Collaby
Projeto Ferramenta colaborativa de aprendizado

Átila Camurça
Péricles Henrique
Jovane Pires

14 de fevereiro de 2014

1 Introdução

Num mundo cada vez mais *online* pessoas necessitam cada vez mais ferramentas onde possam trabalhar juntas sem a necessidade de estarem no mesmo lugar. Com essa ideia surgiram ferramentas colaborativas como Google Docs, que permite editar Documentos, Planilhas e Apresentações com vários usuários envolvidos. Daí surge outra necessidade, uma ferramenta que se encaixe no âmbito científico. Surge então o Projeto Collaby.

1.1 Finalidade

- Proporcionar um local central de informações compartilhadas
- Descentralizar a resolução de exercícios
- Criar um local especializado na geração de documentos científicos
- Incentivar alunos a utilizarem LaTeX desde cedo
- Incentivar uso de tecnologias como markdown na criação de documentos
- Aumentar o nível dos projetos haja vista que o número de projetos diferentes deverá aumentar.

1.2 Definições, Acrônimos e Abreviações

- Front-end: é a parte do sistema de software que interage diretamente com o usuário.
- *free*: termo que, em software, significa liberdades em relação a distribuição do código-fonte, criado por Richard M. Stallman.
- *open-source*: propriedade intelectual que é disponível livremente através de licença pública por seus criadores.
- ajax: Asynchronous JavaScript and XML.

1.3 Referências

- TogetherJS: torna página colaborativa
- Definição da linguagem de marcação Markdown

- Pandoc: converte documentos em Markdown para PDF, HTML
- Mathjax: renderiza fórmulas matemáticas LaTeX no browser
- Zend Framework 2 - ambiente web estável e completo
- Twitter Bootstrap 3 - front-end
- jQuery 2 - eventos, ajax
- Ace Editor - editor markdown web
- FontAwesome 4 - icon web fonts
- Marked - fast mardown compiler
- localForage - offline storage

1.4 Visão Geral

2 Descrição do Produto

Collaby é uma ferramenta web para aprendizado colaborativo, onde é possível compartilhar exercícios, listas, materiais de estudo, apresentação de slides, etc.

3 Contexto do Negócio

Funcionará no contexto acadêmico, tanto para discentes quanto docentes. É um projeto *free* e *open-source*.

Mas porque usar linguagem Markdown? Essa resposta pode ser obtida vendo o uso do Markdown na internet. Veja alguns sites que a utilizam:

- Github - README e outros documentos; Wiki;
- Stackoverflow - perguntas e respostas;
- Leanpub - autoria de livros;
- Ghost - blogging platform;
- dillinger - editor *online* com suporte a upload para Dropbox, Google Drive ou Github.

- entre outros.

Além disso hoje temos a necessidade de ter livros, artigos ou outros tipos de texto disponíveis para Desktop, Tablet e smartphones. Dessa forma usar Markdown centraliza o foco no texto e deixa a formatação para outras ferramentas como o pandoc.

4 Objetivos do Produto

- Exercícios, materiais de aula, etc. são criados e compartilhados.
- Documentos podem ser editados em equipe utilizando TogetherJS
- Documentos são públicos para leitura, podendo também ser públicos para escrita se assim o usuário desejar.
- Documentos são escritos em markdown com suporte a fórmulas matemáticas em LaTeX usando pandoc, MathJax.
- Documentos podem ser exportados para PDF ou HTML usando pandoc.
- Lista de links de referência.
- Importar/Exportar de um arquivo markdown.
- Documentos tem suporte a linguagens de programação, podendo o código ser embutido e mostrado com o devido highlighting.
- Documentos podem ser apresentações em LaTeX beamer ou slides em HTML5.

5 Estimativas Financeiras

Gastos incluem:

- Domínio
- Hospedagem

6 Restrições

- A colaboratividade pode ser afetada caso a ferramenta TogetherJS seja descontinuada.
- O editor web codemirror caso deixe de ser mantido pode deixar de funcionar nos browsers cada vez mais modernos.

7 Requisitos

7.1 Requisitos Funcionais

7.1.1 Página inicial

Mapeada como: `/ [Application\Controller\Index.index]`

Usuário deslogado Deve mostrar uma lista com os últimos documentos com a seguinte ordem: última atualização descendente.

Na lista deve aparecer o nome do documento, um link para visualização, nome do dono do documento, link para o usuário dono, data de atualização, tags para o documento, link para buscar documento por tag.

Caso o documento seja um clone mostrar o link para documento original.

Usuário logado Deve mostrar os documentos do usuário logado, permitindo link de edição ao invés de visualização.

7.1.2 Login

Mapeada como: `/login [Application\Controller\Auth.login]`

Página de login tradicional usando usuário e senha, com adição da opção de login com a conta do *Twitter*, *Facebook* ou *Google+*.

Links úteis:

- https://dev.twitter.com/docs/auth/sign-twitter
- https://developers.facebook.com/docs/facebook-login/login-flow-for-web/
- https://developers.google.com/+/quickstart/javascript

Colocar opções para Redefinir Senha 7.1.3 e Criar Conta 7.1.4.

7.1.3 Redefinir Senha

Mapeado como: `/reset-password [Application\Controller\Account.reset-password]`

Caso o usuário esqueça a senha é necessário que ele a redefina. Para isso ele deve preencher uma tela com os dados:

- nome de usuário ou e-mail
- Captcha, para garantir que não é um robô.

Em seguida um *hash* de recuperação de senha é criado com a validade de 1 dia. Daí um e-mail é enviado para o usuário com os dados da solicitação: mensagem, link de recuperação, validade do link.

Ao acessar o link, verifica se o *hash* é válido e mostra a tela para digitar nova senha.

7.1.4 Criar Conta

Mapeado como: `/signup [Application\Controller\Account.signup]`

Criar uma conta deve ser muito simples. A página requisita apenas:

- nome de usuário
- senha
- e-mail

Em seguida um e-mail será enviado para o e-mail informado contendo uma mensagem de boas vindas e o link para a validação do cadastro. O link é composto pela url do site seguida de um *hash*. Ao fazer a requisição o *hash* é verificado com no banco de dados e o usuário é validado.

O *hash* deve ser criado a partir do nome do usuário, o e-mail e um *salt*, que é um pequeno *hash*.

Referências sobre *salt*:

- Salted Password Hashing - Doing it Right
- Salt (cryptography)

O nome de usuário deve ser único, assim como o e-mail. Assim sendo antes de criar a conta deve-se verificá-los antes de prosseguir com a criação da conta.

7.1.5 Novo Documento

Mapeado como: `/new [Application\Controller\Document.new]`

Quando o usuário requisitar um novo documento, o mesmo é criado com valores padrão e logo em seguida o usuário é levado a Editar Documento .

Ao editar o documento pela primeira vez, uma tela modal aparece para que o usuário possa editar o nome do documento e escolher um tema. Ao final ele clica em um botão “Ok”.

Daí ele vai para a edição do documento em si 7.1.6.

7.1.6 Editar Documento

Mapeado como: `/d/:id [Application\Controller\Document.edit]`

Usuário deslogado ou usuário logado que não é dono do documento
Redireciona usuário para modo de visualização (*read-only*) 7.1.11.

Usuário logado e dono do documento Abre o documento em modo de edição.

Esta tela deve ser aberta em nova aba, isso se deve ao fato de que ela será uma tela diferenciada e além disso será uma tela que pode ser compartilhada para que outras pessoas possam editá-la ao mesmo tempo de forma colaborativa. Além disso faz com que não seja possível usar o botão "Voltar" do browser, tendo em vista que o histórico não existe numa aba aberta pela primeira vez.

Ela possui um editor de texto com suporte a *highlight* para **Markdown**.

Deve haver um menu com as opções de Salvar e Pré-visualizar.

Do lado esquerdo um menu para definir qual arquivo editar, **content** ou **template**.

Logo abaixo a estrutura do arquivo em forma de árvore, onde aparecerão links para as linhas que iniciam com #.

E no rodapé uma caixa de mensagens, com o *log* para detectar possíveis erros ao gerar o ducumento em pdf.

7.1.7 Exportar Documento

Mapeado como: `/d/:id/export[/:type] [Application\Controller\Document.export]`

Opção vista em Editar Documento 7.1.6 e Visualizar Documento 7.1.11.

Permite exportar o documento para PDF ou HTML, sendo o tipo padrão é PDF. Ao gerar o arquivo ele deve ser mandando para o browser em forma de download.

Caso o arquivo seja HTML, ele deve ser mandado como um arquivo único. Isso significa que se ele possui CSS ou Javascript deve vir embutido no arquivo.

7.1.8 Importar Documento

Mapeado como: `/import [Application\Controller\Document.import]`

Página destinada a usuários logados que desejam submeter um arquivo Markdown existente e a partir dele criar um documento.

Além de um formulário contendo um *input file* deve permitir o nome do arquivo bem como o layout a ser usado.

7.1.9 Clonar Documento

Mapeado como: `/clone/:id [Application\Controller\Document.clone]`

Todo documento deve ser público, sendo posível assim clonar qualquer um deles. É possível clonar um documento a partir da página Visualizar Documento 7.1.11.

É um processo simples onde todas as informações do documento são copiadas de um usuário para outro. Todavia, como um clone, a edição dos arquivos se torna independente.

7.1.10 Visualizar Usuário

Mapeado como: `/u/:username [Application\Controller\User.view]`

Página destinada a mostrar dados e os documentos de um usuário. De onde outros usuário podem ver e clonar seus documentos.

Os documentos tem a mesma visualização da página inicial 7.1.1.

7.1.11 Visualizar Documento

Mapeado como: `/view/:id [Application\Controller\Document.view]`

Visualização HTML do documento **Markdown** com suporte a fórmulas matemáticas (MathJax).

7.2 Requisitos Não-funcionais

7.2.1 Traduzir a ferramenta TogetherJS

Por ser uma ferramenta muito nova ainda não possui tradução nem mesmo está sujeita ainda. Devido a isso temos duas opções:

- Traduzir os arquivos de forma bruta (Mais fácil mas de difícil manutenção pois quando a ferramenta for atualizada temos que sair de arquivo em arquivo retraduzindo).
- Criar um mecanismo de tradução que irá ajudar a ferramenta para que seja possível traduzir para qualquer língua (Mais difícil mas podemos ser os primeiros a tornar a ferramenta traduzível e contribuir com um programa da Mozilla).

7.2.2 Sincronização de cursor do Ace Editor

Existe um problema com a ferramenta TogetherJS em conjunto com o Ace Editor no qual quando o documento é atualizado o cursor dos outros participantes é resetado para o início do documento.

Outros detalhes em:

- https://github.com/mozilla/togetherjs/pull/927
- https://github.com/triglian/togetherjs
- Ace Editor

8 Ambientação

8.1 Linguagem a ser utilizada

PHP: devido a facilidade de uso, custo barato de hospedagem, boa integração com outros serviços de internet.

8.2 IDE

NetBeans IDE: bem consolidada, suporta PHP, HTML5, CSS3, JavaScript.

8.3 Controle de Versão

Git: controle de versão descentralizado, trabalha bem offline, possui serviços online gratuitos.

Esse projeto pode ser encontrado em: https://github.com/collaby/collaby.

8.4 Framework

Zend Framework 2: devido ao uso de PHP Zend nesse caso é uma ótima opção. Possui documentação abrangente e de boa qualidade.