

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

Faculdade de Tecnologia Rubens Lara

**Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas**

HUGO BESSA SILVA DE OLIVEIRA

**SISTEMA DE EDIÇÃO COLABORATIVA POR P2P
UTILIZANDO PARADIGMA DE PROGRAMAÇÃO
FUNCIONAL**

**Santos, SP
2016**

HUGO BESSA SILVA DE OLIVEIRA

**SISTEMA DE EDIÇÃO COLABORATIVA POR P2P
UTILIZANDO PARADIGMA DE PROGRAMAÇÃO
FUNCIONAL**

Pré-Projeto de Desenvolvimento apresentada à Faculdade de Tecnologia Rubens Lara, como exigência para a obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Alexandre Garcia de Oliveira

Santos, SP

2016

SUMÁRIO

1. INTRODUÇÃO	3
1.1. OBJETIVO GERAL	5
1.1.1.OBJETIVOS Específicos	5
1.2. ORGANIZAÇÃO.....	5
2. REVISÃO BIBLIOGRÁFICA.....	6
2.1. DISPOSITIVOS MÓVEIS	6
2.2. SMARTPHONES.....	6
2.3. CONFLICT-FREE REPLICATED DATA TYPES.....	6
2.4. LOGOOT	7
2.5. ARQUITETURA DE REDE PEER-TO-PEER.....	7
2.6. CLOJURESCRIPT	8
3. METODOLOGIA.....	8
3.1. CASOS DE USO.....	9
3.1.1. Diagrama de caso de uso	9
4. REFERÊNCIAS BIBLIOGRÁFICAS	10

1. INTRODUÇÃO

A difusão de dispositivos móveis leva, ano após ano, o acesso à internet para cada vez mais pessoas. "Em 2015 existem mais de 7 bilhões de assinaturas de celulares móveis em todo o mundo, partindo de menos de 1 bilhão em 2000." (SANOU; BRAHIMA, 2015, p. 1, tradução nossa).

Pela primeira vez, mais de 2 bilhões de pessoas tiveram acesso à rede mundial de computadores. Isso torna possível a utilização de *softwares online*, como clientes de *e-mail*, sistemas de gerenciamento de conteúdo e sistemas de edição colaborativa de documentos. Disponibilizadas na internet, estas aplicações podem ser utilizadas em diferentes tipos de dispositivos e conectar usuários distantes fisicamente.

Algumas aplicações distribuídas utilizam o modelo de computação cliente-servidor, também conhecida como cliente-servidor de três níveis. Nesta arquitetura, todos os clientes, solicitantes de um recurso, se conectam e se comunicam através de um servidor, fornecedor de recursos.

Segundo Donald Bales (2003, p. 133, tradução nossa) "Uma arquitetura de três níveis adiciona ao modelo de dois níveis uma nova camada que isola o processamento de dados em um lugar centralizado [...]". Este isolamento traz importantes vantagens para aplicações distribuídas. Entretanto, este modelo computacional apresenta problemas graves quando utilizado em redes de internet móveis.

Redes de internet móveis são geralmente mais lentas do que redes banda-larga cabeadas. Além disso, a cobertura geográfica e a qualidade da conexão variam drasticamente entre regiões — incluindo locais não muito distantes. Dados da OpenSignal (2016) mostram uma latência média de 172 milissegundos em redes 3G. O modelo de comunicação cliente-servidor fica, então, prejudicado: com a indisponibilidade da rede e grande latência para comunicação, a experiência em aplicações que dependem da interação constante com o servidor é degradada.

De acordo com Shapiro et al. (2011) modelos de Forte Consistência Eventual (FCE) garantem melhor disponibilidade e performance em sistemas distribuídos, principalmente em redes com grande latência. Isto é alcançado por atualizações

serem realizadas em réplicas locais dos dados e eventualmente sincronizadas com outras réplicas dentro do sistema. Uma estrutura de dados compatível com as exigências de FCE é a *Conflict-free Replicated Data Type* (CRDT, em português Tipo de Dados Replicados Livre de Conflitos).

CRDTs garantem a consistência dos dados pois operações realizadas nesta estrutura são comutativas¹. CRDTs baseadas em operações transmitem seu estado propagando operações de atualização realizadas. Estas operações também devem ser comutativas. Para remover a necessidade de manter garantias de entrega única das mensagens de operação, as funções de operações de atualização também devem ser idempotentes² e associativas³.

Com as garantias de uma CRDT com operações associativas, comutativas e idempotentes, é possível construir um sistema em uma arquitetura de rede em que as réplicas se comunicam diretamente e propagam atualizações de forma distribuída e assíncrona. Esta arquitetura de rede se chama *peer-to-peer* (P2P), onde cada participante da rede atua tanto como cliente (requisitando dados) quanto como servidor (servindo dados) (SCHOLLMEIER; Rüdiger, 2002).

O presente trabalho utiliza a CRDT Logoot para implementar um Sistema de Edição Colaborativa distribuído com comunicação por uma rede *peer-to-peer*. A Logoot utiliza simples identificadores de posição para atingir comutatividade e associatividade em operações de inserção e remoção, que podem ser executadas em qualquer réplica participante do sistema sem alterar a ordem das linhas do documento (WEISS; URSO; MOLLI, 2008, p. 5).

Com o fim de facilitar a implementação da estrutura de dados Logoot, da interface de rede e do editor de texto necessários para o Sistema de Edição Colaborativa a ser desenvolvido como estudo de caso deste projeto, ClojureScript foi a escolhida. ClojureScript é uma linguagem de programação que segue o paradigma funcional, e o projeto tira proveito de suas estruturas de dados imutáveis e *atoms* (uma forma de gerenciar estado compartilhado).

¹ "Dois elementos x e y de um conjunto S são comutativos sob uma operação $*$ se eles satisfazem $x * y = y * x$ " (WEISSTEIN; ERIC, tradução nossa)

² Uma operação é idempotente quando a sua aplicação repetida tem o mesmo efeito que uma (FOLDOC, 1995, tradução nossa)

³ "[...] associativa: $a(bc) = (ab)c$ para todo $a, b, c \in G$ " (HUGENFORD; Thomas, 1974, p. 24, tradução nossa)

1.1. OBJETIVO GERAL

Desenvolver um sistema de edição colaborativa Logoot em um editor de texto com comunicação *peer-to-peer* (P2P) utilizando a linguagem de programação ClojureScript.

1.1.1.OBJETIVOS ESPECÍFICOS

Desenvolver módulo ClojureScript que implemente o sistema Logoot e permita todas as suas funcionalidades: adição, remoção e edição de conteúdo; Implementar módulo ClojureScript que transforme a estrutura de dados Logoot em um editor de texto capaz de realizar operações de adição, remoção e edição de conteúdo no sistema; Implementar infraestrutura de rede pra suportar a Edição Colaborativa por P2P utilizando a tecnologia WebRTC; Demonstrar a utilização de uma linguagem funcional para trabalhar com estruturas de dados complexas e compor pequenas partes do *software* em um sistema completo.

1.2. ORGANIZAÇÃO

A introdução do tema do trabalho está no primeiro capítulo. Nela temos a apresentação do tema do trabalho seus objetivos.

No segundo capítulo a revisão bibliográfica é apresentada — um detalhamento de cada tecnologia utilizada para solucionar o problema apresentado. Estão neste capítulo assuntos como arquitetura de rede ponto-a-ponto, smartphones, Logoot e ClojureScript.

2. REVISÃO BIBLIOGRÁFICA

2.1. DISPOSITIVOS MÓVEIS

Segundo o Dicionário Cambridge (tradução nossa), dispositivos móveis são "qualquer peça de equipamento eletrônico como um telefone móvel ou pequeno computador que pode ser utilizado em lugares diferentes". Os dispositivos móveis estão levando os computadores e a internet a lugares com pouca tecnologia infraestrutura de comunicação, como visto na pesquisa da ICT Data and Statistics Division.

Dispositivos móveis mais avançados são atualmente chamados *smartphones*.

2.2. SMARTPHONES

O avanço da tecnologia desenvolvida durante décadas para dispositivos móveis resultou na miniaturização do poder de processamento necessário para executar um sistema operacional avançado, capaz de executar múltiplas tarefas simultâneas e executar aplicativos multimídia. Como definido pelo PhoneScoop (tradução nossa):

Uma categoria de dispositivos móveis que oferecem capacidades avançadas além de um típico telefone móvel. *Smartphones* rodam um sistema operacional completo que oferecem interfaces e plataformas padronizadas para desenvolvedores de aplicativos.

Este avanço em poder de processamento tornou capaz que aplicativos baseados na web pudessem ser mais completos e bastante interativos. Mesmo os modelos de baixo-custo são capazes de executar Sistemas de Edição Colaborativa, por exemplo.

2.3. CONFLICT-FREE REPLICATED DATA TYPES

Sistemas distribuídos possuem diversas réplicas que compartilham a mesma estrutura de dados. A replicação de dados nestas circunstâncias exige uma abordagem diferenciada, já que a consistência em um sistema distribuído é tão importante quanto em um sistema centralizado. Modelos de Consistência Eventual

(CE) são adequados, já que permitem que estes sistemas escalem e mantenham sua performance. Como escrito por Shapiro et al. (2011):

Replicar dados sob Consistência Eventual (CE) permitem que qualquer réplica aceite atualizações sem sincronização remota. [...] Entretanto, abordagens de CE já publicadas são ad-hoc e suscetíveis a erros. Sob um modelo formal de Forte Consistência eventual, nós estudamos condições suficientes para convergência. Um tipo de dados que satisfaz estas condições é chamado Conflict-free Replicated Data Type (CRDT).

Este tipo de dados é adequado para a implementação de um sistema de edição colaborativo que funciona de forma distribuída em uma arquitetura de rede sem fortes garantias de entrega de mensagens a sem a necessidade de servidor centralizado para sincronizar versões de um mesmo documento.

2.4. LOGOOT

Logoot é uma estrutura de dados criada para ser utilizada em sistemas de edição colaborativa em redes distribuídas ponta-a-ponta. Ela é também uma CRDT, tornando possível que várias cópias de um mesmo documento sejam alteradas e sincronizadas sem conflito (WEISS; URSO; MOLLI, 2008, p. 5, tradução nossa).

Uma vantagem da Logoot quando comparada com outras estruturas de dados como Wooki (WEISS; URSO; MOLLI, 2007) é que ela não adiciona um grande peso em cima dos dados brutos de um documento devido à baixa quantidade de metadados necessária para mantê-la consistente entre sincronizações com réplicas distribuídas.

2.5. ARQUITETURA DE REDE *PEER-TO-PEER*

Definida por Schollmeier apud Kellerer, uma arquitetura de rede é peer-to-peer (P2P) se seus participantes agem tanto como clientes (requisitando recursos) quanto servidores (fornecendo recursos), utilizando seus próprios recursos de hardware que são acessados diretamente por outros participantes da rede sem entidades intermediárias.

Este estilo de arquitetura ficou popularizada com o *software* de compartilhamento de arquivos de áudio Napster (SCHOLLMEIER, 2002, p. 1, tradução nossa). No

presente trabalho, esta arquitetura soluciona o problema da necessidade de um servidor centralizado para sincronizar réplicas, transformando cada usuário em uma réplica do sistema de edição colaborativa.

2.6. CLOJURESCRIPT

A linguagem utilizada para implementar os algoritmos e sistemas necessários para este projeto é Clojure, utilizando o compilador ClojureScript. ClojureScript é um compilador criado para emitir código JavaScript — que pode ser executado em navegadores modernos (CLOJURE).

Clojure é uma linguagem de programação de alto-nível dinâmica de propósito geral, criada por Rich Hickey (CLOJURE). Ela torna abordável conceitos como programação funcional e programação multithread. "Clojure é um dialeto de Lisp, e compartilha com Lisp a filosofia de que código é dado e um poderoso sistema de macros" (CLOJURE).

Neste projeto, as estruturas de dados imutáveis e sistemas de gerenciamento de estado de Clojure possibilitam uma implementação enxuta e densa, menos propensa a problemas de concorrência.

3. METODOLOGIA

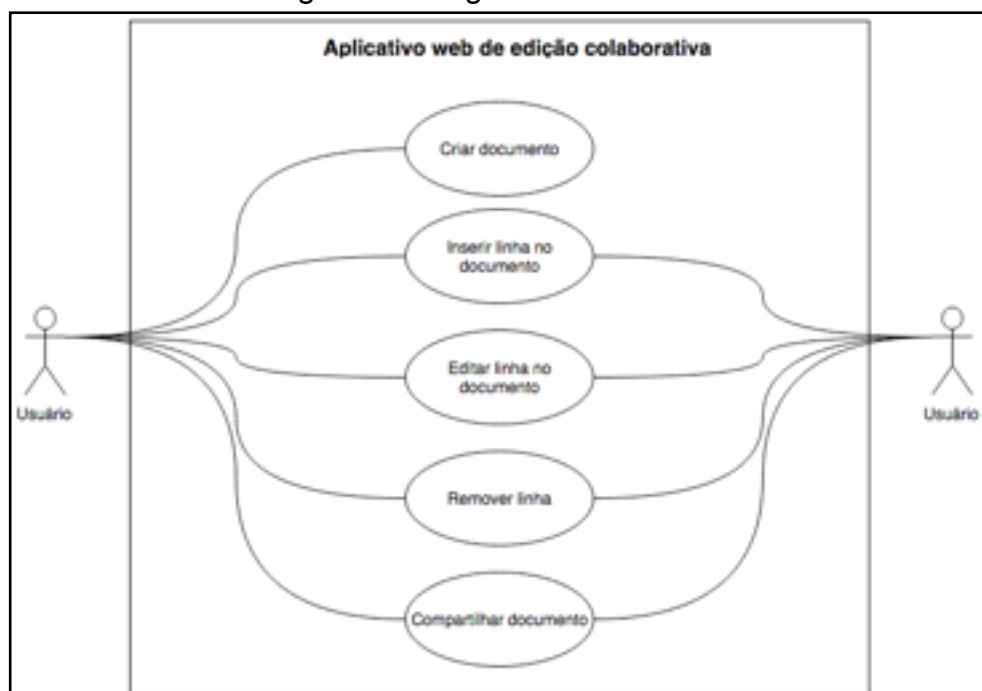
O presente trabalho foi desenvolvido utilizando a metodologia ágil de desenvolvimento de *software*. Segundo Beck et al. (2001), o desenvolvimento ágil valoriza "indivíduos e interações sobre processos e ferramentas, *software* funcional sobre documentação abrangente, colaboração de clientes sobre negociação de contratos, responder a mudanças sobre seguir um plano". Estes valores realçam a importância de *software* desenvolvido, testado e executado acima outros objetivos como a presença de uma documentação extensiva — esta que frequentemente não é atualizada frequentemente pelos participantes do projeto.

Projetos que adotam metodologias de desenvolvimento ágil também possuem documentação, embora possivelmente muito menos do que projetos que usam metodologias mais tradicionais baseadas em grandes quantidades de documentos e planejamento.

3.1. CASOS DE USO

3.1.1. DIAGRAMA DE CASO DE USO

Figura 1 - Diagrama de caso de uso



Fonte: OLIVEIRA, 2016

4. REFERÊNCIAS BIBLIOGRÁFICAS

ICT Data and Statistics Division. **ICT Facts & Figures 2015**. Maio de 2015. Disponível em: <<http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf>>. Acesso em: 10 de abril de 2016.

BALES, Donald. **JDBC Pocket Reference**. julho de 2003. Disponível em: <<https://web.archive.org/web/20061212141647/http://java.sun.com/developer/Books/jdbc/ch07.pdf>>. Acesso em: 29 de abril de 2016.

OpenSignal. **O Estado das Redes Móveis: Brasil (fevereiro 2016)**. fevereiro de 2016. Disponível em: <http://opensignal.com/reports/auto/data-2016-02-brazil/report_pt.pdf>. Acesso em: 16 de maio de 2016.

SHAPIRO, Marc; PREGUIÇA, Nuno; BAQUERO, Carlos; ZAWIRSKI, Marek. **Conflict-free Replicated Data Types**. 19 de julho de 2011. Disponível em: <<https://hal.inria.fr/inria-00609399v1/document>>. Acesso em: 10 de abril de 2016.

WEISSTEIN, Eric. **Commutative**. Disponível em: <<http://mathworld.wolfram.com/Commutative.html>>. Acesso em: 16 de abril de 2016.

FOLDOC. **Idempotent**. 11 de janeiro de 1995. Disponível em: <<http://foldoc.org/idempotent>>. Acesso em: 16 de abril de 2016.

HUNGERFORD, Thomas. **Algebra**. 1 ed. Cleveland: Springer, 1974.

SCHOLLMEIER, Rüdiger. **A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications**. 2002. Disponível em <<http://www.computer.org/csdl/proceedings/p2p/2001/1503/00/15030101.pdf>>. Acesso em: 10 de abril de 2016.

WEISS, Stéphane; URSO, Pascal; MOLLI, Pascal. **Logoot: a P2P collaborative editing system**. 10 de dezembro de 2008. Disponível em: <<https://hal.inria.fr/inria-00336191/PDF/main.pdf>>. Acesso em: 11 de abril de 2016.

PHONESCOOP. **Smartphone**. Disponível em: <<http://www.phonescoop.com/glossary/term.php?gid=131>>. Acesso em: 16 de abril de 2016

WEISS, Stéphane; URSO, Pascal; MOLLI, Pascal. **Wooki: a P2P Wiki-based Collaborative Writing Tool**. 25 de junho de 2007. Disponível em: <<https://hal.inria.fr/inria-00156190v2/document>>. Acesso em: 13 de junho de 2016.

CLOJURE. **The Clojure Programming Language**. Disponível em: <<http://clojure.org/>>. Acesso em: 13 de junho de 2016.

BECK, Kent; BEEDLE, Mike; BENNEKUM, Arie van; COCKBURN, Alistair; CUNNINGHAM, Ward; FOWLER, Martin; GRENNING, James; HIGHSMITH, Jim; HUNT, Andrew; JEFFRIES, Ron; KERN, Jon; MARICK, Brian; MARTIN, Robert C.; MELLOR, Steve; SCHWABER, Ken; SUTHERLAND, Jeff; THOMAS, Dave. **Manifesto for Agile Software Development**. Disponível em: <<http://agilemanifesto.org/iso/en/>>. Acesso em: 14 de junho de 2016.