud Services**

Immagine che contiene Elementi grafici, Carattere, grafica, logo

Descrizione generata automaticamenteNella fase finale del progetto, la Web-App è stata distribuita su Azure utilizzando il servizio di calcolo Azure Virtual Machine. Azure Virtual Machine offre un ambiente di esecuzione flessibile e scalabile per ospitare applicazioni e servizi. Grazie a questo servizio, è stato possibile sfruttare le risorse di calcolo e virtualizzazione fornite da Azure per eseguire un sistema operativo dedicato sulla quale esportare e ospitare la Web-App.

Azure Virtual Machine supporta sia sistemi operativi Windows che Linux, offrendo una vasta scelta di opzioni per la configurazione dell'ambiente di esecuzione. Ciò consente di adattare l'infrastruttura alle specifiche esigenze dell'applicazione e del progetto.

Uno dei vantaggi principali di Azure Virtual Machine è la sua capacità di auto-scaling, che consente di gestire in modo dinamico le risorse allocate in base al carico di lavoro. Questo significa che l'ambiente di esecuzione può espandersi o ridursi automaticamente in risposta alle variazioni di traffico e di richieste degli utenti, garantendo prestazioni ottimali e riducendo i costi inutilizzati.

Azure Virtual Machine offre anche diverse opzioni per il deployment dell'applicazione. È possibile caricare l'immagine del sistema operativo desiderato, configurare le risorse di calcolo e di rete, e quindi distribuire l'applicazione tramite i protocolli e gli strumenti di deployment supportati.

Complessivamente, l'utilizzo del servizio di calcolo Azure Virtual Machine per la distribuzione della Web-App su Azure ha fornito un'infrastruttura affidabile, scalabile e flessibile. Questo ha consentito di garantire un ambiente di esecuzione ottimale per l'applicazione, gestendo in modo efficiente le risorse e adattandole alle esigenze del progetto.

## **Configurazione del servizio Azure**

L'abbonamento Azure for Student offre la quasi totalità dei servizi disponibili e un credito iniziale di 100$. Pertanto, abbiamo scelto di configurare una singola macchina virtuale di tipo B2, poiché ha un costo accessibile e offre una quantità di risorse adeguata al progetto.

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamenteCome si può osservare nell'immagine successiva, la creazione di una risorsa VM con le specifiche e le impostazioni necessarie per distribuire la WebApp comporta la creazione delle seguenti risorse:

Incluso in queste risorse c'è un indirizzo IP pubblico, che consente l'accesso alla porta 80 esposta sulla stessa macchina in cui è ospitata la WebApp.

La VM creata ospita un sistema operativo Linux, nello specifico la distribuzione Ubuntu 20. Questa scelta è stata fatta per la facilità con cui è possibile avviare servizi e gestire la macchina tramite il Terminale. Infatti, per accedere alla macchina, è stato necessario configurare un nome utente e una password per consentire la connessione tramite il protocollo SSH, ampiamente conosciuto e utilizzato, soprattutto in ambiente Linux, per l'accesso remoto sicuro alle macchine.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamenteLe risorse disponibili sulla macchina sono limitate, come accennato in precedenza, ma sono sufficienti per il nostro scopo.

## **Accesso, trasferimento file ed Deploy sulla VM**

Come accennato precedentemente, l'accesso alla risorsa in cloud avviene tramite il servizio SSH, preinstallato nella maggior parte delle distribuzioni Linux. Dopo aver stabilito la connessione, i passaggi successivi per completare il deploy sono stati eseguiti sulla riga di comando del sistema operativo Linux.

Innanzitutto, è stato necessario aggiornare il sistema per garantire la presenza e la compatibilità delle dipendenze e delle librerie. A tal fine, sono stati eseguiti i seguenti comandi:

* + **sudo apt update**
  + **sudo apt upgrade**

Successivamente, sono stati installati i software necessari per procedere:

* + **Java JDK**: è necessario per avviare il file eseguibile JAR ottenuto tramite il processo di confezionamento standard fornito dal gestore di dipendenze Maven, integrato nel framework Spring utilizzato per lo sviluppo del backend.
  + **MySQL server:** è necessario per mantenere la persistenza dei dati. Il software è stato configurato tramite comandi sulla riga di comando, inclusi comandi come CREATE e INSERT, attraverso i quali sono state replicate le tabelle User, Review e Book menzionate in precedenza, e sono state popolate con righe relative a un insieme di libri, recensioni e utenti generati tramite semplici script o reperiti su Internet in formato Excel Sheets.

I file necessari, inclusi il file JAR eseguibile del backend, la cartella radice "Web/" e gli script SQL per la generazione del database, sono stati trasferiti sulla macchina virtuale tramite il seguente comando:

**scp ./FileArchive.rar Pasquale@\*IP MACCHINA AZURE\*:/home/Pasquale/CloudProject**

È stato utilizzato SCP (Secure Copy), un protocollo di rete basato su SSH (Secure Shell), per il trasferimento sicuro dei file tra un computer locale e un server remoto.

Per concludere l'intera procedura di deploy e automatizzare l'esecuzione dei vari servizi, tra cui il backend server e il server MySQL, è stato creato uno script che viene eseguito ogni volta che la VM Azure viene avviata.

# **Risultato Finale**

Per potere testare il prodotto finale, una volta avviata la VM è possibile collegarsi tramite qualsiasi browser all’indirizzo IP della stessa, o meglio ancora, digitando il suo DNS (Domain Name Service) fornito da Azure.

Immagine che contiene testo, schermata, multimediale, Sistema operativo

Descrizione generata automaticamenteUna volta connesso a tale URL è possibile iniziare a navigare e visualizzare la WebApp:

Immagine che contiene testo, schermata, software, Sistema operativo

Descrizione generata automaticamenteQuindi inserendo le credenziali di un generico utente (che se non presente può essere creato tramite la form di Registrazione) o accedendo tramite le credenziali dell’admin è possibile procedere alla schermata successiva di visualizzazione dei file:

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamenteQui è possibile effettuare la ricerca dei file inserendone il nome, e filtrando anche i contenuti per tipo di estensione. È anche possibile effettuare il logout, e soltanto per gli utenti amministratori, si può creare dei nuovi file tramite l’apposito bottone:

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamenteI file visualizzati presenti sul Database possono anche essere ordinati per Nome o per Estensione, e ovviamente possono essere scaricati tramite l’apposito tasto di download.

L’inserimento di un nuovo file è effettuato tramite una Form che richiede di inserire un nome per quel file ed una descrizione; a questo punto è possibile selezionare il file dal proprio filesystem e quindi confermare l’operazione.