

*Rapport* é uma aplicação colaborativa de redes sociais inspirada no Reddit, que incorpora a sua estrutura de subreddit e sistema de comentários.

Os utilizadores têm a capacidade de criar subreddits, estabelecendo assim comunidades encapsuladas para discussões e colaboração em torno de um determinado tema. Ao criar um subreddit, o utilizador define um nome e uma descrição que refletem o propósito e os objetivos da comunidade. Por exemplo, um subreddit pode ser criado para discutir sobre música, enquanto outro pode ser voltado para a discussão de livros ou filmes.

O sistema de comentários permite interagir com o conteúdo publicado, partilhar feedback, fazer perguntas, iniciar discussões e trocar ideias com outros utilizadores dentro de cada documento.

Uma das características distintivas do Rapport é a sua capacidade de colaboração em tempo real num documento. Tal como no Google Docs, vários utilizadores podem trabalhar em conjunto no mesmo documento, possibilitando a edição simultânea e visualização das alterações feitas pelos outros colaboradores em tempo real.

Relativamente às funcionalidade descritas no primeiro trabalho, as seguintes não foram implementadas:

Sistema de Upvotes: O sistema de upvotes ainda não está ativo. Embora a interface mostre os botões de upvote e downvote, atualmente eles não têm efeito no sistema de classificação do conteúdo.

Controlo de Versão: Embora seja possível colaborar em tempo real em um documento, o sistema de colaboração ainda não possui um controle completo de versão.

Pesquisa: Embora haja uma funcionalidade de pesquisa visualmente presente, ainda não está ativa e não fornece resultados de pesquisa relevantes. Os utilizadores atualmente não podem pesquisar subreddits, documentos ou conteúdos específicos dentro do aplicativo.

Sistema de Permissões: O sistema de permissões, que controla o acesso e as restrições para os subreddits e documentos, ainda não está implementado. Atualmente, todos os utilizadores têm acesso completo a todos os subreddits e documentos existentes, sem a capacidade de definir permissões diferenciadas com base em papéis ou níveis de autorização.

## Funcionalidades

### Browsing

A página principal disponibiliza todos os posts do Rapport, sendo organizado por padrão pelo modo *hot* que utiliza uma fórmula tendo em conta a sua recência e o número de upvotes. É possível ainda organizar por *new*, retornando os posts organizados do mais recente ao mais antigo, e por *top*, que organiza os posts pelo seus upvotes.

```
function joinedHotPosts() {
  const now = Date.now();
  const oneHour = 3600 * 1000;
  const hotPosts = posts.map(post => {
    const subreddit = subreddits.find(s => s.id === post.subredditID);
    const author = users.find(u => u.id === post.authorID);
    const ageHours = (now - new Date(post.dateCreated).getTime()) /
oneHour;
    const score = Math.log10(post.upvotes - post.downvotes) + ageHours
/ 24;

    return {
      id: post.id,
      title: post.title,
      body: post.body,
      upvotes: post.upvotes,
      downvotes: post.downvotes,
      type: post.type,
      image: post.image,
      date: post.dateCreated,
      subreddit: subreddit,
      author: author,
      hotScore: score.toFixed(5)
    };
  }).filter(post => post.hotScore >= 0).sort((a, b) => b.hotScore -
a.hotScore);

  return hotPosts;
}
```

## Comentar

Caso o utilizador tenha uma sessão iniciada, poderá comentar diretamente em qualquer post e ainda face a qualquer comentário, criando deste modo um sistema de comentários encadeados.

Cada comentário tem a seguinte estrutura:

```
{
  "id": "2a8bb6f4-d43c-4371-99e9-30d8c6fb3fc9",
  "postID": 24,
  "parentCommentID": null,
  "upvotes": 814,
  "body": "You're a true virtuoso of comments.",
  "authorID": 9,
  "dateCreated": "2023-05-16T19:02:50.034Z",
  "replies": []
},
```

Quando um novo comentário é adicionado, é verificado se ele possui um comentário pai. Se não tiver, significa que é um comentário de nível superior. Isto é estabelecido pelo elemento a qual se dirige o comentário, se for ao post em si, o comentário pai tem um valor de null, se for dirigido a outro comentário, será assinalado o id desse comentário.

```
if (parentCommentID === null ) {
  comments.push(comment)
} else {
  findParentCommentById(comments, parentCommentID, comment)
}
```

Se o novo comentário tiver um comentário pai, ele é anexado como uma resposta ao comentário pai correspondente.

```

function findParentCommentById(commentsArray, parentCommentID, comment)
{
    commentsArray.forEach( c => {

        if (c.id == parentCommentID) {
            c['replies'].push(comment)
        }
        if (c.replies.length > 0) {
            let foundComment = findParentCommentById(c.replies,
parentCommentID, comment)
            if (foundComment) {
                c['replies'].push(comment)
            }
        }
    })
}

```

Caso o comentário pai tenha respostas adicionais, a função é chamada novamente, procurando o comentário pai dentro das respostas existentes. Este processo continua recursivamente até que o comentário pai correspondente seja encontrado ou até que a hierarquia de comentários seja totalmente percorrida. Quando o comentário pai correspondente é encontrado, o novo comentário é adicionado como uma resposta a esse comentário pai.

Isto cria uma estrutura em árvore, onde os comentários são organizados hierarquicamente, permitindo que os usuários respondam a comentários específicos e visualizem as discussões de forma estruturada.