Uma Questão de Inclusão

Para o nosso projeto de **"Informática e Sociedade"**, proponho um tema que é prático, impactante e alinhado com o enunciado: **acessibilidade digital**. A ideia é desenvolver uma extensão de navegador para o Google Chrome que funcione como uma **"caixa de ferramentas de acessibilidade visual"**.

O Problema

A web, em sua forma atual, não é inclusiva. Muitos sites dependem de cores e designs que excluem uma parcela significativa da população. Pessoas com **daltonismo** têm dificuldade em diferenciar botões de status ou informações em gráficos, enquanto aqueles com **baixa visão** sofrem com o baixo contraste, fontes pequenas e falta de zoom adequado.

Este não é um problema de nicho. Ele afeta milhões de pessoas em todo o mundo.

- **Pessoas com deficiência visual:** A Organização Mundial da Saúde (OMS) estima que pelo menos **2,2 bilhões de pessoas** em todo o mundo tenham deficiência visual.
- **Daltonismo:** No Brasil, a prevalência do daltonismo é de aproximadamente **3,5% da população**, afetando mais homens (cerca de 5% a 8%) do que mulheres (0,2% a 0,5%). Isso significa que, em um grupo de 100 pessoas, cerca de 3 a 4 indivíduos podem ter alguma forma de deficiência de cores. A cor, usada para dar significado em interfaces (como verde para "sucesso" e vermelho para "erro"), não é, portanto, universalmente compreendida.
- **Fadiga Visual Digital:** Um problema crescente devido ao tempo excessivo de tela. Estudos mostram que mais de **60% dos adultos** experimentam sintomas como olhos secos e visão embaçada, que podem ser amenizados com ajustes de tela.

Nossa Proposta: "Assistente Visual"

Em vez de criar uma ferramenta para cada problema, podemos integrar diversas funcionalidades em um só lugar. Nossa extensão teria um painel simples e intuitivo, permitindo ao usuário:

- **Filtros para Daltonismo:** O usuário poderá escolher um filtro para corrigir ou adaptar as cores de acordo com o seu tipo de daltonismo, tornando informações que dependem de cores (como gráficos) acessíveis.
- **Ferramentas para Baixa Visão:**
    - **Alto Contraste:** Alternar rapidamente entre esquemas de cores de alto contraste, essencial para quem tem baixa acuidade visual.
    - **Zoom de Texto:** Ampliar apenas o texto da página sem quebrar o layout, garantindo que o conteúdo seja legível e a formatação seja mantida.
    - **Ferramenta de Lupa:** Adicionar uma lupa com zoom editável para inspecionar partes específicas da página.
- **Filtros para Conforto e Leitura:** Uma funcionalidade para reduzir a emissão de luz azul da tela, que prejudica o sono e causa fadiga visual. Para isso, a extensão tornaria as cores da tela mais quentes (amareladas ou alaranjadas), aliviando o cansaço ocular, especialmente ao usar o computador à noite.

Por que esta ideia?

- **Amplo Alcance:** Atualmente, a web, navegadores e sites representam as aplicações mais acessíveis e amplamente utilizadas por usuários finais na computação moderna. Dessa forma, uma extensão que opere neste segmento de mercado significativo promoverá a inclusão de maneira mais eficaz.
- **É um projeto prático:** Podemos desenvolver um protótipo funcional com os conhecimentos que já temos em HTML, CSS e JavaScript. A experiência de criar uma extensão de navegador é muito valiosa para o nosso portfólio.
- **Soluciona um problema real:** Estamos criando uma ferramenta com um propósito genuíno. Acessibilidade não é apenas uma diretriz técnica, é uma questão de inclusão social.
- **Tem um ótimo potencial para o artigo:** O artigo acadêmico documentará toda a nossa pesquisa e o processo de desenvolvimento, da análise do problema e da revisão de literatura até a demonstração dos resultados.

Especificações do Projeto

- O repositório GitHub onde o projeto se encontrará deverá ser público
- O projeto utilizará a licença GNU GPLv3
- A extensão não deverá coletar nenhum dado do usuário, salvar contas ou ler o conteúdo das páginas

---

Cronograma

18/09	Apresentação do projeto
25/09	Entrega de POC
09/10	Entrega do primeira protótipo funcional (MVP) Início do período de testes
30/10	Entrega de uma versão aprimorada Início da escrita do artigo
27/11	Conclusão dos testes Entrega da primeira versão oficial Apresentação final para a comunidade
11/12	Entrega final (produto final + artigo)

Referências e Fontes de Pesquisa

Para aprofundar a pesquisa e usar no artigo acadêmico, aqui estão alguns links e artigos relevantes sobre o tema:

- **Dados da OMS sobre Deficiência Visual:**
  - OMS - Blindness and vision impairment
- **Estudos sobre Daltonismo e Cor na Interface:**
  - W3C Web Content Accessibility Guidelines
- **Artigos sobre Fadiga Visual Digital:**
  - The Vision Council - Digital Eye Strain Report
- **Artigos Técnicos sobre WebExtensions e Acessibilidade:**
  - Desenvolvimento de Extensões do Chrome
  - Extensões / Desenvolver | Chrome for Developers
  - Práticas recomendadas | Chrome Extensions | Chrome for Developers
  - Get Started | Chrome Extensions | Chrome for Developers
  - Program Policies | Chrome Extensions | Chrome for Developers
  - Hello World | Chrome Extensions | Chrome for Developers
- **Extensões utilizadas como referência:**
  - Ampliar para o Google Chrome
  - CDalton
  - Colorblindly
  - High Contrast
  - Verificador de acessibilidade de cores
  - Eye-Able® - Accessibility Assistant

Convenções de Trabalho

Esta seção estabelece as convenções de desenvolvimento adotadas para este projeto, visando garantir organização, colaboração eficiente e um histórico de versões claro e profissional.

1. **Padrão de identação**
   2 espaços (padrão JavaScript/CSS).

2. **Nomes de Branches, Pastas e Arquivos**
   Para ambos os casos, deve-se utilizar o padrão “kebab-case”, onde todas as letras são minúsculas e as palavras são separadas por “-” (hífen). Exemplo de sua utilização: “filtro-daltonismo”, “botao-lupa”, etc.

   Observação: As funções, classes e outros elementos que se encontram "dentro" de arquivos (código, por exemplo) devem seguir o padrão mais comum para a linguagem, e/ou acompanhar os padrões de API. **Exemplo para o JavaScript:** variáveis e funções (lowerCamelCase); constantes (UPPER_SNAKE_CASE); Classes, Objetos e Componentes (UpperCamelCase ou PascalCase), etc.

3. **Modelo de Ramificação**
   Git Flow Adotaremos o modelo de ramificação **Git Flow**. Este fluxo de trabalho organiza o desenvolvimento em torno de duas branches principais e perenes, e branches de suporte temporárias.

- ***main:*** Esta branch é a fonte da verdade para o código em produção. Cada commit na *main* representa uma nova versão estável e deve ser acompanhado por uma tag de versão. O código aqui deve estar sempre em estado de entrega.
- ***develop:*** É a branch principal de desenvolvimento. Ela serve como um ponto de integração para todas as funcionalidades concluídas. O desenvolvimento diário e os Pull Requests de features são direcionados a ela.

Branches de Suporte:

- ***feature:*** Utilizadas para desenvolver novas funcionalidades. São criadas a partir da *develop* e, ao serem finalizadas, são mescladas de volta na *develop*.
- ***release:*** Utilizadas para preparar o lançamento de uma nova versão. Permitem isolar o ciclo de testes finais e correções de última hora, sem interromper o desenvolvimento de novas features na *develop*.
- ***hotfix:*** Utilizadas para aplicar correções críticas em versões que já estão em produção (na *main*).

4. Padrão de Versionamento

Para a numeração de versões, utilizaremos o padrão Versionamento Semântico (SemVer), no formato “MAIOR.MENOR.CORREÇÃO”.

- **MAIOR *(MAJOR)***: Incrementado para mudanças incompatíveis com versões anteriores (**breaking changes**).
- ***MENOR (MINOR)*:** Incrementado para a adição de novas funcionalidades de forma compatível com a versão anterior.
- **CORREÇÃO *(PATCH)*:** Incrementado para correções de bugs retrocompatíveis.

5. Marcos do Projeto e Versões

- **Versões *0.x.y*:** Serão utilizadas para todo o ciclo de desenvolvimento anterior ao MVP. Cada *release* nesta fase representará um avanço significativo no desenvolvimento (ex: *v0.1.0*, *v0.2.0*).
- **Versão *1.0.0*:** Este será o marco que define a entrega do nosso **MVP (Produto Mínimo Viável)**. Atingir a *v1.0.0* significa que o produto possui o conjunto essencial de funcionalidades e está em um estado estável e apresentável.
- **Versões Pós-*1.0.0*:** Seguirão as regras do SemVer. Novas funcionalidades incrementarão o número *MENOR* (ex: *v1.1.0*) e correções de bugs incrementarão o *PATCH* (ex: *v1.1.1*).

6. **Exemplo da Estrutura do Projeto**