

Tabi

Extensão de navegador para visualização de abas

Ricardo Geraldes Tolesano

Orientadora: Prof^a. Dr^a. Kelly Rosa Braghetto

13 de Dezembro de 2022

Trabalho de Conclusão de Curso

Departamento de Ciência da Computação

IME USP





Figura 1: *Captura de tela de uma lista com mais de 800 abas abertas no navegador Vivaldi. Cada aba é consideravelmente pequena. Pode ser difícil gerenciar essa quantidade de abas. Fonte: autor*

- **O design de abas de navegador possui diversos problemas**
- **Objetivos do trabalho**
 - ▶ Descrever problemas com o design atual de abas
 - ▶ Desenvolver de uma extensão de navegador para solucionar diversos dos problemas
 - ▶ Analisar os benefícios e problemas da extensão

- **O design tradicional de abas é uma lista horizontal em que os itens podem ser ordenados**
- **Abas são utilizadas de diversas formas**
 - ▶ Acessar sites paralelamente
 - ▶ Abrir links em segundo plano
 - ▶ Manter sites relacionados a uma tarefa
 - ▶ Comparar e curar fontes de informação
 - ▶ Lembretes de tarefas a serem feitas
- **Pesquisa exploratória: pesquisa sobre um assunto não familiar**
 - ▶ Tarefa complexa
 - ▶ Incerteza sobre relevância de cada aba
 - ▶ Mudança constante de prioridade das páginas

Problemas do design atual de abas

- **Visualização em lista única horizontal é limitada**
- **Design não é adequado para suportar todos os usos das abas**
- **Abas são desconectadas do contexto, intenção, e tarefa referentes ao momento em que foram abertas**
- **O design não lida bem com dezenas ou centenas de abas**
 - ▶ Abas se tornam muito pequenas
 - ▶ Abas saem do campo de visão do usuário e são esquecidas
 - ▶ Dificuldade em organizar muitas abas
 - ▶ Confusão
 - ▶ Estresse
 - ▶ Perda de produtividade

Problemas do design atual de abas

- **Poucas opções para organização não permitem que o usuário recrie seu modelo mental da tarefa sendo realizada. Por exemplo, indicar quais abas são relevantes.**
- **A organização e ordenação das abas é quebrada com o tempo conforme novas abas são abertas**
- **Organização manual é custosa para o usuário, principalmente em tarefas complexas tais como pesquisa exploratória**

Como resolver os problemas

- **Mostrar mais informações e contexto sobre as abas, tais como aba progenitora e abas relacionadas**
- **Oferece mais formas de organização e categorização**
- **Diminuir a necessidade de organização manual por meio de um algoritmo ou inteligência artificial**
- **Utilizar uma forma diferente de visualização das abas**
- **Guiar a atenção do usuário para abas relevantes, por exemplo, reduzindo ou aumentando sua opacidade ou tamanho da fonte**

Trabalhos Relacionados

- **Organização manual de abas em grupos: Toby e Workona**
- **Fechar todas as abas e movê-las para outra lista: OneTab, Session Buddy**
- **Organização automática de abas em grupos com base em regras: Sidebery, Cluster**
- **Organização de abas em hierarquias: Tree Style Tab, Tabs Outliner**
- **Alguns artigos descrevem agrupamento automático de abas utilizando inteligência artificial**
- **Os artigos Tabs.do e Bento Browser desenvolveram uma interface para abas similar a de um aplicativo de gerenciamento de tarefas**

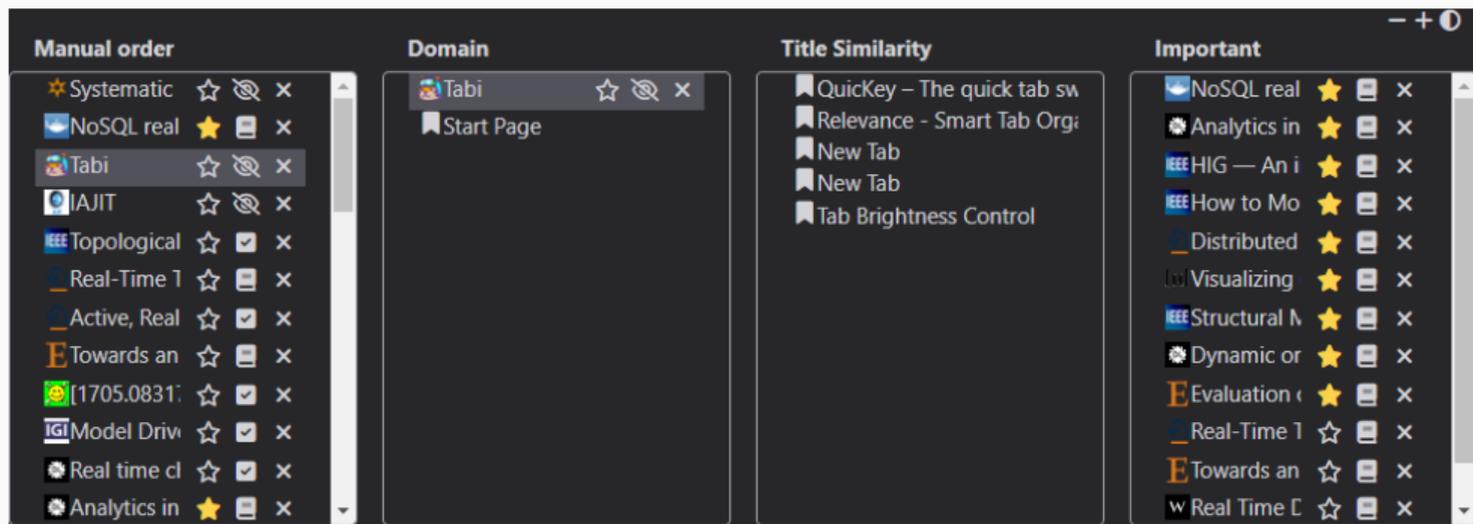


Figura 2: O design inicial da extensão utilizava uma página inteira

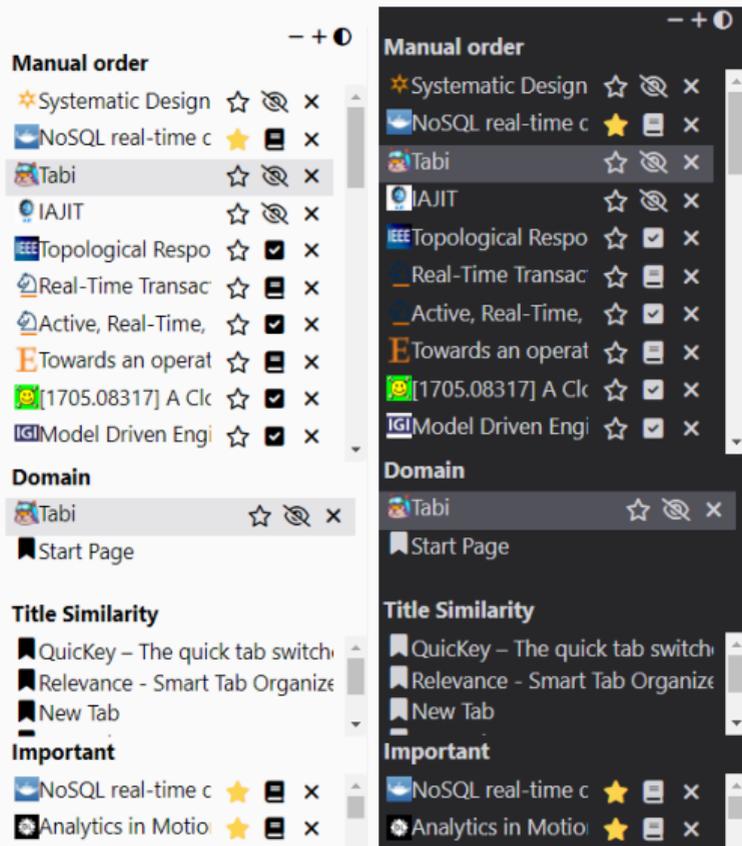


Figura 3: Design final da extensão nos modos claro e escuro. Fonte: autor

Funcionalidades da extensão

- **Permite marcar abas como "importantes", "para leitura" e "já lidas". Essas abas aparecem também em uma lista própria**
- **Exibe abas e favoritos relacionados por domínio e por similaridade de título**
- **Permite modo claro, modo escuro e escolha do tamanho da fonte**

Como funciona o desenvolvimento de uma extensão

- **HTML, CSS, JavaScript**
- **API de extensões do Firefox**
 - ▶ Dados sobre abas e favoritos
 - ▶ Definição de páginas a serem abertas no painel lateral
 - ▶ Permite guardar dados no armazenamento do navegador
- **Código empacotado em um zip**
- **Publicação da extensão no site do Firefox**

- **Svelte: framework front-end**
 - ▶ Alta performance
 - ▶ Fácil de escrever
- **Tailwind: framework para CSS**
 - ▶ Mesma aparência em múltiplos navegadores
 - ▶ Define padrões e cores
 - ▶ Mais simples e legível do que escrever CSS
- **Limitações na API dos navegadores para**
 - ▶ Obtenção do contexto em que as abas foram abertas e qual foi sua aba progenitora
 - ▶ Criação do design proposto com boa usabilidade e performance

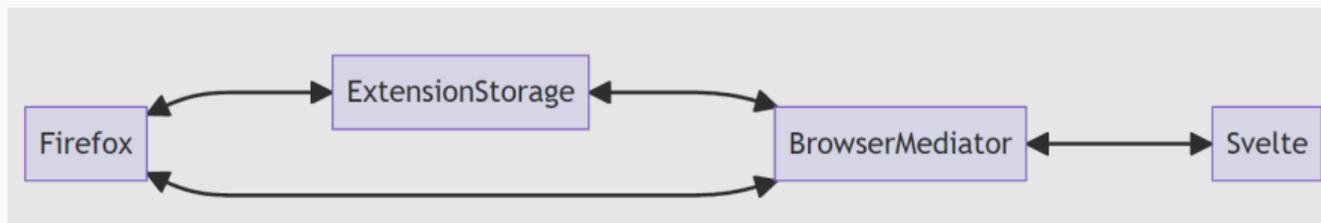


Figura 4: *Arquitetura do projeto. Fonte: autor*

- A extensão Tabi permite categorizar abas e mostrar as abas importantes em uma lista própria e isso parece auxiliar na realização de pesquisa exploratória
- O algoritmo para mostrar abas com títulos similares não funciona tão bem
- Limite de espaço do painel lateral pode impactar a usabilidade da extensão

Possibilidades futuras

- Permitir adicionar anotações às abas
- Utilizar outro algoritmo para abas relacionadas
- Reduzir a opacidade de abas já lidas
- Mostrar abas antigas aleatoriamente para que o usuário não as perca de vista
- Outros designs, por exemplo, disposição de abas em grafos
- Obter feedback dos usuários e implementar melhorias
- Medir a usabilidade da extensão por meio de um questionário padronizado (System Usability Scale)

- 1 Introdução e objetivos
- 2 Conceitos
- 3 Problemas do design atual de abas
- 4 Como resolver os problemas
- 5 Trabalhos relacionados
- 6 Desenvolvimento de uma extensão de navegador
- 7 Resultados
- 8 Possibilidades futuras



