



UNIVERSITÀ DI PISA

**Documentazione del
Progetto “MyAgenda”
di Programmazione Avanzata**

Anno Accademico 2024-25

Simone Calò

Sommario

CARATTERI GENERALI	3
CASO D'USO	3
STRUTTURA DELL'APPLICAZIONE	3
CLIENT	4
SCHERMATE.....	4
SCHERMATA HOME	4
SCHERMATA REGISTRATI E SCHERMATA ACCEDI	4
SCHERMATA TUTTI	5
SCHERMATA CERCA.....	6
SCHERMATA INSERISCI	6
CARATTERISTICHE TECNICHE	7
CREDENZIALI DI PROVA	7
SERVIZIO WEB	8
CARATTERISTICHE TECNICHE	8

CARATTERI GENERALI

CASO D'USO

MyAgenda è un'applicazione che permette all'utente di mantenere un elenco delle attività da svolgere, assegnando ad ognuna un titolo, una categoria e una breve descrizione.

STRUTTURA DELL'APPLICAZIONE

MyAgenda è un'applicazione distribuita:

- L'utente interagisce con l'applicazione usando un client realizzato attraverso JavaFX;
- Il client comunica con un servizio web realizzato attraverso Spring;
- Il servizio web risponde alle richieste del client sulla base dei dati memorizzati in un database MySQL.

Il client richiede un'operazione al servizio web attraverso il protocollo HTTP:

- Per ottenere un dato, il client invia al servizio web una richiesta HTTP di tipo GET, codificando nell'URL eventuali informazioni aggiuntive sul dato da ottenere, e il servizio web risponde inviando un oggetto che rappresenta tale dato, codificato in formato JSON nel corpo della risposta;
- Per inviare un dato, il client invia al servizio web una richiesta HTTP di tipo POST, codificando nel corpo della richiesta il dato da inviare in formato JSON, e il servizio web risponde inviando un oggetto *Esito*, codificato in formato JSON nel corpo della risposta;
- Per eliminare un dato, il client invia al servizio web una richiesta HTTP di tipo DELETE, codificando nell'URL eventuali informazioni aggiuntive sul dato da eliminare, e il servizio web risponde inviando un oggetto *Esito*, codificato in formato JSON nel corpo della risposta.

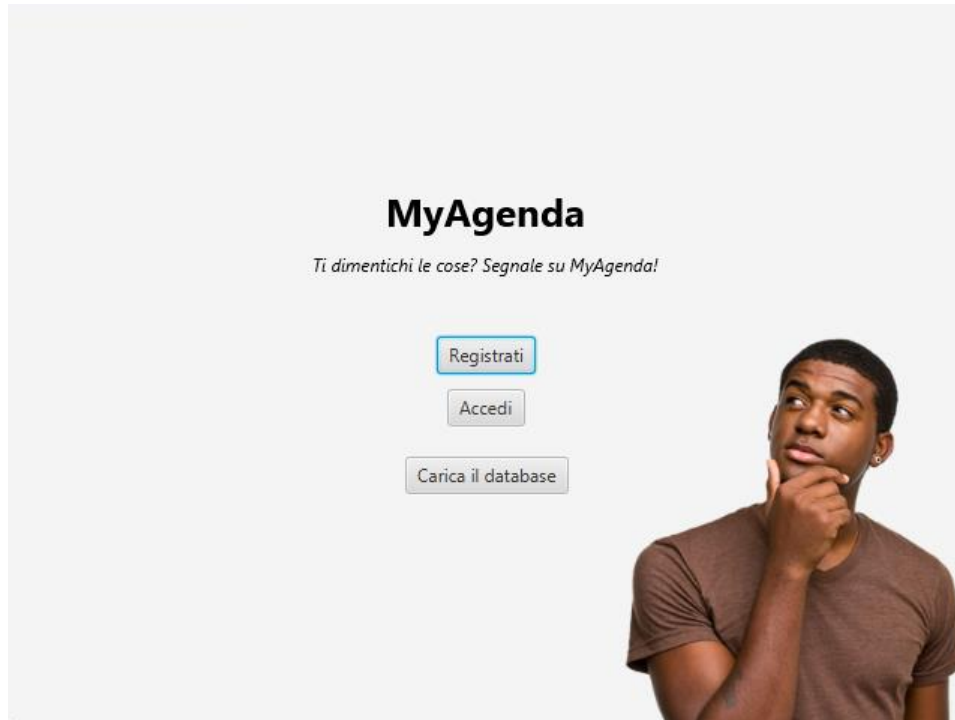
Un oggetto *Esito* rappresenta l'esito di un'operazione richiesta dal client. E' costituito da due campi:

- Un campo boolean *status*, che vale false se l'esito dell'operazione è negativo;
- Un campo String *messaggio*, contenente informazioni aggiuntive sull'esito dell'operazione.

CLIENT

SCHERMATE

SCHERMATA HOME

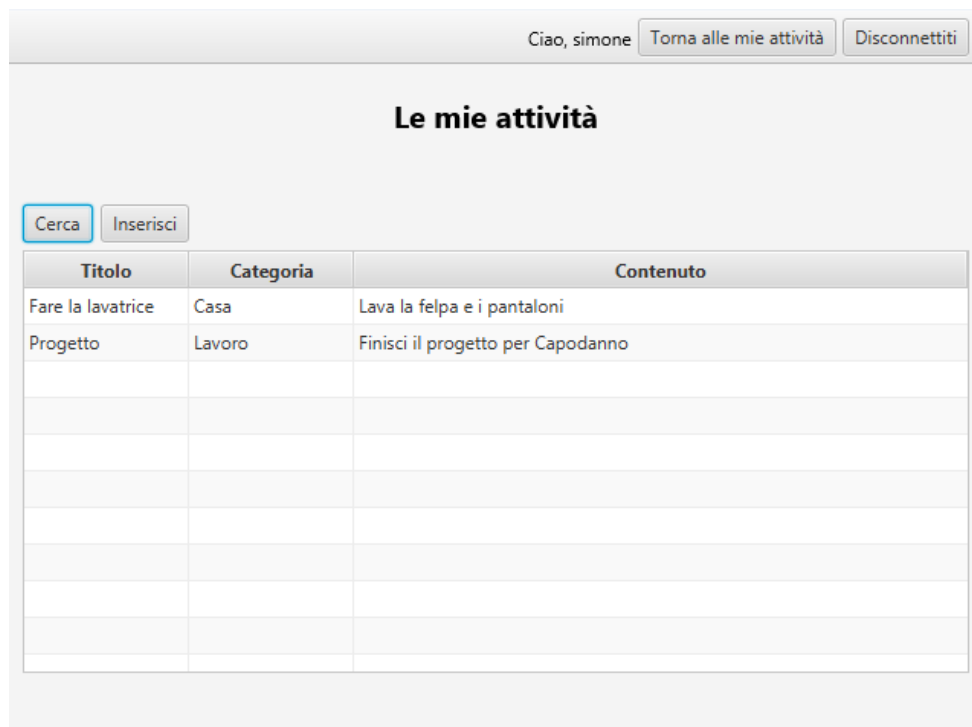


Il client si apre con la schermata “home”, che permette all’utente di registrarsi o accedere all’applicazione. E’ possibile inoltre caricare il database con dati di prova, in modo da testare l’applicazione.

SCHERMATA REGISTRATI E SCHERMATA ACCEDI

Le schermate “registrati” e “accedi” permettono all’utente rispettivamente di registrarsi e accedere all’applicazione, inserendo un username univoco e una password.

SCHERMATA TUTTI



La schermata “tutti” mostra tutte le attività dell’utente. Premendo sugli appositi bottoni è possibile cercare un’attività o inserirne una nuova, mentre premendo con il tasto destro del mouse su un’attività è possibile rimuoverla.

Questa schermata e tutte le schermate successive mostrano in alto una toolbar contenente l’username dell’utente, un bottone per tornare a questa schermata e un altro bottone per disconnettersi dall’applicazione, tornando così alla schermata “home”.

SCHERMATA CERCA

Ciao, simone

Torna alle mie attività

Disconnettiti

Cerca un'attività

Testo

Categoria

Cerca per:

lavatrice

Casa

☒ Titolo
☐ Contenuto

Cerca

Titolo	Categoria	Contenuto
Fare la lavatrice	Casa	Lava la felpa e i pantaloni

La schermata “cerca” permette all’utente di cercare un’attività. Una volta inserito il testo da cercare, è possibile specificare la categoria delle attività da cercare e se le attività da cercare devono presentare il testo inserito nel titolo o nel contenuto. Premendo con il tasto destro su un’attività cercata è possibile rimuoverla.

SCHERMATA INSERISCI

Ciao, simone

Torna alle mie attività

Disconnettiti

Inserisci un'attività

Titolo

Categoria

Contenuto

Soldi Capodanno

Finanze

Dai 10 euro a GabrYLiv_ per la cena di Capodanno

Inserisci

La schermata “inserisci” permette all’utente di inserire una nuova attività. Una volta specificato il titolo e la categoria dell’attività, è possibile specificare una breve descrizione.

CARATTERISTICHE TECNICHE

Per ridurre la ripetizione di codice nella definizione delle classi di controllo delle schermate sono state definite le seguenti sottoclassi:

- *BaseController*: contiene i metodi per mostrare messaggi di errore/successo e cambiare schermata;
- *AutenticazioneController*: contiene i riferimenti agli elementi dell'interfaccia comuni alla schermata "registrati" e alla schermata "accedi";
- *AgendaController*: contiene i riferimenti agli elementi della toolbar e i metodi da eseguire quando si interagisce con essa;
- *TabellaController*: contiene un riferimento alla tabella delle attività e il metodo per costruire le sue colonne dinamicamente.

Le operazioni che richiedono l'interazione con il servizio web, potendo portare a lunghe attese, vengono realizzate mediante Task. Mentre un Task è in esecuzione, le parti dell'interfaccia che potrebbero dare origine ad operazioni in conflitto con quelle eseguite dal Task vengono temporaneamente disabilitate, riabilitandole solo al termine dell'esecuzione del Task.

Per ridurre la ripetizione di codice nella comunicazione con il servizio web sono state definite due classi *HTTPConnector* e *Utilities*, dove la prima contiene alcuni metodi per effettuare richieste HTTP al servizio web, mentre la seconda, sfruttando la prima, contiene alcuni metodi per ottenere, inviare o cancellare particolari dati.

Per rendere più efficiente la gestione degli errori sono state definite le seguenti eccezioni:

- *CaricamentoException*: lanciata in caso di errore nel caricamento del database con dati di prova;
- *UtenteException*: lanciata in caso di errore nella registrazione/accesso;
- *AttivitaException*: lanciata in caso di errore nell'inserimento/ricerca/eliminazione di un'attività;
- *TestException*: lanciata in caso di errore nello Unit Test.

Per mantenere i dettagli dell'utente corrente è stata definita una classe *Sessione*, realizzata seguendo il Singleton Pattern. L'unico oggetto *Sessione* istanziato è costituito da due campi:

- Un campo int *ID*, contenente l'ID dell'utente corrente. Questo ID viene assegnato dal database all'utente in fase di registrazione, viene mandato al client dal servizio web al termine della registrazione o dell'accesso ed è usato per identificare univocamente l'utente e le sue attività, per cui fa parte dei dati mandati dal client al servizio web nelle operazioni di inserimento e ricerca di attività.
- Un campo String *username*, contenente l'username dell'utente corrente. Viene usato per mostrare l'username dell'utente nella schermata "tutti" e in quelle successive.

Lo Unit Test verifica che l'oggetto *Sessione* sia gestito correttamente.

CREDENZIALI DI PROVA

Una volta caricato il database con i dati di prova, è possibile testare l'applicazione inserendo, in fase di accesso, le seguenti credenziali:

- Username: simone
- Password: segreto

SERVIZIO WEB

CARATTERISTICHE TECNICHE

L'interazione con il database avviene usando Spring JPA durante l'esecuzione normale e Spring JDBC per lo Unit Test e per caricare il database con i dati di prova dal file "615600.sql". Per rendere più semplice la scrittura di codice sia lato client che lato servizio web è stato assunto che, durante l'esecuzione normale, nell'interazione con il database non si verifichino mai errori.

Per rendere più efficiente la gestione di un errore commesso dal client in fase di accesso/registrazione è stata definita l'eccezione *UtenteException*.

Lo Unit Test verifica che il database sia stato effettivamente creato.