



UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

SISTEMAS DISTRIBUÍDOS EM LARGA ESCALA

# DECENTRALIZED TIMELINE

**AUTORES:**

BERNARDO MOTA (A77607)

FILIPPE NUNES (A78074)

LUÍS NETO (A77763)

4 de Setembro de 2019

# 1 Introdução

O presente documento tem como finalidade justificar as escolhas do grupo na realização deste projeto. Este projeto foi proposto na unidade curricular de Sistemas Distribuídos em Larga Escala e consiste na implementação de um serviço de *timeline* descentralizado, baseado numa arquitetura *peer-to-peer* com dispositivos periféricos.

Atualmente, o serviço de *timeline* pode ser visto com um maior grau de importância, pois é utilizado cada vez mais na maioria das redes sociais. Este serviço corresponde à organização de vários conteúdos do interesse e gosto de um dado utilizador, ou seja, a informação que a rede social achar mais pertinente para o utilizador terão prioridade na *timeline*. Assim, o principal objetivo desta é a organização cronológica das informações publicadas no perfil de um utilizador, numa rede social.

Ao longo do presente documento será apresentado de forma mais detalhada cada um dos componentes do sistema e será descrita a sua função, tecnologias utilizadas e a justificação das mesmas.

## 2 Estratégia utilizada

De modo a traçar o plano de trabalho, teve de se definir quais as escolhas necessárias a serem feitas. De entre elas, a mais importante foi a de definir o protocolo de comunicação entre os diferentes dispositivos, sendo utilizado o ***Spread***.

O ***Spread*** é um conjunto de ferramentas que providencia um serviço de mensagens de alto desempenho, flexível a faltas numa rede. Funciona, também, como um *bus* de mensagens únicas para aplicações distribuídas, providencia um *multicast* ao nível da aplicação e suporte para a comunicação em grupo. Para além disso, dá garantias que as mensagens trocadas entre os dispositivos serão sempre entregues e de forma completamente ordenada, mesmo que haja uma falha num dispositivo.

Assim, devido ao seu *multicast* eficiente, ao facto da comunicação ser por grupos e à existência de vistas, este conjunto de ferramentas é um forte aliado para o desenvolvimento e gestão das diferentes *timelines*.

Para além disso, esta combinação de características permite que um conjunto de utilizadores que subscrevam a *timeline* de outro utilizador, ou seja, utilizadores que pertençam ao mesmo grupo, consigam enviar a *timeline* completa do utilizador ao qual fizeram a subscrição, até quando este está desligado.

Por fim, de forma a garantir que os dados dos diferentes utilizadores possam ser enviados e lidos em máquinas diferentes, recorreu-se ao ***Atomix*** para fazer a sua serialização.

### 3 Arquitetura do Sistema

A seguir será apresentada a arquitetura do sistema desenvolvido, assim como a descrição de cada componente do sistema e de algumas decisões de implementação.

Cada cliente do sistema executa a classe **Timeline**, que lida com a interação do cliente com a rede descentralizada e recebe as interações dos outros clientes. Ao introduzir o seu nome de utilizador, o cliente passa a poder introduzir *posts* e comandos e passa a receber mensagens dos outros utilizadores a quem subscreveu. Cada cliente tem um **SpreadGroup** associado ao seu nome de utilizador que é utilizado para difundir a *timeline* do cliente, caso este esteja ligado ou não. Todos os utilizadores que estejam ligados e sigam um dado utilizador estão no **SpreadGroup** deste.

O programa permite publicar *posts*, seguir e deixar de seguir utilizadores, publicar *posts* temporários e apresentar os *posts* de um dado utilizador que segue.

Ao fazer um *post* de texto, este é adicionado à sua *timeline* e é transmitido a todos os seus seguidores, através de um *multicast* para o seu **SpreadGroup**.

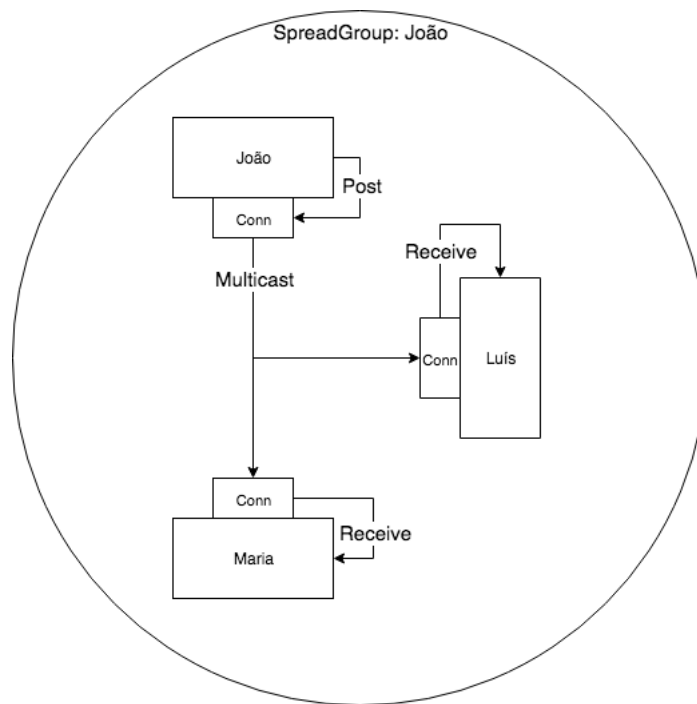


Figura 1: Envio de uma publicação do utilizador João para os seus subscritores

A subscrição a outros utilizadores é feita através do nome de utilizador. Ao subscrever um utilizador, o cliente passa a armazenar os seus *posts* e a fazer parte do seu **SpreadGroup**, recebendo as mensagens do utilizador e mantendo uma *view* do grupo. Ao deixar de seguir um utilizador, os seus *posts* são removidos localmente e o grupo é abandonado.

A receção dos *posts* dos utilizadores que o cliente segue e das mensagens de *membership* do *Spread*, necessárias para manter as *views*, é feita por uma *thread* que processa as mensagens recebidas da conexão *Spread*, apresentando o conteúdo caso seja um *post* e atualizando a *view* do grupo correspondente caso seja mensagem de *membership*.

Ao seguir um utilizador, o cliente recebe a sua *timeline* completa. Visto que uma *timeline* pode conter uma carga considerável de publicações, foi decidido que a carga de transmissão desta deve ser partilhada entre todos os utilizadores presentes no *Spread-Group*, ou seja, todos os utilizadores que seguem o utilizador em questão e estão ligados ao serviço. Esta divisão de trabalho é feita através da segmentação da *timeline* completa, que é dividida de acordo com o número de clientes no grupo. Através da *view* do grupo mantida consistentemente por cada cliente, é possível ordenar os clientes e atribuir a cada um dos segmentos para transmitir.

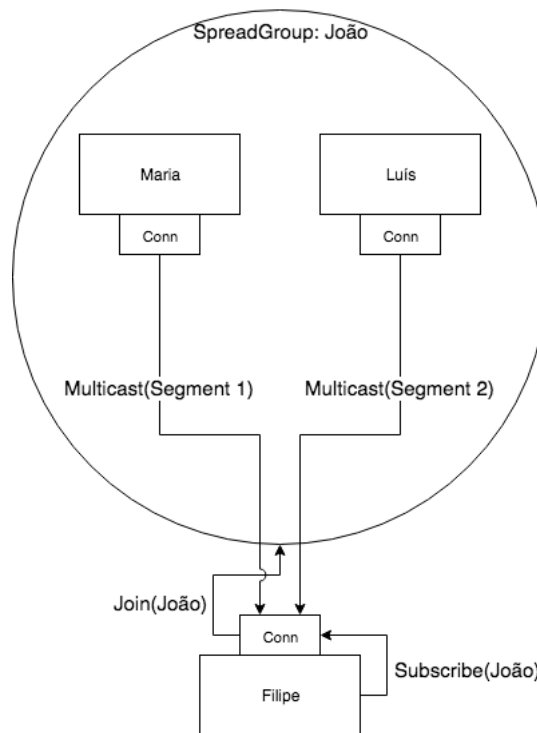


Figura 2: Subscrição do utilizador Filipe para o João e transmissão da sua *timeline* completa por parte dos subscritores ligados

Ao enviar o seu segmento, cada cliente envia o número de clientes online no grupo para verificação por parte do cliente que subscreveu.

Ao receber os segmentos, o cliente que fez a subscrição, armazena-os e conta o número de segmentos que recebeu, comparando este com o número de clientes online no grupo. Após receber todos os segmentos, é apresentada a *timeline* completa.

Para além de dividir a carga da transmissão de *timelines* completas, os seguidores de um dado utilizador armazenam e transmitem as publicações deste sempre que o utilizador

não está ligado ao serviço. Isto é, a transmissão dos *posts* do utilizador é feita caso o próprio utilizador ou outro utilizador que o siga estejam ligados. Visto que no caso do próprio utilizador não estar ligado a única informação a ser transmitida é para novos subscritores, o funcionamento da retransmissão é análogo à transmissão de uma *timeline* completa.

O utilizador pode ainda publicar *posts* temporários, ou seja, publicações que tenham uma data de expiro e que, após essa data, deixam de ser transmitidos e lidos. Para fazer um *post* temporário, o utilizador deve introduzir o tempo de vida da publicação e o seu conteúdo. Para garantir que um *post* temporário com tempo de vida ultrapassado não é transmitido ou lido, ao transmitir cada publicação ou a ler uma *timeline*, é verificado o tempo de expiro para cada publicação.

Finalmente, o cliente pode ler a *timeline* de cada utilizador que segue, apresentando todos os *posts* armazenados localmente de um dado utilizador.

## 4 Conclusão

O desenvolvimento deste projeto permitiu, através da implementação de várias técnicas, reunir e consolidar os conceitos acerca de sistemas que usam arquiteturas *peer-to-peer* para a transmissão de dados, neste caso sistemas de *timeline*.

O maior desafio que se teve de ultrapassar foi o facto de a tecnologia *Spread* utilizar *multicast* no envio dos novos *posts* feitos para o grupo, algo que não é baseado numa arquitetura *peer-to-peer* tal como era suposto. A maneira de resolver esse problema foi a distribuição de trabalho no envio da *timeline* para os utilizadores novos, através dos segmentos que são enviados pelos restantes utilizadores do grupo.