



NOME COMPLETO DO AUTOR

Licenciado em Engenharia Informática

SINCRONIZAÇÃO DE DOCUMENTOS NA EDIÇÃO COLABORATIVA EM TEMPO REAL

MESTRADO EM ENGENHARIA INFORMÁTICA

Universidade NOVA de Lisboa
Outubro, 2026

SINCRONIZAÇÃO DE DOCUMENTOS NA EDIÇÃO COLABORATIVA EM TEMPO REAL

NOME COMPLETO DO AUTOR

Licenciado em Engenharia Informática

Orientador: Nome do Orientador

Professor Auxiliar, Universidade NOVA de Lisboa

Coorientadora: Nome da Coorientadora

Professora Associada, Universidade NOVA de Lisboa

Júri

Presidente: Nome do Presidente

Professor Catedrático, Universidade NOVA de Lisboa

Arguente: Nome da Arguente

Professora Auxiliar, Universidade de Lisboa

Orientador: Nome do Orientador

Professor Auxiliar, Universidade NOVA de Lisboa

MESTRADO EM ENGENHARIA INFORMÁTICA

Universidade NOVA de Lisboa

Outubro, 2026

Sincronização de documentos na edição colaborativa em tempo real

Copyright © Nome Completo do Autor, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa detêm o direito perpétuo e sem limites geográficos de arquivar e publicar esta dissertação em formato impresso ou digital, ou por qualquer outro existente ou que venha a ser criado, bem como de a divulgar através de repositórios científicos e de permitir a sua cópia e distribuição para fins educacionais ou de investigação, não-comerciais, desde que seja dado o devido crédito ao autor e ao editor.

AGRADECIMENTOS

Os agradecimentos são um texto pessoal e uma expressão livre do autor. Começam habitualmente pelos agradecimentos académicos (orientadores), seguem-se os institucionais (centro de investigação, departamento, faculdade, bolsas) e, por fim, os pessoais (amigos, família).

RESUMO

O resumo apresenta em poucos parágrafos o contexto do trabalho, o problema abordado, a abordagem seguida e os principais resultados obtidos. Não deve ultrapassar uma página.

O segundo parágrafo pode indicar o alcance dos resultados, as suas limitações e as perspectivas de trabalho futuro.

Palavras-chave: Primeira palavra-chave · Segunda palavra-chave · Terceira palavra-chave

ABSTRACT

The abstract presents in a few paragraphs the context of the work, the problem addressed, the approach taken and the main results. It should not exceed one page.

The second paragraph may state the scope of the results, their limitations and the perspectives for future work.

Keywords: First keyword · Second keyword · Third keyword

ÍNDICE

Resumo	iv
Abstract	v
Índice	vi
Índice de Figuras	vii
Índice de Tabelas	viii
Introdução	1
1 Edição colaborativa de documentos	2
1.1 Requisitos do sistema	3
2 Abordagens à sincronização	4
2.1 Transformação operacional	4
2.1.1 Função de transformação	4
2.2 Tipos de dados replicados livres de conflitos	4
3 Conceção e implementação	6
3.1 Armazenamento do histórico	6
4 Avaliação	7
4.1 Discussão dos resultados	8
Conclusão	9
Referências bibliográficas	10

ÍNDICE DE FIGURAS

1.1	Decurso típico da escrita colaborativa de um documento	2
4.1	Número de alterações por dia da semana e hora nas sessões analisadas . . .	8

ÍNDICE DE TABELAS

4.1	Latência média e dimensão dos dados armazenados consoante o número de utilizadores	7
-----	--	---

INTRODUÇÃO

Hoje em dia, os documentos de texto raramente são obra de um único autor. Os estudantes redigem os seus trabalhos de seminário em pares, os programadores mantêm em conjunto a documentação técnica e as equipas de investigação concluem os seus artigos à última hora a partir de vários locais em simultâneo. Para que este tipo de escrita funcione, o editor tem de garantir que todos os participantes veem o mesmo texto e que nenhuma alteração se perde.

O objetivo deste trabalho é comparar as duas principais abordagens à sincronização de documentos na edição colaborativa em tempo real, a transformação operacional e os tipos de dados replicados livres de conflitos, e validá-las num protótipo de editor web. O primeiro capítulo delimita os requisitos de um sistema deste tipo, o segundo descreve ambas as abordagens, o terceiro a conceção do protótipo e o quarto os resultados das medições.

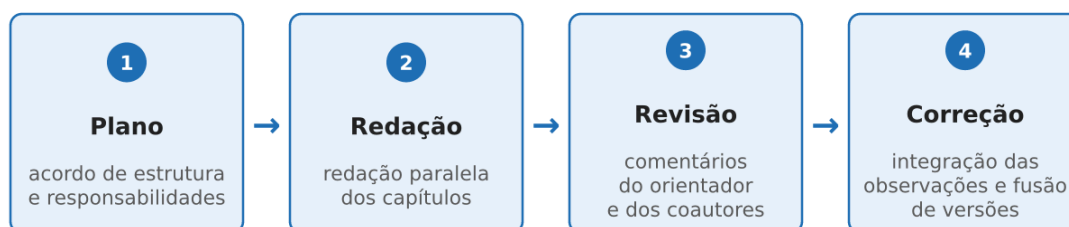
EDIÇÃO COLABORATIVA DE DOCUMENTOS

Por edição colaborativa entendemos a situação em que vários utilizadores alteram o mesmo documento e as respetivas alterações se propagam mutuamente sem fusão manual. Consoante a rapidez com que as alterações se propagam, distinguimos três modos de trabalho fundamentais:

- **edição assíncrona**, na qual os autores trocam entre si versões completas do documento e só as fundem posteriormente;
- **edição com bloqueios**, em que apenas um autor de cada vez pode alterar uma determinada parte do documento;
- **edição em tempo real**, na qual cada alteração se reflete imediatamente para os restantes utilizadores.

Este trabalho ocupa-se exclusivamente do último modo. É o mais cómodo para os utilizadores, mas é o que impõe maiores exigências à conceção do sistema, uma vez que as alterações surgem em simultâneo em muitos locais e chegam por ordens diferentes.

A edição em tempo real não altera, aliás, apenas a técnica, mas também o decurso de toda a colaboração. O processo típico de escrita colaborativa, tal como foi descrito pelos utilizadores entrevistados, é apresentado na [Figura 1.1](#).



os passos 2 a 4 repetem-se enquanto a revisão trouxer observações substanciais

Figura 1.1: Decurso típico da escrita colaborativa de um documento

1.1 Requisitos do sistema

A partir de entrevistas com utilizadores de editores existentes e da literatura da especialidade, derivámos quatro requisitos que um sistema de edição colaborativa tem de cumprir:

1. convergência, ou seja, após a entrega de todas as alterações, todas as cópias do documento têm o mesmo conteúdo;
2. preservação da intenção, ou seja, o resultado de uma alteração corresponde ao que o autor via no momento em que a fez;
3. capacidade de resposta, ou seja, uma alteração local reflete-se imediatamente, sem esperar pelo servidor;
4. robustez, ou seja, uma breve falha da ligação não leva à perda do texto que está a ser escrito.

Os dois primeiros requisitos dizem respeito à correção e os dois restantes à usabilidade. Na prática, verifica-se que é precisamente a combinação de resposta imediata e convergência garantida a parte mais difícil de toda a conceção.

ABORDAGENS À SINCRONIZAÇÃO

Os sistemas de edição colaborativa em tempo real aplicam primeiro cada alteração localmente e só depois a difundem pelos restantes participantes. Distinguem-se sobretudo pela forma como tratam a situação em que duas alterações surgem em simultâneo e cada participante as recebe por uma ordem diferente.

2.1 Transformação operacional

A transformação operacional baseia-se no trabalho de Ellis e Gibbs [1], que foram os primeiros a descrever como ajustar uma operação concorrente de modo a que possa ser executada mesmo depois de outra operação que o seu autor desconhecia. Cada alteração é registada como uma operação de inserção ou de eliminação numa determinada posição do texto.

Se, por exemplo, um utilizador inserir uma palavra no início de um parágrafo e outro eliminar em simultâneo uma palavra no seu final, a posição da eliminação tem de ser deslocada no comprimento da palavra inserida. Este deslocamento é calculado pela função de transformação.

2.1.1 Função de transformação

A função de transformação recebe um par de operações concorrentes e devolve as respectivas versões ajustadas. Para que todas as cópias do documento convirjam, a função tem de satisfazer as chamadas propriedades de transformação, cujo cumprimento é difícil de demonstrar para operações gerais. Por isso, a maioria dos sistemas implementados assenta num servidor central que determina uma ordem única das operações.

2.2 Tipos de dados replicados livres de conflitos

A segunda abordagem é constituída pelos tipos de dados replicados livres de conflitos, designados coletivamente pela sigla CRDT [2]. Em vez de ajustar as operações posteri-

ormente, opta por uma representação do documento na qual as operações concorrentes comutam e a sua ordem não tem influência no resultado.

Um tipo de dados replicado é livre de conflitos se quaisquer duas réplicas que tenham recebido o mesmo conjunto de atualizações tiverem o mesmo estado, independentemente da ordem pela qual estas lhes foram entregues.

Cada carácter do documento recebe um identificador único que se mantém inalterado durante toda a existência do carácter. A posição de inserção deixa assim de ser determinada por um número e passa a sê-lo por uma referência ao identificador do carácter vizinho, pelo que as alterações concorrentes noutras partes do texto não a podem deslocar.

CONCEÇÃO E IMPLEMENTAÇÃO

O protótipo do editor é composto por um cliente web, um servidor de sincronização e uma base de dados na qual o servidor guarda continuamente o estado dos documentos. Tanto o cliente como o servidor estão escritos de forma a permitir substituir a camada de sincronização e comparar ambas as abordagens descritas sobre a mesma interface de utilizador.

O cliente mantém uma réplica local do documento e converte cada alteração do utilizador numa operação sobre o tipo de dados replicado. As operações são enviadas ao servidor em lotes, para não sobrecarregar a rede durante uma escrita rápida. A listagem seguinte mostra o processamento simplificado de um lote recebido no servidor:

```
def processar_lote(documento, lote):  
    for operacao in lote.operacoes:  
        if operacao.id in documento.aplicadas:  
            continue  
        documento.aplicar(operacao)  
        documento.aplicadas.add(operacao.id)  
    difundir_aos_outros(documento.id, lote)
```

Como as operações comutam, o servidor não precisa de ordenar nem de transformar os lotes. Basta-lhe garantir que nenhuma operação é executada duas vezes e que cada lote chega a todos os clientes ligados.

3.1 Armazenamento do histórico

O servidor não guarda apenas o texto final, mas todo o registo de operações. A partir do registo, é possível restaurar qualquer versão anterior do documento e reconstituir o estado da réplica de um cliente que esteve desligado durante muito tempo. Esta abordagem corresponde a uma arquitetura baseada num registo imutável de eventos, tal como a descreve Kleppmann [3]. Para que o registo não cresça indefinidamente, o servidor compacta-o, a intervalos regulares, em instantâneos.

AVALIAÇÃO

Comparámos o protótipo com uma implementação de referência da transformação operacional num conjunto de registos de sessões de edição reais. Para cada sessão, medimos a latência, ou seja, o tempo decorrido entre uma alteração do lado do autor e a sua apresentação aos restantes participantes, e calculámos a latência média da sessão de acordo com (4.1):

$$\bar{t} = \frac{1}{n} \sum_{i=1}^n (t_i^{\text{entrega}} - t_i^{\text{alteração}}) \quad (4.1)$$

onde n é o número de operações na sessão. Os valores medidos para diferentes números de utilizadores a trabalhar em simultâneo são resumidos na [Tabela 4.1](#).

Tabela 4.1: Latência média e dimensão dos dados armazenados consoante o número de utilizadores

Utilizadores	Latência OT (ms)	Latência CRDT (ms)	Dados (kB)
2	38	35	12
5	52	41	19
10	87	48	31
20	164	63	58

À medida que o número de utilizadores aumenta, a latência da transformação operacional cresce muito mais depressa, porque o servidor tem de transformar cada operação relativamente a todas as operações concorrentes. O preço da menor latência dos CRDT é um maior volume de dados armazenados, uma vez que cada carácter transporta o seu identificador. A distribuição temporal das alterações, na qual se baseia o teste de carga, é apresentada na [Figura 4.1](#).

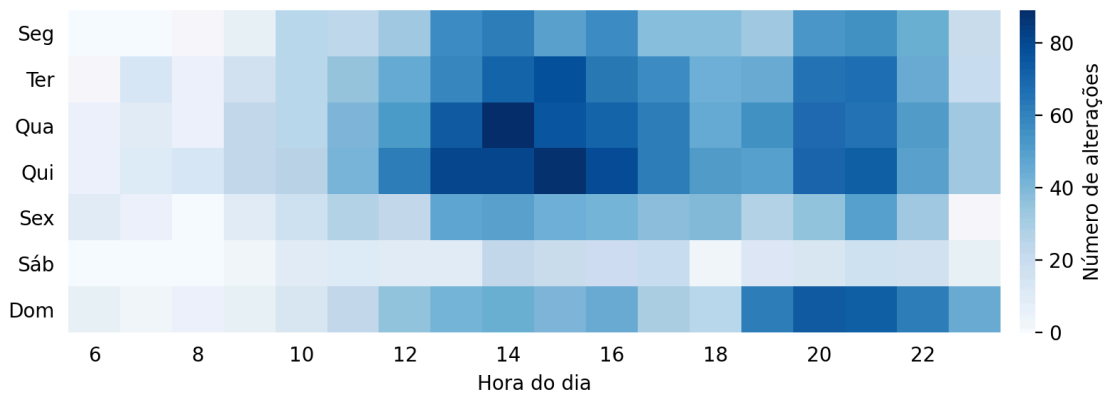


Figura 4.1: Número de alterações por dia da semana e hora nas sessões analisadas

4.1 Discussão dos resultados

Os resultados confirmam que, para equipas pequenas, ambas as abordagens são equivalentes e que o fator decisivo é sobretudo a simplicidade da implementação. Com um maior número de utilizadores simultâneos, os CRDT apresentam uma clara vantagem tanto na latência como na robustez face a falhas de ligação. Entrevistas complementares com utilizadores, conduzidas segundo os princípios da investigação qualitativa [4], mostraram que os utilizadores não percecionam uma diferença de latência inferior a cem milissegundos, mas consideram um problema grave a perda do texto que estava a ser escrito após uma falha de ligação.

CONCLUSÃO

Este trabalho comparou a transformação operacional e os tipos de dados replicados livres de conflitos enquanto duas abordagens à sincronização de documentos na edição colaborativa em tempo real. Com base num protótipo de editor web, mostrámos que os CRDT permitem um servidor mais simples, sem ordenação central das operações, e oferecem uma latência inferior com um maior número de utilizadores.

A principal desvantagem dos CRDT continua a ser o maior volume de dados armazenados. O próximo passo será, por isso, verificar de que forma a compactação periódica do registo em instantâneos influencia a dimensão dos dados e como se comporta o protótipo em funcionamento prolongado com dezenas de milhares de documentos.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] C. A. Ellis e S. J. Gibbs, «Concurrency control in groupware systems», em *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data*, New York: ACM, 1989, pp. 399–407. doi: [10.1145/67544.66963](https://doi.org/10.1145/67544.66963).
- [2] M. Shapiro, N. Preguiça, C. Baquero, e M. Zawirski, «Conflict-free replicated data types», em *Stabilization, Safety, and Security of Distributed Systems*, Berlin: Springer, 2011, pp. 386–400. doi: [10.1007/978-3-642-24550-3_29](https://doi.org/10.1007/978-3-642-24550-3_29).
- [3] M. Kleppmann, *Designing data-intensive applications*. Sebastopol: O'Reilly Media, 2017.
- [4] J. Hendl, *Kvalitativní výzkum: základní metody a aplikace*. Praha: Portál, 2005.

