



**Ciências
ULisboa**

Faculdade
de Ciências
da Universidade
de Lisboa

UbiWAN

Accessibility Checker



Documentação

Ana Salvado

Inês de Matos

João Vicente

ÍNDICE

INTRODUÇÃO	3
QUALWEB (BACKEND)	4
ARQUITECTURA	5
<i>Núcleo</i>	6
<i>Simulador de Processamento Browser (BPS)</i>	6
<i>Simulador de Interação(IS)</i>	7
AVALIAÇÃO: TIPOS	7
AVALIAÇÃO: COMO	7
PLUGIN	8
TECNOLOGIAS USADAS.....	8
INSTALAÇÃO	8
NAVEGAÇÃO.....	9
ARQUITECTURA	10
<i>Background</i>	11
<i>Result</i>	11
<i>Sidebar</i>	12
OUTROS FICHEIROS.....	16
<i>Manifest.json</i>	16
<i>Index.html</i>	16
<i>Result.html</i>	16
<i>Options.html</i>	16
<i>Index.js</i>	16
<i>error.html</i>	16
<i>Options.JS</i>	16
AVALIAÇÃO	17
RESULTADOS.....	18
TRABALHO FUTURO	20

INTRODUÇÃO

A acessibilidade é uma preocupação que tem surgido cada vez mais nos dias de hoje, porque queremos que todas as pessoas, quanto possível, possam usufruir dos seus direitos de igual forma. Porém na nossa sociedade, este conceito, já tem vindo a ser vinculado em alguns cenários, por exemplo, o uso de rampas que facilita que as pessoas de mobilidade reduzida se possam deslocar.

As dificuldades geralmente são relacionadas com a visão, a parte cognitiva, a audição e a mobilidade. Mais especificamente na web, existem alguns problemas de acessibilidade tais como a perceção de imagens, vídeo e áudio, leitura e escrita de texto e manipulação de periféricos.

Por muito desagradável que seja, não é possível obter uma acessibilidade perfeita que alcance todas as pessoas, pois por vezes os perfis são opostos ou muito peculiares, o que torna complicado a sua resolução.

Muitas das tecnologias surgiram para ajudar nestas resoluções. No entanto, esta preocupação não tem sido suprimida no que toca à navegação em aplicações web. Muitas vezes este fator é descartado porque os webmasters não acham que seja necessário, não têm o conhecimento suficiente ou por falta de consciência dos problemas que existem. Por outro lado, a falta de acessibilidade pode advir da dificuldade de avaliar um website.

Para este efeito existem uma série de ferramentas disponíveis, tais como aChecker, WAVE, TotalValidator, QualWeb, entre outros para ajudar a detetar erros nos websites.

O QualWeb é uma aplicação construída na Faculdade de Ciências, por uma ex-aluna¹, e que nos foi proposto continuar o seu desenvolvimento, visto que o projeto estava suspenso. Mais tarde, introduzimos uma nova componente ao projeto, que evita que as pessoas tenham que aceder ao website para avaliar a acessibilidade de uma página. Essa componente possibilita que a avaliação seja automática (ou seja, não é necessário introduzir o url no formulário) e o acesso mais intuitivo: UbiWAN.

O UbiWAN é um plugin pensado para ajudar os programadores web a avaliar os sites em relação à sua acessibilidade. Esta ferramenta, essencialmente, tem o objetivo de visualizar as heurísticas avaliadas e os seus resultados de uma forma clara e eficiente, além da automatização da avaliação. Estas heurísticas estão implementadas no servidor do QualWeb e baseiam-se nas orientações dadas pela WCAG 2.0.

Pelo que este projeto ainda está em desenvolvimento, e continuará, esta documentação poderá sofrer alterações, tendo sido alterada pela última vez a **1 de Fevereiro de 2016**.

¹ Fernandes, Nádia, et al. "Three web accessibility evaluation perspectives for RIA." *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*. ACM, 2013.

QUALWEB (BACKEND)

O QualWeb é uma ferramenta para avaliar a acessibilidade de páginas Web. Através desta avaliação é determinado um nível aproximado da acessibilidade da página. Para determinar esta acessibilidade são usadas orientações específicas, mas conhecidas como WCAG, para saber se essa página respeita as especificações lá descritas.

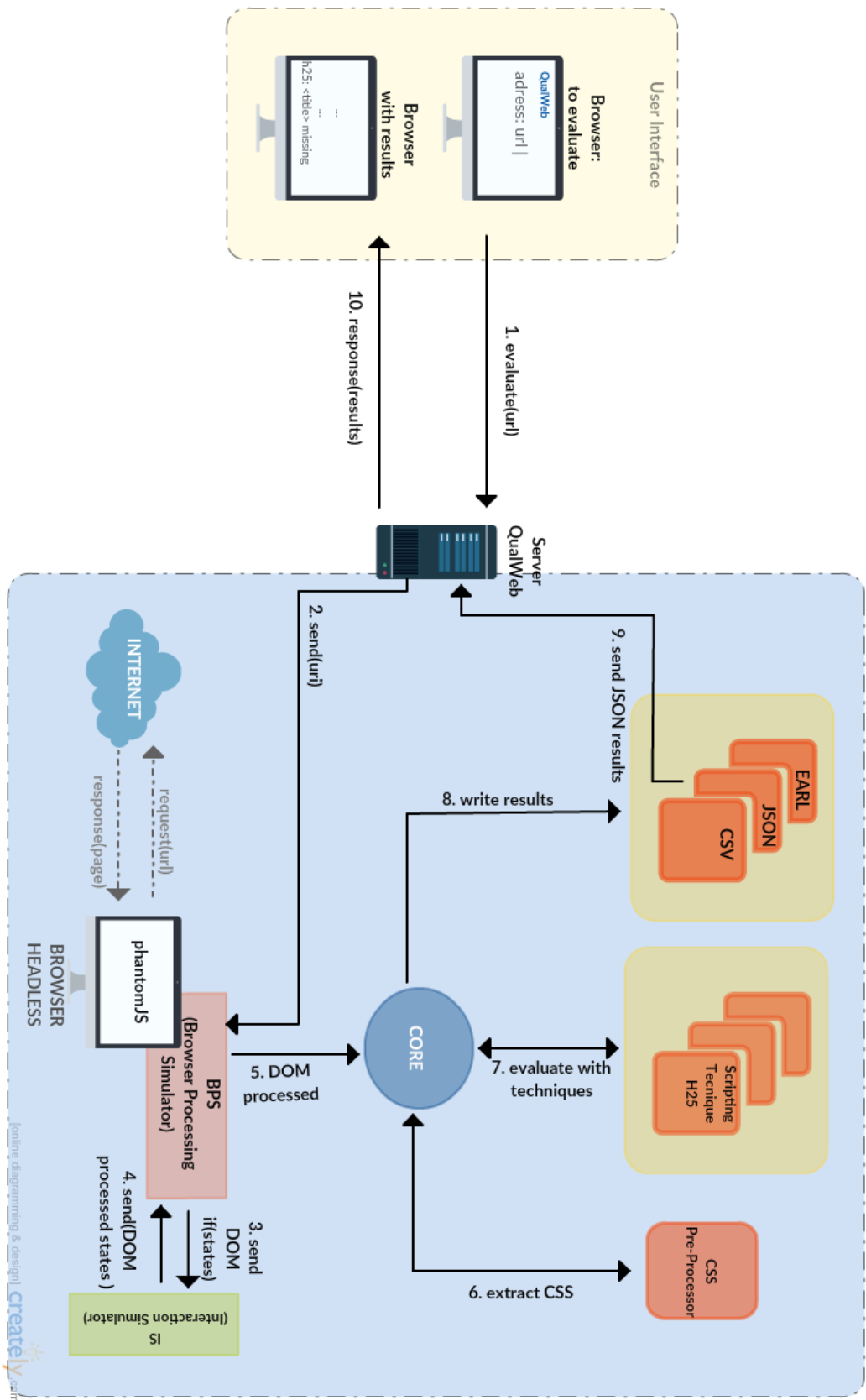
Para fazer esta avaliação é necessário introduzir o url da página que se pretende avaliar e é dada ao utilizador a possibilidade de escolher várias opções da avaliação, como o contexto (*Desktop* ou *Mobile*), se quer avaliar CSS e/ou JavaScript e o modo de apresentação, por critérios ou por técnicas. Depois de ser feita a avaliação são apresentados os resultados. É mostrado o número de elementos que foram avaliados, bem como a percentagem dos que passaram, falharam e tiveram avisos. A percentagem do número de elementos que passaram nos testes equivale à acessibilidade de uma página, ou seja, se for indicado que dos elementos avaliados, 60% passaram aos testes, indica que a página é 60% acessível.

Para além deste quadro resumido, também é possível ver com mais detalhe todas as técnicas, ou seja, a descrição que a técnica representa, o respetivo elemento ou mesmo a falta dele, por exemplo, não inserir a tag <title> pode despoletar a falha da técnica H25, e o erro que é causado (em caso de falha ou aviso).

Como é descrito nas orientações do WCAG, as técnicas atualmente que são usadas são unicamente das classes H(HTML) e C(CSS).

O QualWeb é composto por três grandes componentes: o núcleo, o simulador de processamento no browser(BPS) e o simulador de interação(IS).

ARQUITECTURA



NÚCLEO

O simulador (BPS, browser processing simulator) que processa os pedidos do browser recebe o URI da página a ser avaliada e processa-a. Se a página tiver mais do que um estado, o DOM é enviado para o Simulador de Interação, que vai simular os vários estados da página. Um estado é quando o DOM de uma página é alterado (com JavaScript, por exemplo), sem que o URL mude. Depois de simular os estados devolve-os ao BPS para serem processados e posteriormente o DOM (processado) é enviado para o núcleo para este proceder à avaliação.

Desta forma, o DOM serve de matéria prima para o núcleo que cumulativamente avalia a qualidade da RIA. A RIA (Rich Internet Applications) é uma aplicação Web que se caracteriza por ser semelhante às aplicações de software de desktop, ou seja, páginas que são dinâmicas. Hoje em dia as páginas web são instáveis e isto deve-se ao uso de linguagens dinâmicas, mais complexas, tais como JavaScript e/ou AJAX, que ao serem executadas provocam alterações em diversos aspetos da página.

Para extrair o CSS da árvore do DOM é utilizado um pré-processador de CSS que obtém todo o CSS da referente página. De seguida, para realizar a avaliação o QualWeb usa a componente das Técnicas, estipuladas pelo WCAG 2.0.

Em relação aos resultados, existem três formas de output: EARL, JSON e CSV. O EARL é um formato padrão para relatórios de acessibilidade. O CSV, mais apropriado para propósitos estatísticos, guarda todas as técnicas de uma forma mais resumida, sendo apenas acumulado por cada técnica o número de elementos que passaram, falhou ou que tiveram aviso. Por último, em formato JSON é guardado todo o resultado para ser usado em páginas web (como Web Service), assim é obtido o resultado no browser.

Atualmente, esta versão do QualWeb implementa 31 técnicas de HTML e 13 de CSS do WCAG 2.0, representando as práticas básicas para a criação de conteúdo acessível.

SIMULADOR DE PROCESSAMENTO BROWSER (BPS)

Este módulo simula o processamento do web browser através do PhantomJS. O PhantomJS é um headless browser, ou seja, um browser que não beneficia de uma representação gráfica, portanto é CLI (command-line interface) e que usa WebKit. O WebKit é um motor de software, open source, para a “renderização” de páginas web em web browsers. Assim, é possível processar as páginas sequencialmente de uma forma rápida, pelo que não oferece uma GUI que seja pesada, favorecendo, também, avaliações de larga-escala.

SIMULADOR DE INTERAÇÃO (IS)

Para lidar com os desafios das RIAs, foi integrado o simulador de interação. Esta componente, é responsável por simular as ações dos utilizadores e despoletar elementos interativos na página. Como resultado temos acesso aos diferentes estados das RIAs, no formato de DOM. Para realizar esta simulação é usado o JQuery para interagir com os elementos clicáveis (por exemplo, botão, divs, inputs). Para cada estado são obtidos os estados filhos, que são alcançáveis por cada elemento clicável, e processados pelo BPS. Assim, para cada estado filho é verificado se já foi visitado e, senão é repetido o processo. No final é obtido um grafo de toda a interação de estados da RIA.

AValiação: TIPOS

Existem três tipos de avaliação a serem considerados:

Avaliação	Descrição	IS	BPS
A1	Avaliação realizada antes do processamento do web browser	INACTIVO	INACTIVO
A2	Avaliação realizada depois do processamento do web browser sem considerar estados da página	INACTIVO	ACTIVO
A3	Avaliação realizada depois do processamento do web browser considerando estados da página	ACTIVO	ACTIVO

Atualmente, o QualWeb está a atuar como A2 e, assim, o plugin também.

AValiação: COMO

Para proceder a uma avaliação existem duas formas, que é através de um website ou através da linha de comandos, sendo esta última exclusiva para os utilizadores proprietários do serviço.

Pela primeira forma, basta entrar em <http://qualweb.di.fc.ul.pt/>, aceder ao separador "Accesibility", preencher o campo "Address" com o URL pretendido e configurar os campos de avaliação.

Isto gera um link composto pelo URL e por flags, que identificam as configurações da avaliação, para depois ser interpretado pelo servidor. Existe uma outra documentação mais técnica relativa ao QualWeb, a explicar como é que este link é produzido e o que é que significa cada flag. Assim como, o detalhe da implementação deste.

PLUGIN

Este plugin, partilha do mesmo objetivo que o QualWeb (não fosse este o motor do UbiWAN), de ajudar os programadores de aplicações web a entender o nível de acessibilidade das suas páginas, avaliando-as e indicando também possíveis soluções. Contudo, o único meio que os utilizadores têm de aceder a esta ferramenta, atualmente, é através de um website, que os obriga a ter que navegar dentro dele até chegar ao formulário de inserção do URL para dar início à avaliação. A implementação do plugin, servirá como uma extensão do website ou mesmo um substituto deste.

Através da avaliação automática, isto é, um pedido de avaliação que é enviado a partir do momento em que a página se encontra totalmente carregada, evita-se que o utilizador proceda a ações, talvez, desnecessárias para o efeito, tais como o ter de navegar até ao formulário respetivo da avaliação.

O facto de introduzirmos um painel lateral com destaques na página avaliada, pode facilitar a interpretação das falhas no respetivo código. Pelo que a componente de visualização lado a lado, dá um maior controlo sobre os erros apontados e o elemento que é afetado. Ao contrário do website, que simplesmente mostra os resultados da avaliação numa só página, tirando o facto de não existir a visualização, o que restringe a identificação dos elementos na página processada.

Assim, de uma forma muito simples, queremos melhorar o processo de avaliação através do UbiWAN.

TECNOLOGIAS USADAS

Este plugin foi concebido unicamente para o browser **Google Chrome**, logo não irá funcionar noutros.

A nossa aplicação é implementada através de **JQuery** para as funcionalidades, **Bootstrap** para o estilo da apresentação da informação e **JSON** para a comunicação entre servidor e páginas.

INSTALAÇÃO

Existe duas formas de instalar o plugin: através da forma comprimida e da forma descomprimida.

Da primeira forma, basta:

1. Aceder a **chrome://extensions** (escrever na barra de endereço)
2. Arrastar o plugin para a janela anterior

Na forma descomprimida, é preciso:

1. Aceder a **chrome://extensions** (escrever na barra de endereço)
2. Permitir o modo de programador (checkbox no cimo da página)
3. Carregar extensão descomprimida
4. Aceder até à pasta raiz da aplicação, onde se encontra o manifesto

NAVEGAÇÃO

Nós apresentamos uma interface diversificada, ou seja, temos uma componente onde se inicia a avaliação e outra onde apresentamos os resultados.

Primeiramente, é apresentado um ícone do plugin na barra de ferramentas do browser assim que este é instalado, que dá acesso à primeira componente. A avaliação começa imediatamente quando a página é carregada. Enquanto este processo ocorre, é apresentado uma etiqueta no ícone a dizer “Eval” e assim que é concluída a avaliação, esta etiqueta passa a ter o valor da percentagem de testes bem-sucedidos da página.

Ao clicar no ícone surge um pop-up onde é apresentado uma tabela de valores percentuais do número de testes que passaram, falharam e com algum aviso. É apresentada nesta componente, três botões com as seguintes funcionalidades: opções, reavaliação da página e acesso aos detalhes.

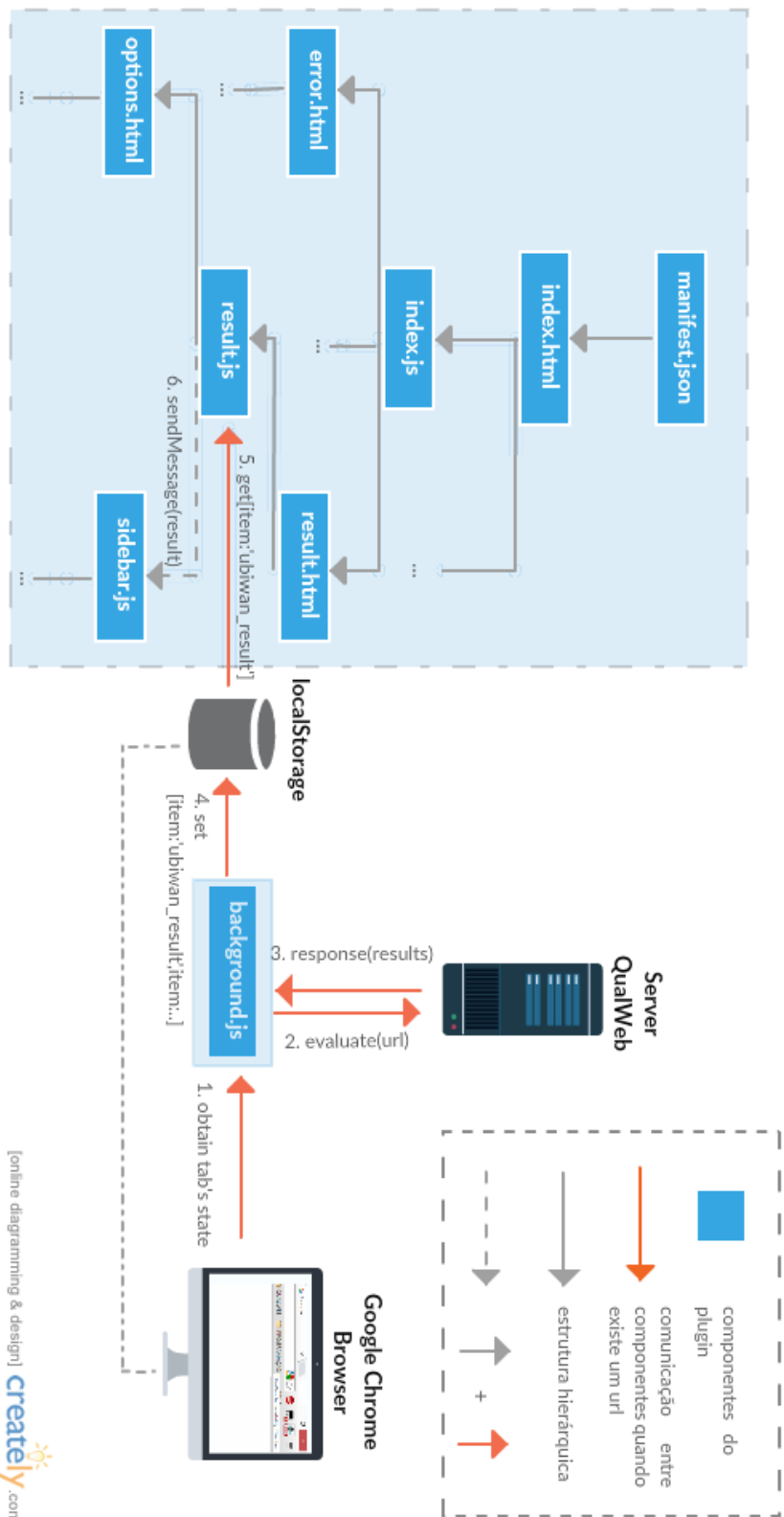
Ao carregar nas opções, é aberto um novo separador onde dá ao utilizador a possibilidade de escolher o contexto (desktop ou mobile) e optar por avaliar css e/ou JavaScript. No botão da reavaliação, faz com que a página seja avaliada outra vez.

Por último, temos o botão que dá acesso, então, à nossa segunda componente de interface: os detalhes.

Nesta parte, é aberto um painel lateral com uma lista dos critérios que foram avaliados, um filtro rápido para restringir a lista apresentada e um botão de fechar. Clicando num dos critérios, é se redirecionado para um conjunto de técnicas avaliadas, associadas ao mesmo. Fornecemos um acordeão para cada uma delas, de forma a apresentar os respetivos dados, tais como, em caso de falha, a linha onde se encontra o erro, o elemento afetado e a sua descrição.

É apresentado uma “checkbox” por cada elemento que falhou no teste, que possibilita ao utilizador visualizar graficamente a componente afetada e o código correspondente. À medida que estas checkboxes são ativadas, o efeito é cumulativo, ou seja, vai incrementando na apresentação. Para remover todas as apresentações basta clicar no botão “Clear”.

ARQUITECTURA



Existem três componentes principais que fazem com que os resultados sejam obtidos no plugin: `background.js`, `result.js` e `sidebar.js`. Tal como foi mostrado anteriormente, na arquitetura do QualWeb, o pedido é efetuado, automaticamente, a partir do momento em que um separador é aberto e é verificado que existe um url válido. Assim que isto acontece, é enviado um pedido de avaliação para o servidor do QualWeb no qual se obtém uma estrutura JSON com o resultado da avaliação.

Este processo é gerido pelo **background.js**, que depois informa o **result.js** e o **sidebar.js** de alterações que estejam a decorrer no separador tal como os resultados em si.

BACKGROUND

Este script está à escuta de eventos que ocorrem no browser. Ou seja, quando é que separadores são abertos, mudados, fechados e/ou atualizados de forma a gerir a informação em função de uma destas ações. Também verifica se o que está na barra de endereço é um URI válido para a avaliação; caso seja, é enviado um pedido de avaliação ao servidor do QualWeb. Este após avaliar o URL respetivo do separador, envia os resultados numa estrutura JSON para este script. Quando este obtém os resultados, guarda-os num `localStorage` numa série de itens que foram previamente processados e separados, destacando o “`ubiwan_result`” que guarda o resultado total. Caso seja verificado que o URI não é válido, por exemplo um separador que não tenha URL ou que esteja em páginas do próprio browser (histórico, extensões e definições), é redirecionado para uma página de erro (`error.html`) indicando as possíveis razões. Se o utilizador optar por reavaliar a página, é enviado, novamente, um pedido de avaliação ao servidor, ficando a página do painel lateral em modo de espera. No caso de se fechar um separador com um URL avaliado, toda a informação é removida do `localStorage`. Por fim, caso o utilizador mude a sua navegação para outro separador que já tenha um URI associado (que não tenha sido ainda avaliado), cancela-se a avaliação do URL anterior (caso esteja a decorrer alguma) e volta-se a processar a nova página. Fazendo refresh à página, ou abri-la ao mesmo tempo noutra separador não irá reavaliar a página.

Aqui também é gerida a etiqueta que aparece no ícone do plugin apresentado no browser, ou seja, entra num estado “Eval” quando está a ser avaliado, não apresenta nada quando existe um erro e depois de ser avaliado apresenta a percentagem de testes passados.

RESULT

Nesta componente é gerida a informação mais resumida do plugin, ou seja, a informação apresentada no pop-up, que surge quando se carrega no ícone. Aqui definimos os valores para a tabela estruturada no **result.html**, quando a avaliação é realizada. Este script também é responsável por ativar o painel lateral (**sidebar.js**), no qual definimos todas as bibliotecas necessárias (de CSS e JQuery) para a execução do `sidebar.js`.

SIDEBAR

Nesta secção, como já foi indicado antes, dá acesso às informações detalhadas dos critérios e das técnicas avaliadas.

Aqui a página avaliada é transformada num iframe, para que se possa manipular os elementos do DOM da página, sendo assim possível a introdução de um <div> para a construção do nosso painel lateral.

A estrutura da nossa página baseia-se em tabelas dinâmicas com três botões que filtram o conteúdo. Dividimos a informação por “all”, “pass”, “fail” e “warning”, daí existirem funções distintas para cada tipo. Em cada função criamos as tabelas respetivas e filtramos com o que é pretendido, inserindo uma linha por critério. Cada critério é verificado pelo seu nome, para depois poder ser associado à sua respetiva descrição. Após carregar num dos critérios, passamos a apresentar as técnicas, obtendo o código fonte através do localStorage. O processo de apresentação destas é semelhante à dos critérios, com a exceção de que é inserida uma nova tabela, com os conteúdos, da tabela já existente.

Existe uma série de variáveis que guardam a informação associada a cada critério e técnica (como por exemplo, nome, descrição, resultados), que servem para mais tarde mostrar a informação relativa a cada um/a.

Relativamente ao destaque dos elementos na página, quando o utilizador provoca esta ação, estes são identificados por um ID, que já trazem à priori. Isto significa que os elementos que não têm ID não aparecem destacados. A nossa solução é inserir IDs próprios de forma a poder identificar todos os elementos do DOM, porém este processo exige que tanto o DOM que é digerido pelo plugin como o DOM que é processado pelo próprio QualWeb, tenham IDs do UbiWAN inseridos. Também existem casos em que as falhas encontradas representam elementos em falta no código fonte, o que leva a ser necessário termos de inserir no DOM as tags respetivas para que possam ser destacadas. Contudo este último processo irá ser introduzido mais tarde, como trabalho futuro.

De seguida apresentamos três tabelas a detalhar as variáveis, os IDs principais usados e as funções. Os IDs são importantes porque a implementação é feita em JQuery e todo o CSS está embebido no código, assim a única forma de manipular os elementos do painel lateral é identificá-los com o atributo “id”.

O facto de detalharmos, nestes termos, apenas o sidebar.js é devido à sua maior complexidade, comparativamente às outras duas componentes; envolvendo mais código e mais trabalho aplicado.

Tabela 1: Variáveis

Variável	Descrição
tecs/crit[]	nome da técnica/critério no array
tecs/crit[].url	URL de referência à documentação do WCAG 2.0
crit[].desTitle	título da descrição da técnica/elemento
tecs/crit[].des	descrição da técnica/elemento
tecs/crit[].countW	contador para o número de warnings
tecs/crit[].countP	contador para o número de técnicas/critérios que passaram
tecs/crit[].countF	contador para o número de técnicas/critérios que falharam
results	guarda os resultados obtidos da estrutura JSON
actualOut	estado do output actual (passed, failed, warning e tudo) serve para controlar o tipo de conteúdo que surge nas tabelas (ou seja, que é filtrado)
tecAux/tecElem/objec	técnica
tecElem.file	nome do ficheiro onde ocorre uma falha (0 se estiver embebido no código HTML ou URL se estiver num ficheiro externo de CSS)
tecElem.linenum;	linha onde ocorre a falha
elemm/tecElem.elem	elemento onde ocorre a falha
tecElem.src	código CSS onde ocorre a falha
tecElem.result	resultado da técnica (se passou ou falhou)
objec.technique	nome da técnica que detetou a falha
elemm[0]	tag do elemento
elemm[1]	código fonte, nomeadamente os atributos CSS
cb[]	array de todas as checkboxes que foram accionadas
hl[]	array de todos os IDs("id=") dos elementos que estão destacados
border[] ... bgc[]	arrays que guardam as alterações de CSS na página permite que seja possível ver vários elementos destacados (e.g. border[] guarda todas as alterações nas bordas, bgc[] guarda todas as alterações na cor do fundo)

Id_number	Número do ID gerado para os elementos da página sem ID
------------------	--------------------------------------------------------

Tabela 2: IDs

IDs	Descrição
ubiwan_page	iframe(da página avaliada) gerado pelo plugin
ubiwan_sidebar	elemento que engloba todo o painel lateral
ubiwan_opt	elemento dos botões(topo da página)
ubiwan_result	elemento onde se apresentam as tabelas
content_c	ID maioritariamente usado para inserir conteúdos num determinado elemento criado para o painel lateral (aparece em vários sítios, por isso ter em conta o código procedimental)
clear	botão para limpar os destaques da página
back_back	botão para voltar atrás
close_sidebar	botão para fechar o painel lateral
all	botão para filtrar todos/as critérios/técnicas
passed	botão para filtrar critérios/técnicas que passaram
failed	botão para filtrar critérios/técnicas que falharam
warning	botão para filtrar critérios/técnicas que têm aviso
evaluating	imagem de loading
"b_'+c+'"	identifica o critério c
"div_'+tecAux+'_'+c+'"	identifica o elemento tr da técnica tecAux do critério c (header da técnica)
"div_'+tecAux+'_'+c+'-2"	identifica o elemento div (dentro do tr) da técnica tecAux do critério c (corpo da técnica)
"b_'+tec+'_'+c+'_checkbox_'+i_count+'"	identifica a checkbox da técnica tec do critério c com um contador i_count
"b_'+tec+'_'+c+'_code_'+i_count+'"	identifica a secção do código da técnica tec do critério c com um contador i_count

Tabela 3: Funções

Função	Funcionalidades	Callbacks
...onMessage.addListener	Trata de escutar as mensagens vindas do result.js com o estado da página e/ou com o resultado da avaliação (esta função irá desaparecer com o uso do localStorage)	organize() sidebar()
organize	Transforma a página avaliada num iframe e insere um elemento <div> para a introdução do painel lateral	-----
sidebar	Constrói a página dos critérios, com uma tabela dos resultados obtidos e uma série de botões funcionais	aparecerT/P/F/W()
aparecerT/P/F/W	Faz aparecer na tabela os resultados dependendo do filtro selecionado(pass, fail ou warning)	createTable() getCriterio()
createTable	Cria a tabela dos critérios	-----
getCriterio	Dependendo do filtro, seleciona os critérios que são apresentados na tabela	aparecerCriterion() generateIds()
aparecerCriterion	Constrói a primeira tabela que apenas enumera as técnicas(semelhante ao dos critérios)	getSpecial()
generateIds	Onde está a ser desenvolvido a adição de IDs aos elementos por destacar	generateIds2()
generateIds2	Função recursiva que adiciona IDs aos elementos e aos seus filhos, que não tenham ID	generateIds2()
getSpecial	Constrói a segunda tabela das técnicas, mas agora com os detalhes do erro (descrição, linha e código fonte) (tabela que aparece ao clicar numa técnica)	getCorrectTec() aparecerTec()
aparecerTec	Faz aparecer a técnica selecionada	tecValues()
getCorrectTec	Passo intermédio para selecionar a técnica correta. Existem técnicas que estão agregadas e que precisam de ser identificadas apenas como uma (tal como acontece com todas as outras) (c121314 e c20242628)	-----
tecValues	Faz aparecer os detalhes da técnica selecionada	addHighlight() removeHighlight()
addHighlight	Realça o erro selecionado na página que está a ser avaliada, adicionando aos arrays as modificações de CSS dos elementos.	-----
removeHighlight	Remove dos arrays o CSS e o ID do erro desseleccionado ou caso tenham limpo o ecrã todo, retirando o elemento destacado na página.	-----

OUTROS FICHEIROS

MANIFEST.JSON

Ficheiro responsável por configurar o plugin, pois é aqui que é definido o nome, a versão, o modo como é apresentado inicialmente (pop-up ou painel), os recursos e permissões. O painel aqui mencionado é diferente da noção do que é implementando, pelo que este posiciona o plugin ao nível da consola do browser, em vez de abrir na própria página.

No nosso caso, temos em *browser_action* definido o modo pop-up, onde redireciona para o ficheiro **index.html**; nas permissões definimos o acesso aos separadores (“tabs”) e os tipos de url(“*:/*/*”), neste caso todo o tipo; a razão pelo qual o pop-up é usado, tem haver com o facto do painel do browser não processar JavaScript.

INDEX.HTML

Esta é a primeira página html que aparece, no pop-up. Esta simplesmente redireciona para outras duas páginas, **error.html** em caso de a avaliação não ter sido bem-sucedida ou caso contrário para o **result.html**. Este redireccionamento é efetuado no **index.js**.

RESULT.HTML

Nesta página é onde é apresentada a tabela das percentagens e onde permite navegar até às opções(**options.html**) ou até aos detalhes(**sidebar.js**). Este acede ao localStorage pela chave “ubiwan_result” para aceder ao resultado da avaliação.

OPTIONS.HTML

Nesta página é mostrado ao utilizador várias opções para efetuar a sua avaliação, tais como, qual o contexto a ser avaliado (*Desktop* ou *Mobile*) e se pretende avaliar o CSS e/ou o JavaScript.

INDEX.JS

Aqui é efectuado o redireccionamento da página para o **error.html**, em caso de erro, ou para o **result.html** em caso de sucesso da avaliação.

ERROR.HTML

Nesta página é apresentada uma lista de possíveis erros pelo qual não foi possível efetuar a avaliação.

OPTIONS.JS

Gere o estado das configurações feitas pelo utilizador na página **options.html**, guardando-o no localStorage com três itens: “evaluate_context”, “evaluate_js” e “evaluate_css”.

AVALIAÇÃO

Como se mantém incógnita as vantagens do plugin em relação ao website, pretendemos entender com base em testes de utilizadores, se o uso do plugin traz realmente vantagens. No meio deste processo, também queremos entender se a forma como a interface está implementada é suficiente ou se apresenta lacunas que possam ser desenvolvidas e/ou melhoradas.

Assim foram realizados 8 testes para a usabilidade, com pessoas que costumam programar aplicações web (a nível académico), mas que pouco ou nada entendem de acessibilidade Web, de forma a comparar a interface da página no QualWeb com o plugin UbiWAN. Cada utilizador teve de realizar 8 tarefas similares para cada ferramenta para que pudesse haver um meio de comparação.

Como tarefa principal, pedimos que a página <http://www.fc.ul.pt>, fosse avaliada afim de concretizar as tarefas seguintes. De seguida, era pedido que nos indicassem alguns valores, só para entender se o cabeçalho resumido era entendido corretamente. Para cada técnica que encontrassem tinham de justificar a razão pelo qual passou ou falhou nos testes, assim para verificarmos se as descrições existentes eram suficientes ou corretas. Por fim, perguntámos a preferência entre ambos.

Na tabela seguinte apresentamos os resultados de cada tarefa nas duas ocasiões numa escala de 1 a 3, em que 1 significa que não entendeu/não fez a tarefa e 3 significa que realizou a tarefa com sucesso.

Tabela 4: Tabela de Resultados dos Testes

[illegible]

Encontre o elemento na página(h44)	--	2	--	3	--	2	--	2	--	2	--	2	--	3	--	2
Preferência		x	x		x			x		x		x		x		x

RESULTADOS

Com base nos testes podemos concluir que existiram algumas diferenças, em geral positivas, no nível de realização das tarefas. É preciso ter em conta que a maior parte dos utilizadores desta experiência, não entende a nomenclatura do WCAG 2.0, o que por um lado preenche o perfil ideal, pois temos de pensar nos programadores que são principiantes nesta matéria.

Em relação à primeira tarefa, a maioria (7 em 8) conseguiu avaliar a página mais facilmente pelo UbiWAN, pelo que houve quem comentasse que a avaliação ao ser automática era uma vantagem. Em relação ao QualWeb, as pessoas mostraram alguma dificuldade em encontrar o formulário de avaliação; após ter começado a avaliação não entenderam se já tinha ou não iniciado, isto porque o indicador atual do progresso é pouco perceptível.

Não foram encontradas quaisquer relevâncias nas tarefas 2 e 3, visto que a maioria conseguiu concretizar todas com sucesso tanto no QualWeb como no UbiWAN. Porém houve um comentário negativo relativo à disposição dos elementos na tabela do QualWeb.

Em grande parte houve dificuldades em descobrir no QualWeb a técnica pedida, pois nem o filtro de estados (passou, falhou ou warning) nem a disposição dos resultados (acordeão que não destaca as sub-componentes), nem o ícone “+” eram perceptíveis, o que fez com que o utilizador se senti-se perdido. Houve casos com o UbiWAN em que também não era claro a forma como se divulgava mais informações, e também foi criticado que falta uma sinalização de quando é que é aberto/fechado os acordeões. Também houve alguma dificuldade em entender a diferença entre os critérios e as técnicas avaliadas. Acrescentando que o nome das técnicas gerou alguma confusão, por exemplo muitas das vezes eram encontradas técnicas de CSS no qual o nome começava por “c” (e.g. c9) e consideravam que eram critérios; desta forma pudemos também concluir que não existiu um entedimento sobre a diferenças das classes (h44 é uma técnica de HTML e c9 é uma técnica de CSS).

Outro problema que “cortava” a navegação era o facto dos nomes dos critérios e das técnicas terem uma hiperligação para a página respetiva do W3C, e na maior parte das vezes as pessoas carregavam no nome e não na linha.

Apesar de termos pedido para encontrarem uma determinada técnica, é de salientar que nós não consideramos que seja uma tarefa frequentemente realizada, porque não é suposto os programadores saberem todas as técnicas decor e saberem exactamente o que procurar, pelo que a maior parte das vezes estão “às cegas” com o resultado da avaliação. Decidimos testá-la porque queríamos que houvesse uma forma de controlo para poder comparar as reacções/opiniões dos utilizadores da mesma técnica nas duas situações. Ainda no contexto da procura de uma técnica, em relação ao QualWeb este disponha de opções de apresentação que facilitavam a procura por critério ou técnica, ao contrário do UbiWAN.

Quando se pedia para justificar, por vezes não conseguiam responder, porque as descrições eram ambíguas ou porque realmente não entendiam o contexto delas, ou seja, a descrição não correspondia ao estado do elemento (se passou ou falhou), por exemplo, num elemento que tenha passado a descrição mostrada era negativa. Houve também situações em que não encontravam a técnica ou quando a encontravam não entendiam que existia mais informação associada ao critério/técnica. Ainda assim, houve algumas melhorias com o uso do UbiWAN,

pois não exige tanto a navegação pela página e porque existe uma certa divisão entre critérios e técnicas.

Na última tarefa, relativa ao elemento, pudemos observar que existiram complicações no reconhecimento dos elementos destacados, isto porque as pessoas estavam mais focadas no painel lateral e não entendiam que existiam alterações a decorrer na página. Isto indica-nos que faltam avisos no painel lateral de que existem alterações ou que a forma como avisamos não é suficiente, “see Code”, quando na verdade mostra o código e apresenta o destaque do elemento.

Quando foi necessário que os utilizadores fizessem a avaliação com o UbiWAN, notaram que havia um bug aquando aparecia o pop-up e queriam passar para o painel lateral, pois este não desaparecia automaticamente.

Foi feito um pequeno inquérito sobre qual das duas interfaces era a mais usável e 6 em 8 preferiram o UbiWAN, justificando que era mais simples de usar, mais directo e o facto da página estar ao lado dos resultados também é mais vantajoso. Dos dois utilizadores que optaram pelo QualWeb indicaram que ter a janela dividida não era, na opinião deles, a forma mais fácil de usar pois condicionava a visualização total da página.

De uma forma geral, os utilizadores afirmaram que a etiqueta apresentada no ícone do plugin era bastante perceptível e útil.

Por fim, houve uma série de sugestões dadas pelos utilizados tais como a implementação de um tutorial/glossário ou de um campo que ajude o utilizador a entender as noções de contexto(WCAG 2.0), mudar a cor da etiqueta(vermelha) que aparece no plugin, o que sugere que é a percentagem de testes que falharam ao contrário do que é suposto (que seria os que passaram), filtros para distinguir as técnicas dos critérios e alterar o nome do botão “Clear”.

TRABALHO FUTURO

Como trabalho futuro, gostaríamos de incluir mais opções para filtrar resultados, tais como, o nível de conformidade (A, AA, AAA), as classes de técnicas existentes e que são aplicadas (com base no WCAG 2.0), isto é a classe H (para HTML), classe C (para CSS) e assim sucessivamente e com base nos testes efectuados, introduzir um filtro que distinga os critérios das técnicas. Também está em aberto a solução para a melhor forma de apresentar os conteúdos destacados na página, tendo em ideia a realização de pequenos pop-ups com as soluções e os erros respetivos.

Como este plugin só é dirigido para o Chrome, futuramente, iremos também implementar para outros browsers, mais especificamente para o IE (Internet Explorer), pelo que foi verificado² anteriormente que é o browser mais utilizado entre pessoas com deficiências para a navegação na Web, não excluindo os outros. Porém não foi desenvolvido primeiramente para o IE, pelo que exigia conhecimentos que na altura não tínhamos e que iria gerar mais complicações.

Um dos grandes objetivos, também por desenvolver, passa pela correção automática do código, de forma a lucidar ao utilizador de quaisquer ambiguidades que as regras do WCAG possam trazer e oferecendo, assim, uma solução mais directa.

Juntamente com o UbiWAN, é requerido melhorias no servidor do QualWeb e da sua página Web, incluindo novas técnicas para a avaliação (acrescentando às 44 já implementadas) e otimização do código já desenvolvido.

² <http://webaim.org/projects/screenreadersurvey5/> consultado em Outubro de 2015