ENERGY SOFTWARE

Alma Mater Studiorum · Università di Bologna

Progettazione e sviluppo di un sistema distribuito di notifiche basato su database real-time

Presentato da:
Francesco Foschini

Anno accademico 2019-2020

Obiettivo e descrizione del progetto

Obiettivo:

• Sviluppo di un sistema che garantisca la distribuzione di notifiche in tempo reale relative ad eventi (es. inserimento di un nuovo ordine) tra i vari client

Attività svolte:

• Automazione dell'avvio del server tramite Docker

• Progettazione e implementazione della libreria *UtilityRethink*

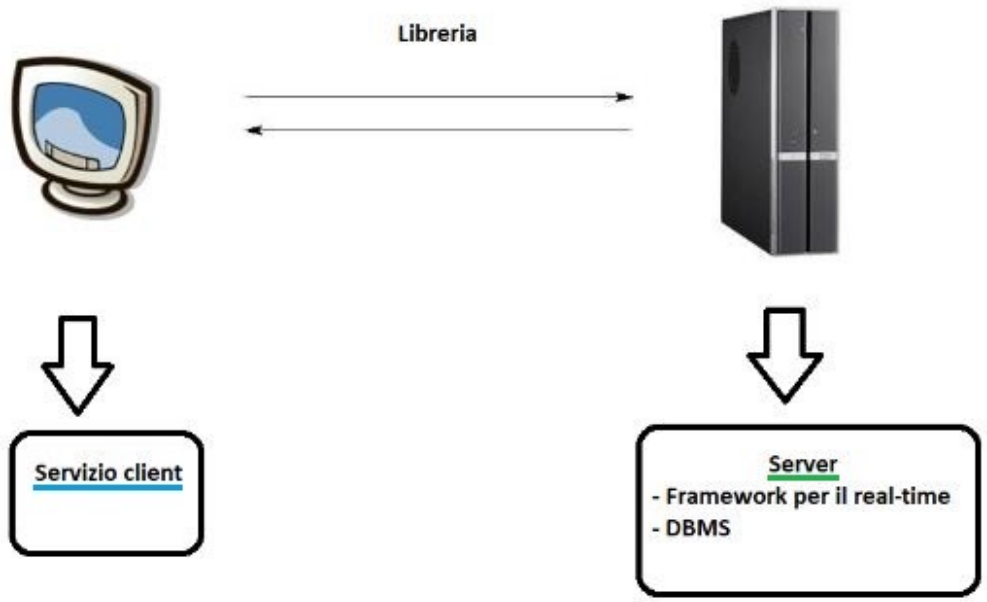
Specifiche architetturali e funzionali

Specifiche architetturali:

- Servizio client
- Libreria
- Server (Framework per il real-time + DBMS)

Specifiche Funzionali:

- Inserire nuove notifiche sul server
- Registrazione per la ricezione delle notifiche di interesse in tempo reale
- Lettura delle notifiche presenti sul server
- Il server deve inoltrare le notifiche ai client in tempo reale



Tecnologia scelta e alternative

Database real-time:

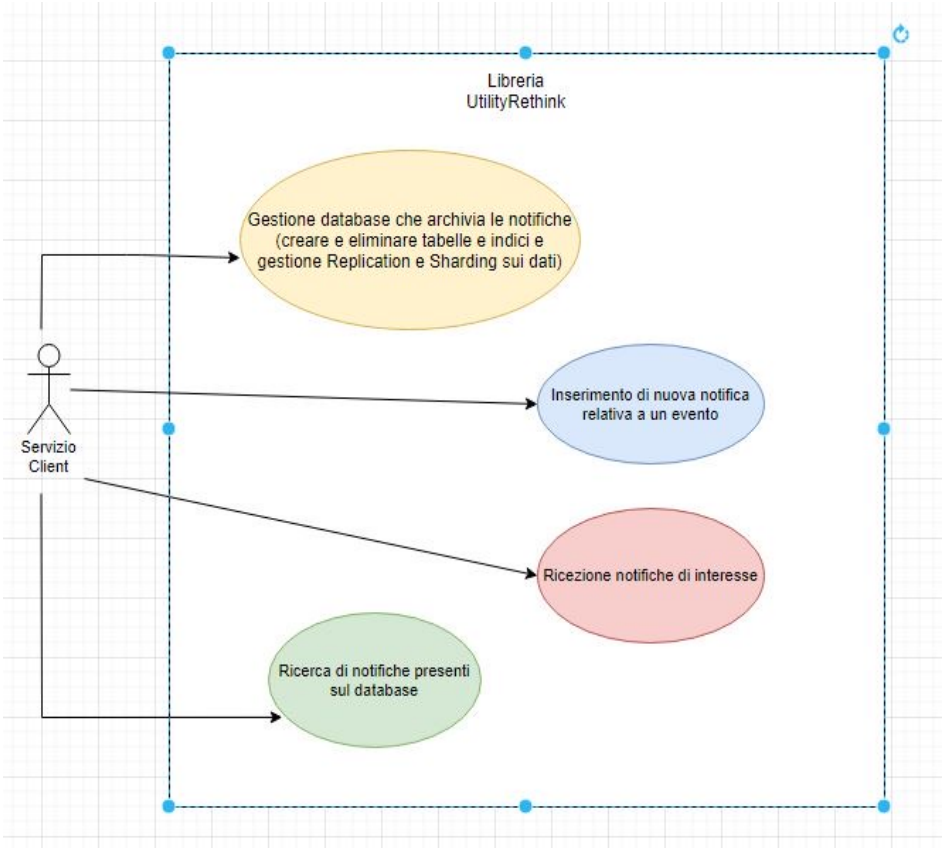
- Nuova classe di sistemi orientati al *push delle modifiche*
- Facilitano notevolmente lo sviluppo di applicazioni web reattive

Possibili soluzioni:

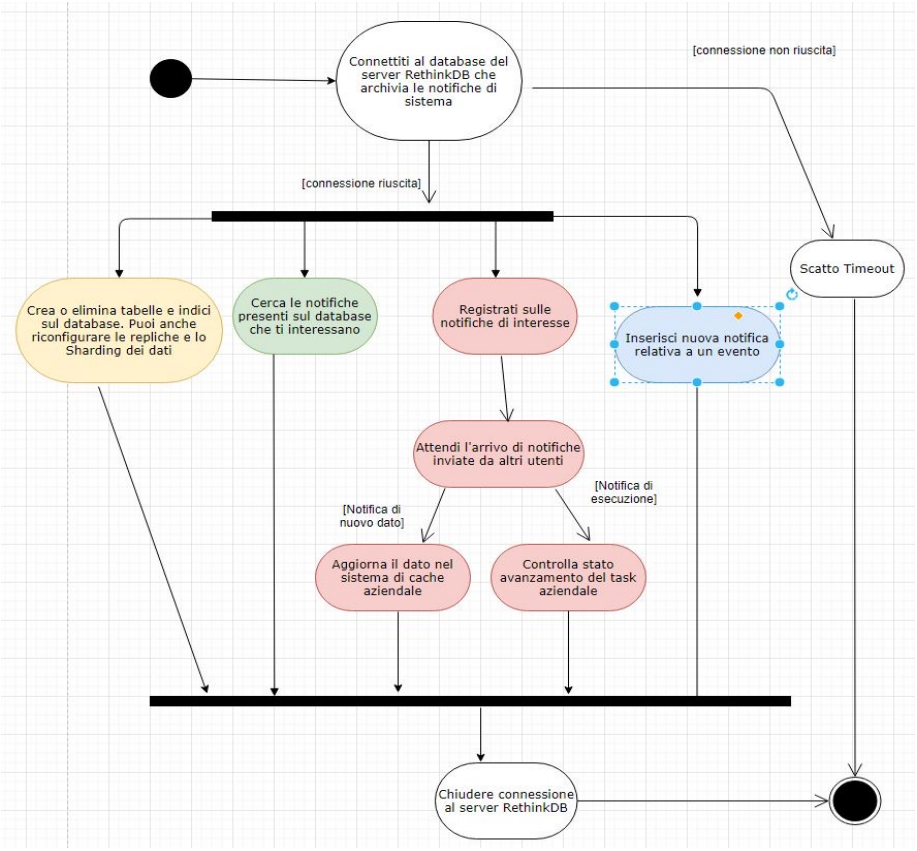
- Firebase
- MongoDB + Meteor
- MongoDB + Parse
- RethinkDB

Progettazione libreria UtilityRethink

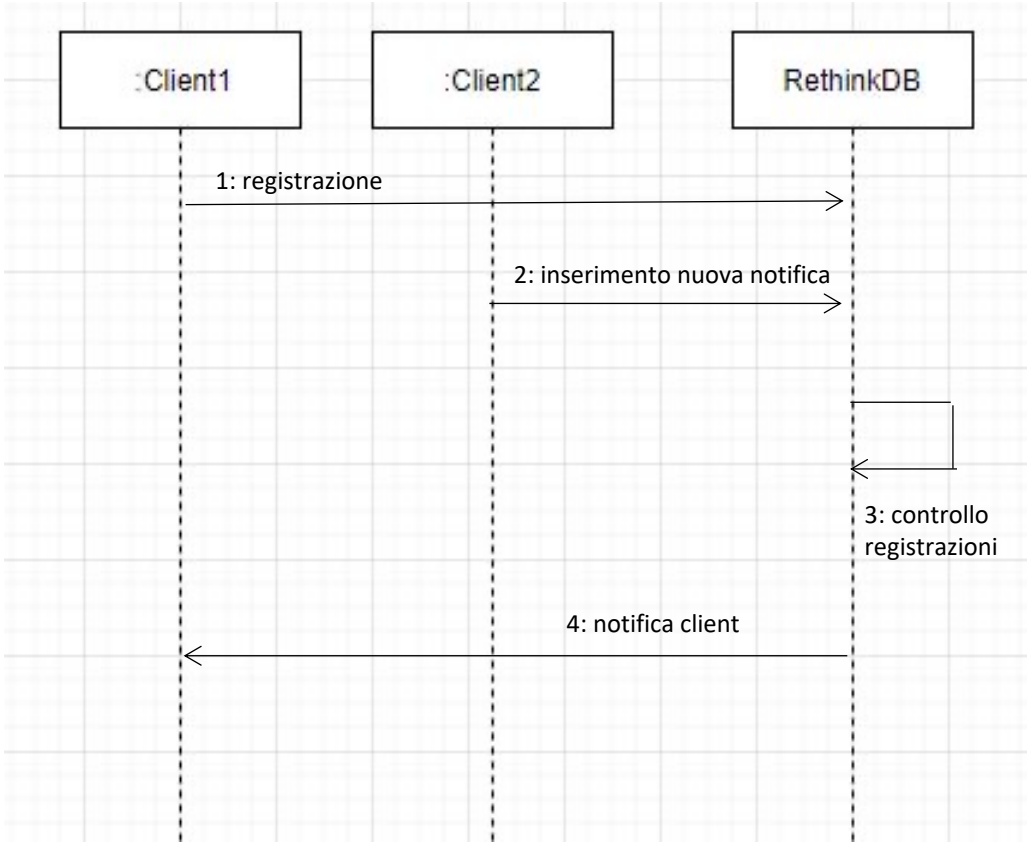
Funzionalità offerte:



Possibili attività:



Esempio di utilizzo della libreria



- Il client 1 si registra sul server per ricevere le notifiche con argomento di proprio interesse

```
var notifier = utilityRethink.GetNotificationsManager().GetNotifier<NotificationNewData>();
NotificationSubscription<NotificationNewData> notSub = notifier.SignUpForTopics("O", "C");
IObservable<Change<NotificationNewData>> observervable = notSub.Observable;
observervable.SubscribeOn(NewThreadScheduler.Default)
    .Subscribe(
        x => OnNext(x, ref onNext),
        e => OnError(e, ref onError),
        () => OnCompleted(ref onCompleted)
    );
```

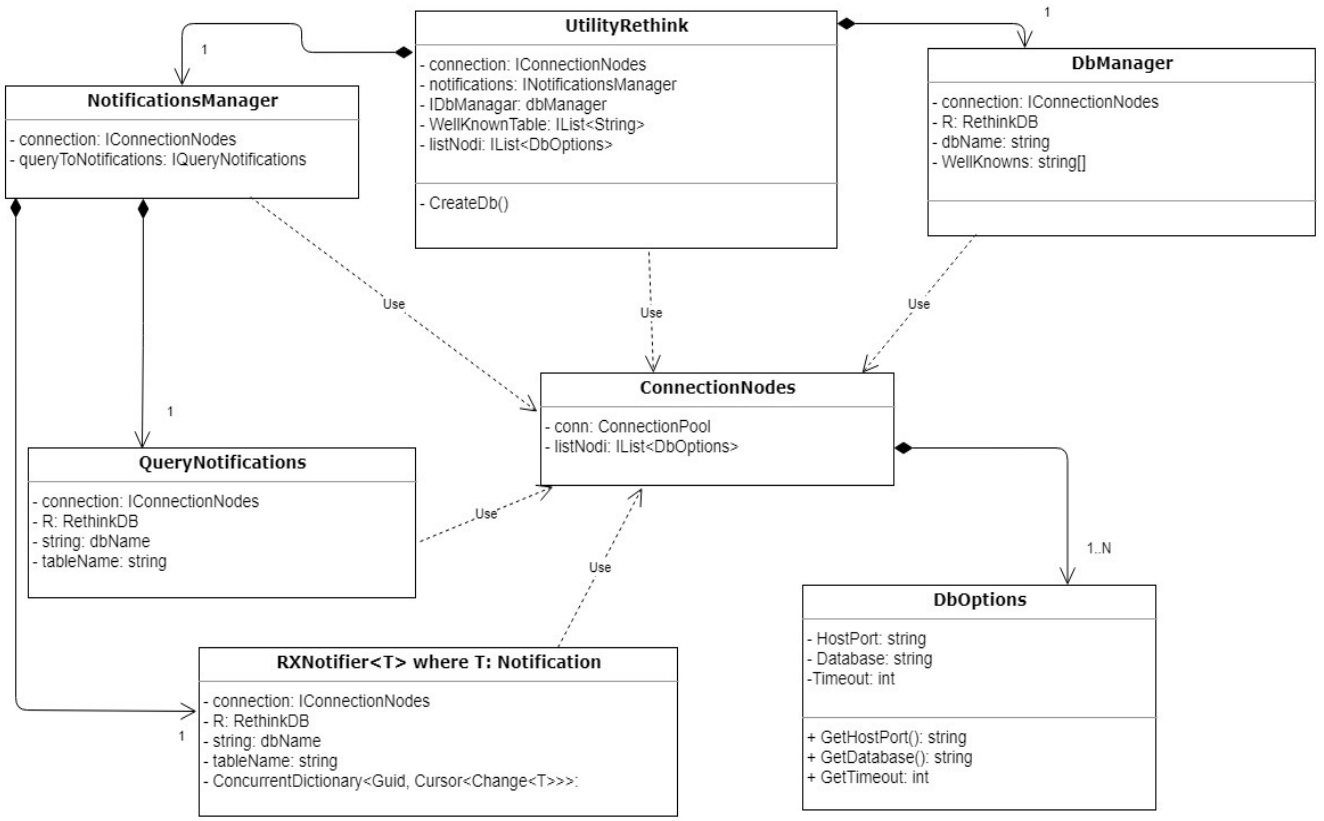
- Il client 2 invia al server una nuova notifica

```
var queryService = utilityRethink.GetNotificationsManager().GetQueryService();
queryService.NewNotification(notificationNewData);
```

Implementazione UtilityRethink

Librerie utilizzate:

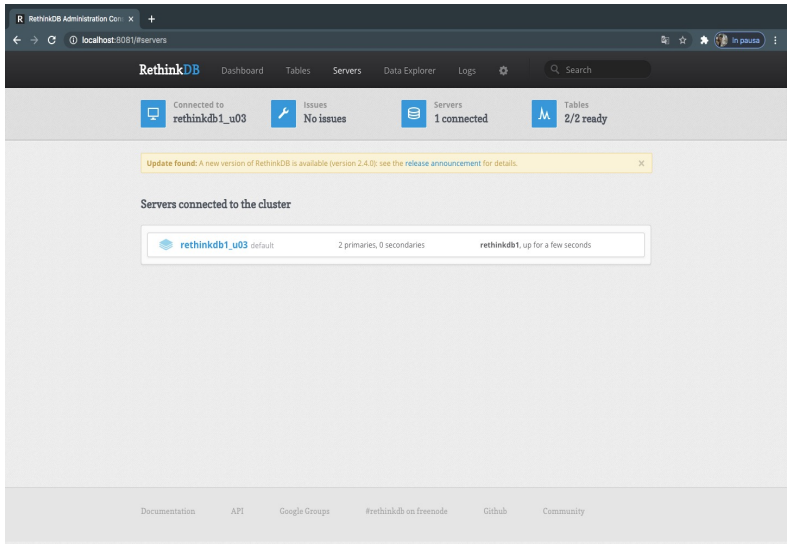
- *Bchavez/RethinkDb.Driver*
- *Reactive Extensions (Rx)*



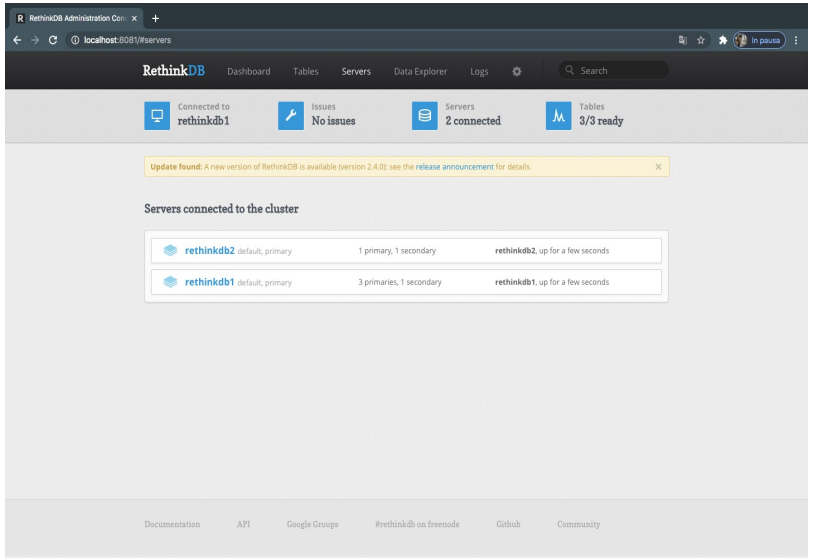
Progettazione server

Tecnologie utilizzate:
- Docker Desktop for windows
- immagine ufficiale *rethinkdb*

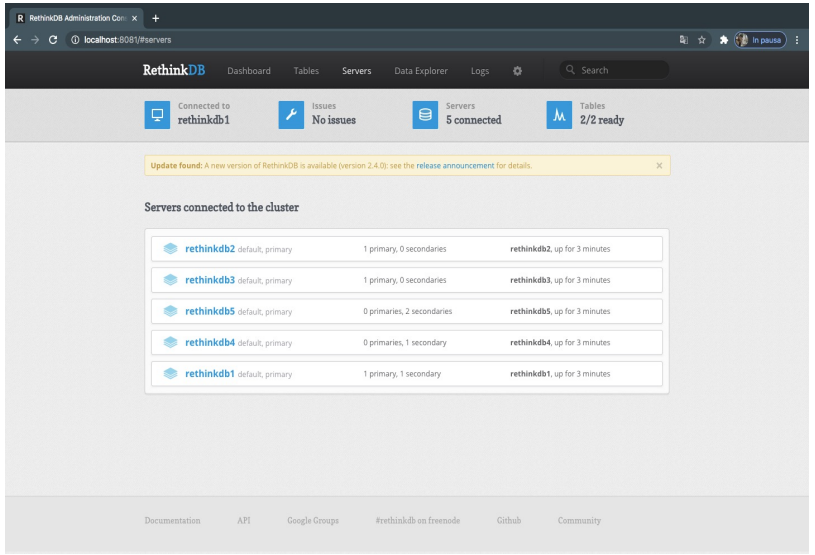
Server a singolo nodo:



Server a due nodi:



Server a cinque nodi



Conclusioni

- Libreria pubblicata come pacchetto NuGet su Visual Studio per i prodotti della società

possibili miglioramenti e sviluppi futuri:

• Espandere i nodi del server RethinkDB su più macchine affidandosi a Kubernetes o Docker Swarm