

Web Colaborativa com Braid e CRDTs

João Oliveira

Orientação de: Nuno Preguiça

DI - Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa

`jfv.oliveira@campus.fct.unl.pt`

Abstract. Neste relatório exploramos as aplicações *web* colaborativas e a tecnologia que as suporta. É feita uma reflexão sobre as necessidades a nível tecnológico, tais como a falta de *standardização* e as dificuldades associadas. Por fim é exposta uma possível solução a estes problemas, através da construção de uma solução de sincronização de dados.

Keywords: colaboração, web

1 Introdução

Nos dias de hoje, as aplicações *web* colaborativas são uma parte cada vez mais importante da vida de qualquer pessoa. Desde a marcação de reuniões num calendário partilhado até à realização das próprias reuniões, que cada vez mais se fazem em ambientes online. Também estão presentes em ambientes menos formais, como na comunicação, através de aplicações de *chatting* ou redes sociais.

É evidente a popularidade deste tipo de aplicações. Na realidade, estão por todo o lado e englobam muito mais do que redes sociais e calendários partilhados. A colaboração deixou de ser um nicho e tornou-se o padrão em qualquer tipo de aplicação.

Neste relatório exploramos a tecnologia que está por detrás das aplicações *web* colaborativas e os seus defeitos.

2 Trabalho Relacionado

2.1 *Groupware* e Aplicações *Web* Colaborativas

Uma das primeiras definições de *software* colaborativo surgiu em 1991, com a designação de *groupware* (*group* + *software*) por Ellis et al.: "Especificamente, definimos *groupware* como: sistemas baseados em computadores que suportam grupos de pessoas envolvidas numa tarefa (ou objetivo) e que fornecem uma interface para um ambiente partilhado" [2].

Já nos dias de hoje, com esta área muito mais maturada, o contexto em que o *groupware* se encontra é diferente. Devido ao seu alcance demográfico, a *web* e os dispositivos moveis tornaram-se as plataformas primárias para difundir serviços de *groupware*. Como exemplo temos serviços como Google Docs, Trello, WhatsApp, Facebook, GitHub e muitos mais.

Sob este pretexto, podemos distinguir as atuais aplicações *web* colaborativas de outras aplicações *web* é o facto de existir a noção de um estado partilhado para o qual os utilizadores contribuem com um objetivo ou propósito em vista, de forma intencional ou não. Por exemplo no caso das redes sociais, o objetivo de um utilizador é comentar e trocar impressões, com um ou mais utilizadores, sobre um tema. Com esse objetivo em vista o utilizador contribui, muitas vezes não intencionalmente, para um registo público semi-permanente.

2.2 Popularidade das Aplicações Web Colaborativas

Mesmo na enorme piscina que são as aplicações *web*, as colaborativas são de longe as mais usadas e que trazem mais pessoas para a *web*.

Um estudo feito em 2021, aponta para que 57%[3] da população europeia já tenha usado redes sociais. Já nos EUA, os valores apontam para tão alto quanto 70%[1]. Fora do mundo ocidental os números também não ficam muito a baixo, com a China e a Índia com valores¹ de 68% e 34%, respetivamente[4].

3 Estado da Arte

3.1 Problemas de *standardização* e o Protocolo Braid

Para as aplicações *web* colaborativas é essencial a sincronização de dados. Logo, a *standardização* dos protocolos de comunicação que são usados para sincronizar esses mesmos dados é importante para a interoperabilidade de sistemas.

Atualmente, a falta de um protocolo de sincronização geralmente aceite leva a que ao longo do desenho de uma aplicação colaborativa se recorra a modelos de comunicação *ad-hoc*, tal como comunicação por eventos² ou técnicas como *long-polling*.

O Protocolo Braid surge como resposta a este problema.

Segundo a própria equipa Braid o protocolo Braid é "um conjunto de extensões que generalizam o protocolo HTTP de um protocolo de transferência de estado para um protocolo de sincronização de estado [...] colocando o poder de OT e dos CRDTs na web, melhorando a performance de rede e permitindo aplicações web serem nativamente P2P, colaborativamente editáveis e *offline-first*" [6].

No fundo, é um protocolo de sincronização de dados que procura ser *backwards-compatible* e que está em desenvolvimento no contexto da IETF.

¹ É importante notar que o relatório citado usa o número de utilizadores bruto, ou seja, não filtra utilizadores falsos ou duplicados. Não obstante disso, com estas figuras podemos concluir que as redes sociais abrangem largas faixas de populações não ocidentais, visto o valor obtido para a Índia ser o mais baixo da região asiática e representar cerca de 1/3 da população.

² Por exemplo *socket.io* ou *web-sockets*

3.2 CRDTs

Outro aspeto a considerar é como os dados efetivamente são sincronizados.

Os "Tipos de dados Replicados sem Conflito", habitualmente denominados *CRDTs* (do inglês *Conflict Free Replicated Data Types*), são abstrações de tipos de dados desenhadas para permitirem a replicação de dados por múltiplos processos com coordenação assíncrona, através da troca de mensagens entre as réplicas e de métodos determinísticos[5].

Mais especificamente, os *CRDTs* garantem alta disponibilidade no acesso a dados replicados, consistência eventual e tolerância a falhas de rede.

Isto torna a tecnologia dos *CRDTs* muito atrativa, especialmente para um sistema que vise suportar aplicações *web* colaborativas distribuídas disponíveis *offline*. Há soluções que fazem uso desta tecnologia tanto no *backend*, com bases de dados replicadas de alta disponibilidade³, como no *frontend* das aplicações, onde se vêem certas partes de aplicações a serem replicadas por utilizadores através da tecnologia dos *CRDTs*.

3.3 Bibliotecas de CRDTs

De maneira a aplicar esta tecnologia ao mundo real, é necessário dotar os desenvolvedores de *software* de ferramentas que lhes permitam abstrair-se da complexidade do desenho da parte interna dos *CRDTs*. Para tal, existem inúmeras implementações de *CRDTs* que escondem a complexidade atrás de interfaces bem conhecidas, como listas, dicionários, árvores, etc., e interfaces de sincronização, estas que variam de implementação para implementação.

Para o teste da nossa solução iremos usar duas bibliotecas de *CRDTs* que ficam abaixo descritas:

Legion - Framework concebido em 2017 que permite a aplicações *web* replicar nos clientes, e de forma segura, dados presentes em servidores. Funciona à base de *deltas* (Δ -CRDT), que representam diferenças entre o estado interno de dois *CRDTs* e permitem otimizar o processo de sincronização.

Automerge - Biblioteca para sincronização de dados entre dispositivos móveis. Usa os seus próprios *CRDTs* para oferecer um modelo de dados de tipo *JSON*.

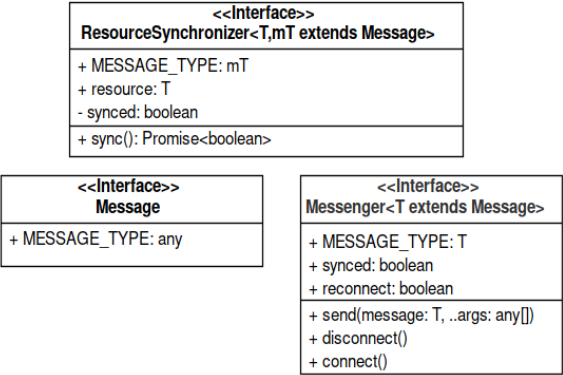
4 TejoSynchronizer

O *TejoSynchronizer* é um framework que procura facilitar o trabalho do desenvolvedor de aplicações *web* colaborativas, através de uma interface que *standardiza* a interação com soluções de sincronização específicas, como o Legion ou o Automerge.

Nomeadamente, esta solução especifica três componentes:

³ Ver Redis (<https://redis.com/>)

- **Sincronizador**: Componente associada à solução de sincronização. Implementa um protocolo de sincronização e interage com um ou mais mensageiros.
- **Mensageiro**: Permite a troca de mensagens entre sincronizadores.
- **Parser**: Faz a tradução de mensagens entre formatos. Serve, por exemplo, para converter mensagens de um Sincronizador para mensagens que podem ser enviadas pelo Mensageiro.



Sample Heading (Fourth Level) The contribution should contain no more than four levels of headings. Table 1 gives a summary of all heading levels.

Table 1. Table captions should be placed above the tables.

| Heading level | Example | Font size and style |
|-------------------|---|---------------------|
| Title (centered) | Lecture Notes | 14 point, bold |
| 1st-level heading | 1 Introduction | 12 point, bold |
| 2nd-level heading | 2.1 Printing Area | 10 point, bold |
| 3rd-level heading | Run-in Heading in Bold. Text follows | 10 point, bold |
| 4th-level heading | <i>Lowest Level Heading.</i> Text follows | 10 point, italic |

Displayed equations are centered and set on a separate line.F

$$x + y = z \tag{1}$$

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see Fig. ??).

Theorem 1. *This is a sample theorem. The run-in heading is set in bold, while the following text appears in italics. Definitions, lemmas, propositions, and corollaries are styled the same way.*

Proof. Proofs, examples, and remarks have the initial word in italics, while the following text appears in normal font.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [?], an LNCS chapter [?], a book [?], proceedings without editors [?], and a homepage [?]. Multiple citations are grouped [?,?,?], [?,?,?,?].

References

1. Auxier, B., Anderson, M.: Social media use in 2021 (2021), https://www.pewresearch.org/internet/wp-content/uploads/sites/9/2021/04/PI.2021.04.07_Social-Media-Use.FINAL.pdf, acedido a 27/06/2022
2. Ellis, C.A., Gibbs, S.J., Rein, G.: Groupware: Some issues and experiences. *Commun. ACM* **34**(1), 39–58 (jan 1991). <https://doi.org/10.1145/99977.99987>
3. eurostat: Individuals using the internet for participating in social networks (2022), <https://ec.europa.eu/eurostat/databrowser/bookmark/fe089ebc-7e3a-46b0-a0c3-7ff07f7b68e5?lang=en>, acedido a 26/06/2022
4. Kemp, S.: Digital 2022: Global overview report (2021), <https://datareportal.com/reports/digital-2022-global-overview-report>
5. Shapiro, M., Preguiça, N., Baquero, C., Zawirski, M.: Conflict-free replicated data types. In: *Proceedings of the 13th International Conference on Stabilization, Safety, and Security of Distributed Systems*. p. 386–400. SSS’11, Springer-Verlag, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24550-3_29, <https://dl.acm.org/doi/10.5555/2050613.2050642>
6. Toomim M., Little G., W.R., B., B.: Braid-http: Synchronization for http (2020), <https://raw.githubusercontent.com/braid-org/braid-spec/master/draft-toomim-httpbis-braid-http-03.txt>, acedido a 26/06/2022