

## Architettura del sistema

L'applicazione simil IFTTT è implementata usando linguaggio Java con l'utilizzo di JavaFX tramite tool Scene Builder per l'implementazione dell'interfaccia grafica. Verrà utilizzato il pattern Model-View-Controller (MVC) in quanto JavaFX consente di progettare con MVC, attraverso l'uso di FXML e Java. Nel nostro caso le tre parti del pattern conterranno le seguenti logiche:

- Model: dovrà contenere le regole descritte dall'utente (in futura implementazione esse saranno salvate in maniera persistente).
- View: un insieme di viste dove la prima vista renderà visibili le diverse regole attive e disattive definite dall'utente, mentre la seconda vista permetterà di eseguire azioni di aggiunta di tali regole (selezionando trigger e/o azione associata).
- Controller: questa parte dovrà controllare la verifica dei trigger associati alle diverse regole ed eseguire le azioni (mostrando in interfaccia utente possibili risultati).

In particolare, si pensa di implementare:

- "Rule View" che sarà responsabile della visualizzazione delle regole;
- "Rule Controller" gestore dell'interazione utente e intermediario tra Rule Model (database delle regole) e Rule View. Esso aggiornerà la vista tramite "Rule model".
- "Rule Model", che rappresenta il modello delle regole. Il "Rule Controller" aggiornerà la vista attraverso esso.
- "Rule Service" sarà il servizio che gestisce la logica delle regole, in particolare la loro verifica periodica, questo componente genera eventi quando una regola è verificata dal suo Trigger.
- Un modulo che si occuperà di svolgere le diverse azioni definite dall'utente in base alla sua selezione.

Di seguito ecco mostrati dei prototipi di interfaccia "customization rule" e "rules view" in cui:

- La prima View mostra all'utente una pagina in cui vengono mostrate le diverse rules da lui stesso create con diverse opzioni e modalità di utilizzo per tali rules
- La seconda View accede l'utente che ha intenzione di modificare la specifica regola.

