# EBAC

## Qualidade de Software

### Exercício 1

### Relatório de desempenho, acessibilidade e SEO: principais problemas encontrados

1. Recursos como folhas de estilo e scripts estão sendo carregados ao mesmo tempo que o conteúdo principal do site, o que faz com que os dados demorem a ser exibidos;
2. O banner principal do site está muito pesado, o que potencialmente pode ser um problema em dispositivos menores e conexões mais lentas (vide evidência 1 no final deste documento);
3. O banner principal está configurado como *lazy loading*, o que posterga sua renderização mas é um recurso recomendado quando as imagens não estão na área principal de visualização e aí seriam carregados conforme o usuário rola a página. Neste cenário, tratando-se de uma *hero image*, não é recomendado (vide evidência 2);
4. Recursos como folhas de estilo e scripts devem ser minificados (processo de remoção de caracteres desnecessários que não influenciam a funcionalidade) quando utilizados em produção;
5. Há elementos de acessibilidade sendo usados de forma incorreta/incompleta (vide evidência 3);
6. Botões e links precisam ter um texto que o diferenciem nos leitores de tela (vide evidência 4);
7. Alguns textos não trazem contraste suficiente para leitura (vide evidência 5);
8. O site não utiliza certificado e conexão segura (HTTPS);
9. O site não traz uma descrição que possa ser utilizada nos indexadores de busca.

### Soluções propostas

1. Arquivos .js e .css devem ser carregados de forma padrão somente quando estritamente necessários para o primeiro conteúdo a ser mostrado na página; demais recursos devem ser carregados com as opções “defer” ou “async” ou, no caso de .js, serem colocados no final da página;
2. Deve-se comprimir as imagens utilizando uma ferramenta como Photoshop para otimizá-las para apresentação na web/mobile;
3. Deve-se remover a opção de *lazy loading* do banner principal; esta opção deve ser utilizada em imagens que ficam inicialmente escondidas e são reveladas com a rolagem da página;
4. Deve-se utilizar alguma ferramenta/plugin para minificar recursos .js e .css;
5. Quando uma lista utiliza o atributo *role=”tablist”*, os elementos filhos devem utilizar o atributo *role=”tab”*;
6. Aplicar o atributo *id*, *title* ou *aria-label* a botões e links;
7. Aumentar o contraste entre a cor de fundo e a cor de texto; para evitar tentativa e erro, existem algumas ferramentas que podem ajudar neste cenário como o plugin Stark para o Figma;
8. Utilizar um certificado de segurança para trafegar os dados em conexão protegida (HTTPS);
9. Adicionar a tag *meta description* ao HTML.

### Exercício 2

### Alteração do front do carrinho via DevTools

Alterei a cor principal da loja para azul e coloquei todos os termos em português.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

### Anexo: Evidências

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Evidência 1

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Evidência 2

Texto

Descrição gerada automaticamente

Evidência 3

Tela de celular com publicação numa rede social

Descrição gerada automaticamente

Evidência 4

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Evidência 5