

Daniel Augusto Rebelo Cunha

KLM para medir as variações de performance das aplicações web



Daniel Augusto Rebelo Cunha

KLM para medir as variações de performance das aplicações web

Projeto de Mestrado

Mestrado em Sistemas e Tecnologias da Informação para as Organizações

Professor Doutor Rui Pedro Monteiro Amaro Duarte

Professor Doutor Carlos Augusto da Silva Cunha



Dezembro de 2020

Aos meus pais, por tudo, desde sempre.

RESUMO

O recente aumento da oferta de aplicações de software contribuiu para o aumento da exigência dos utilizadores, o que originou um acréscimo da complexidade dessas aplicações. Tal contribuiu para a adoção de mecanismos de automatização do processo de testes de software, de forma a reduzir erros de codificação e encurtar o tempo necessário para disponibilizar uma nova versão da aplicação ao utilizador. Atualmente, automatizar o processo de testes de aplicações é um processo já bem estudado e suportado por muitas ferramentas. No entanto, a avaliação da usabilidade de uma aplicação requer soluções que permitam determinar, antecipadamente, o tipo de melhorias que podem ser necessárias na aplicação sem a execução de um elevado número de testes com utilizadores.

A performance é um fator tão importante numa aplicação web como a avaliação da sua usabilidade. O número de interações cliente-servidor, bem como o volume de dados transferido em cada interação estão relacionados com o desempenho da aplicação web, pelo que a satisfação do utilizador está dependente destes fatores.

O presente documento aborda a automatização da análise do impacto das alterações do desenho da aplicação no utilizador, assim como a análise de dependências entre interações cliente-servidor. Através da implementação do modelo *Keystroke Level Model* (KLM) são medidas interações não planeadas do utilizador, sendo posteriormente identificados padrões com base nos operadores KLM e nos tempos de cada operador. Com base nestes padrões são gerados indicadores de desempenho que são analisados pelo designer de interfaces, possibilitando alterações no desenho da aplicação.

De forma a facilitar a análise da performance das aplicações web, é também proposta uma ferramenta que identifica os recursos que podem ser transferidos do servidor com a antecedência necessária de forma a evitar atrasos na interação do utilizador com a aplicação. No sentido de priorizar o carregamento de recursos é identificada uma ordem para o seu carregamento com base nas diferenças temporais entre eles.

ABSTRACT

The recent increase in the availability of software applications has contributed to the increase in user demand, which has led to an increase in the complexity of these applications. This contributed to the adoption of automation mechanisms for the software testing process, in order to reduce coding errors and shorten the time needed to make a new version of the application available to the user. Currently, automating the application testing process is a well-established reality and supported by many tools. However, the evaluation of the usability of an application requires solutions that allow to determine, in advance, the type of improvements that may be necessary in the application without the execution of a high number of tests with users.

In a web application, its performance is a factor as important as the evaluation of its usability. The number of client-server interactions, as well as the volume of data transferred in each interaction are related to the performance of the web application, so user satisfaction is dependent on these factors.

This work deals with the automation of the analysis of the impact on the user of changes in the design of the application, as well as the analysis of dependencies between client-server interactions. Through the implementation of the Keystroke Level Model (KLM), unplanned user interactions are measured, and patterns are subsequently identified based on the KLM operators and the times of each operator. Based on these standards, performance indicators are generated and analyzed by the interface designer, allowing changes in the design of the application.

In order to facilitate the analysis of the performance of web applications, a tool is also proposed that identifies the resources that can be transferred from the server with the necessary advance in order to avoid delays in the user's interaction with the application. In order to prioritize the loading of resources, an order is defined based on the temporal differences between them.

PALAVRAS CHAVE

Keystroke level model

Interações

Automatização

Operadores

Usabilidade

Desempenho

KEY WORDS

Keystroke level model

Interactions

Automation

Operators

Usability

Performance

AGRADECIMENTOS

Um trabalho de mestrado é um longo processo, que inclui uma trajetória permeada por muitos percalços pelo caminho. Trilhar este caminho só foi possível com o apoio e força de várias pessoas, a quem dedico este projeto.

Gostaria de agradecer em primeiro lugar aos meus pais, os pilares importantes da minha educação, da minha vida pessoal e profissional. Sempre me deram todas as condições para que pudesse ter sucesso na concretização de todas as metas existentes durante o meu percurso académico.

Um agradecimento especial ao meu irmão que sempre me acompanhou durante todo o meu percurso académico e que em muitas situações, sem ele não seria possível a concretização de muitos dos objetivos alcançados.

À minha namorada, Catarina Loureiro, pela ajuda, paciência e apoio incondicional que sempre demonstrou durante esta etapa.

E, como não podia deixar de ser, um muito obrigado aos meus orientadores, Professor Doutor Rui Duarte e Professor Doutor Carlos Cunha, da Escola Superior de Tecnologia e Gestão de Viseu pela confiança que sempre depositaram em mim e acima de tudo pela disponibilidade e apoio sempre prestado.

Um muito obrigado a todos, porque sem o contributo de todos vocês nada disto teria sido possível.

ÍNDICE GERAL

ÍNDICE GERAL	xiii
ÍNDICE DE FIGURAS	xvi
ÍNDICE DE Tabelas.....	xviii
ABREVIATURAS E SIGLAS	xx
1. Introdução.....	1
1.1 Contextualização.....	2
1.1.1 Modelos Preditivos	3
1.1.2 Performance.....	4
1.2 Motivação	5
1.3 Objetivos e Abordagens.....	6
1.3.1 Modelo de interpretação de ações não planeadas por parte dos utilizadores	6
1.3.2 Criar um sistema de indicadores de desempenho para os designers	7
1.3.3 Desenvolvimento de uma ferramenta para análise de dependências entre recursos web	7
1.4 Contribuições Principais	8
1.5 Estrutura do documento	8
2. Estado da arte.....	9
2.1 Modelo KLM	9
2.2 Aplicações do Modelo KLM	11
2.3 Performance e QoS	15
2.4 Performance e Usabilidade	16
2.5 Conclusão.....	18
3. Interação não planeada usando KLM	19
3.1 Descrição de alto nível da <i>framework</i> não planeada.....	19
3.2 Extração da <i>string</i> KLM	20
3.3 Estatísticas e Detecção do tipo de interação	23
4. Performance.....	31
4.1 Identificação do problema.....	31

4.2	Processo de Recolha de Dados.....	32
4.3	Ferramenta de Análise de Dependências.....	34
4.4	Análise de Dependências entre Recursos.....	36
4.4.1	Análise Individual de Recursos.....	36
4.4.2	Agrupamento de Recursos.....	39
4.5	Priorização de Transferência de Recursos.....	41
5.	Conclusões e Trabalho Futuro.....	43
5.1	Conclusões.....	43
5.2	Trabalho Futuro.....	44
	Apêndice 1.....	52
	Anexo 1 – Formulários de registo das aplicações web.....	62
	Anexo 2 – Resultados da diferença temporal entre recursos.....	74

ÍNDICE DE FIGURAS

Figura 1 - Visão geral da <i>framework</i> com ações não planeadas por parte do utilizador.....	20
Figura 2 - Tags XML da caixa de texto email: o utilizador clica	21
Figura 3 - Conversão do XML: do ficheiro XML para a <i>string</i> KLM.....	22
Figura 4 - Transformação <i>string</i> KLM em estatísticas com uma interação baseada no rato e no teclado	24
Figura 5 - Transformação <i>string</i> KLM em estatísticas com uma interação baseada no teclado	25
Figura 6 - Formulários de registo usados para testes em diferentes cenários de interação.....	28
Figura 7 – Indicação do tipo de interação efetuada pelo utilizador no website [S24]	30
Figura 8 - Visão geral da solução implementada ao nível da performance	33
Figura 9 - UI da Ferramenta de Análise de Dependências	34
Figura 10 - Informação acerca de cada recurso na Ferramenta de Análise de Dependências .	35
Figura 11 - Relação ente dois recursos na Ferramenta de Análise de Dependências	36
Figura 12 - Registo de um novo utilizador no website <i>BBC</i>	37
Figura 13 - Registo de um novo utilizador no website de compras online <i>Walmart</i>	38
Figura 14 - Pesquisa de notícias no website <i>BBC</i>	38
Figura 15 - Pesquisa de produtos na loja online <i>Walmart</i>	39
Figura 16 - Análise da probabilidade relativamente à dimensão das combinações.....	40
Figura 17 - Análise das diferenças temporais entre recursos.....	42

ÍNDICE DE TABELAS

Tabela 1- Operadores KLM e previsão dos tempos de execução (Card S. K., 1980)	10
Tabela 2- Diferenças de performance entre 3 tipos de aplicações	16
Tabela 3- Resultados não planeados mostrando médias e, entre parêntesis, desvios padrão ..	23
Tabela 4 - Estratégias de interação seguidas em dez fontes, e percentagens correspondentes de tempo gasto e número de operadores usados	26
Tabela 5 – Padrões detetados consoante o tipo de interação	29
Tabela 6 – Padrões obtidos em 10 sites diferentes.....	29
Tabela 7 – Websites usados nos testes e correspondentes contextos de interação	52
Tabela 8 – Websites utilizados para os testes alusivos à identificação de padrões.....	52
Tabela 9 – Recursos estudados para o registo no website <i>BBC news</i>	53
Tabela 10 - Recursos estudados para a pesquisa no website <i>BBC news</i>	54
Tabela 11 - Recursos estudados para o registo no website <i>Walmart</i>	57
Tabela 12 - Recursos estudados para a pesquisa no website <i>Walmart</i>	57

ABREVIATURAS E SIGLAS

KLM	Keystroke Level Model
GOMS	Goals Operators Methods and Selection Rules
UI	User Interface
UX	User Experience
API	Application Programming Interface
PWA	Progressive Web App
QoS	Quality of Service

1. Introdução

Um dos maiores desafios para impulsionar melhorias de usabilidade através de métricas, é que, quando a usabilidade é medida, é tarde demais para alterar a interface. Esse é, essencialmente, o caso do tempo que o utilizador demora a realizar cada tarefa, pois, normalmente, apenas é refletido em aplicações totalmente funcionais e ativas. Na melhor das hipóteses, a maioria das atividades relacionadas com a avaliação de usabilidade de protótipos que se encontram numa fase inicial, concentram-se em encontrar e consertar problemas (que é uma atividade essencial). Muitas vezes, esses protótipos carecem de funcionalidades, pelo que, mesmo que se quisessem efetuar alguns testes com utilizadores, seria difícil verificar o quão confiáveis seriam as estimativas de tempo geradas para cada tarefa. Assim, a possibilidade de saber antecipadamente que um determinado design pode ser executado em menos tempo (ou que pelo menos não leva mais do que um determinado tempo), seria o ideal. Desta forma, seria possível inferir mais rapidamente se um design de uma aplicação não vai ser usável ou, pelo menos, ter-se-ia uma ideia sobre as melhorias que seriam necessárias introduzir.

Atualmente, a procura de aplicações por parte dos utilizadores está a aumentar e a interface entre o utilizador e o sistema desempenha um papel importante na aceitação destes, bem como na confiança e na satisfação do utilizador (Pommeranz, 2012). Ao nível da interface, a maior procura concentra-se na usabilidade, performance, qualidade de serviço e no design da interface (Moulin, 2002). Quando a interface é bem compreendida pelos utilizadores, é estabelecida uma boa relação entre eles e o sistema. Caso contrário, os utilizadores tendem a ficar frustrados e mais intolerantes aos erros, levando a que, na maior parte dos casos, estes abandonem o sistema. Ratcliff (Ratcliff, 2019) refere que 69% dos visitantes de uma aplicação recusam-se a completar um formulário se este contiver demasiados campos para preencher. Além disso, 39% dos utilizadores deixam de usar a aplicação se o conteúdo desta demorar a carregar.

Segundo o estudo efetuado por Lara Hogan (Hogan, 2014), a maioria dos utilizadores abandona uma aplicação web quando o tempo de espera de carregamento ultrapassa os 2 segundos. Por

sua vez, quando o tempo de carregamento da página ultrapassa os 3 segundos, cerca de 40% dos utilizadores tendem a abandonar a aplicação. Para além deste facto, 85% dos utilizadores que utilizam com frequência aplicações móveis tendem a ser mais exigentes e esperam que o carregamento das aplicações web nesses mesmos dispositivos seja igual ou inferior ao carregamento da mesma aplicação web, nos seus computadores pessoais.

1.1 Contextualização

Tal como já foi referido anteriormente, a exigência dos utilizadores relativamente às aplicações web está relacionada não só com a performance, mas também com a usabilidade dessas aplicações.

O conceito de Usabilidade tem vindo a tornar-se cada vez mais um fator fundamental no desenvolvimento de aplicações (Nadeem Ahmad, 2015), (Nielsen J. , 2012), (Brad A. Myers, 2016), (Szymański J.M., 2015) e (Shitkova, 2015). De facto, a usabilidade é uma área da Interação Humano-Computador (IHC) em que são identificadas necessidades dos utilizadores para garantir que um produto possa atingir objetivos específicos de forma eficaz e eficiente, o que resulta em satisfação e sucesso geral.

A usabilidade é também parte integrante do termo mais lato Experiência do Utilizador (UX). A UX refere-se à facilidade com que pessoas usam uma ferramenta ou objeto para realizar uma tarefa específica. Na IHC, a UX refere-se à facilidade e simplicidade com que uma interface pode ser utilizada. A Interface do Utilizador (UI) combinada com a UX permite determinar a usabilidade de um determinado sistema (Michael O. Leavitt, 2006).

O design de interfaces é uma das tarefas mais críticas na análise do UX e da UI, que tem como objetivo melhorar a UX, tornando a interface mais intuitiva e simples. Posto isto, um dos aspetos a ter conta é o de prever qual será o comportamento dos utilizadores e assim reduzir o tempo despendido em cada interação através da introdução de melhorias ao desenho da interface. De forma a ser possível alcançar-se isso, existem dois grupos de métodos que permitem avaliar a usabilidade de uma aplicação: métodos empíricos e métodos analíticos. Enquanto que o primeiro tipo de métodos envolve utilizadores na avaliação das aplicações, o segundo usa técnicas para prever o comportamento do utilizador, sem o envolver na avaliação. Os métodos empíricos seguem um processo de desenho iterativo baseado numa sequência de tarefas tipificada e sequencial: análise de requisitos, onde é feita a análise de tarefas e os cenários do problema; desenho do modelo conceptual e o desenvolvimento de cenários de atividade; prototipagem, através de storyboards e cenários de interação; avaliação, onde é feita a avaliação com utilizadores, permitindo determinar o nível de usabilidade do sistema.

Os métodos analíticos não envolvem utilizadores e um dos exemplos destes métodos são os baseados em avaliação preditiva de tarefas, seguindo vários critérios de análise. Um modelo preditivo usa dados históricos de várias fontes, normaliza esses dados de forma a impedir possíveis anomalias e processa-os de forma a seguir um formato adequado que facilite a sua análise. Exemplo disso é a família de modelos preditivos que compara e avalia Objetivos,

Operadores, Métodos e Seleção (tipicamente conhecido pela sigla GOMS que representa *Goals, Operators, Methods and Selection*) de desempenhos, qualificados e sem erros (Card S. K., 1983). Neste modelo, é descrita a estrutura cognitiva do utilizador em quatro componentes, que são: os objetivos a atingir com a interação, os operadores usados nessa interação, o método que o utilizador seguiu e um conjunto de regras de seleção para escolher os métodos usados para atingir os objetivos. Este modelo GOMS possuiu quatro variantes com diferentes complexidades e atividades associadas: *Keystroke Level Model* (KLM), CMN-GOMS, NGOMSL, CPM-GOMS e SGOMS.

1.1.1 Modelos Preditivos

Os modelos preditivos focam-se na hierarquia de tarefas compostas por ações e operadores cognitivos. O modelo mais simples da família GOMS é o KLM, que prevê o tempo de execução de uma tarefa usando o rato e o teclado como entradas principais (Card S. K., 1980). O GOMS está diretamente relacionado com o processamento mental e decomposição hierárquica de objetivos, ou seja, define objetivos relacionados com interações do utilizador e divide esses mesmos objetivos em tarefas e sub-tarefas. Para ser possível efetuar esta análise, é necessário identificar o objetivo definido, perceber a forma como este pode ser atingido (definindo várias tarefas) e aplicar os resultados à UI. O resultado pode produzir medidas de desempenho precisas para utilizadores experientes, embora sejam menos precisas para utilizadores inexperientes. De salientar que os modelos preditivos dependem de conjuntos de dados históricos, logo é necessário ter mecanismos de registo de eventos do sistema onde as tarefas irão ser realizadas, de modo a poder ser feita uma previsão.

O KLM modela interações ao nível do dispositivo, como cliques no rato ou o premir das teclas de um teclado. Consegue prever o desempenho dos utilizadores na realização de tarefas, e por isso, o seu principal objetivo é determinar os tempos de execução e conclusão de uma tarefa por parte destes num design da UI específico. Este modelo tem um conjunto de operadores físicos e mentais que possuem valores estimados, permitindo ao programador fazer uso destes para uma análise mais concreta e fidedigna do desempenho da UI (Card S. K., 1980). O modelo KLM normalmente é utilizado em conjunto com o modelo GOMS, na vertente das métricas dos métodos deste modelo e tem sido amplamente utilizado para medir o desempenho de utilizadores experientes e a sua utilidade tem sido demonstrada ao longo dos anos (Katsanos C. T., 2006) (John, 2004) (Blackmon, 2005).

Um dos maiores desafios no que toca ao design de interfaces é ajudar os designers a melhorar as suas interfaces sem terem de configurar um elevado número de sessões de testes com utilizadores, como é característica dos modelos empíricos. Este processo exige uma logística pesada e custos associados que, mesmo sendo uma parte importante do processo de design, o tempo consumido para produzir resultados é elevado. Além disso, os designers de interfaces tomam pequenas decisões ao longo do caminho, que podem ter impacto na interação com o utilizador e precisam de ferramentas que lhes permitam medir a qualidade das suas interfaces.

De acordo com Al-Megren (Al-Megren, 2018) é de extrema importância criar métodos que utilizem o modelo KLM de forma a que seja possível fornecer orientação e sugestões aos designers para que estes consigam introduzir mudanças nos seus designs que visem a melhoria da qualidade da interação com o utilizador, reduzindo o tempo necessário para completar uma tarefa.

Uma das vertentes deste projeto foca-se numa aplicação do KLM que usa a interface criada pelo designer para fornecer *feedback* sobre a qualidade da mesma e disponibilizar indicadores sobre como melhorar a qualidade da interação. Para tudo isso ser alcançado, é proposto um algoritmo baseado no modelo KLM que extrai uma *string* KLM da interação, é detetado um tipo de interação e, a partir daí, são identificados padrões de comportamento na interação que aumentam o tempo de execução das tarefas do utilizador e que originam uma interação pouca usável.

1.1.2 Performance

Apesar dos modelos preditivos incorporarem a avaliação do desempenho das aplicações, eles assumem que o tempo de resposta do sistema é estável, sendo muitas vezes negligenciado, não admitindo que o tempo de carregamento das páginas pode ser demorado. Este fato pode constituir um problema, uma vez que a aplicação faz frequentemente solicitações de recursos ao servidor. Sendo a latência da rede e o tempo de resposta do servidor variável em função de um conjunto de condições muitas vezes não previsíveis, o tempo de resposta do sistema pode ser afetado consideravelmente. Tal sucede com o modelo GOMS, em que o tempo de resposta do sistema, W , é na maioria das vezes desprezado.

A usabilidade de uma aplicação não está apenas relacionada com a estrutura das suas interfaces, mas também com a performance de todo o sistema. Segundo Jean Bolot (Jean-Chrysostome Bolot, 1996), o desempenho de uma aplicação está diretamente relacionado com a performance do servidor onde está essa mesma aplicação, com o número de interações cliente-servidor, com o volume de dados transferido em cada interação e com as condições existentes na rede onde serão efetuados os pedidos ao servidor.

Numa interação cliente-servidor, existe troca de dados entre ambos, pelo que, quanto maior for o volume de dados transferido e o número de interações entre estes, maior é a probabilidade de interferência de potenciais problemas de performance no sistema e, por consequência, na experiência do utilizador. Uma das formas de reduzir o número de interações cliente-servidor, é a utilização da memória cache. Este tipo de memória permite o armazenando os dados que são mais vezes manipulados pelo utilizador, aumentando assim a performance da aplicação e permitindo o acesso mais rápido aos dados por parte do utilizador (Michael J. Franklin, 1997). Portanto, é fundamental entender a relação entre a performance e a usabilidade durante a fase de design das aplicações.

Devido ao facto de o desempenho ser vital para o sucesso e lucratividade de qualquer aplicação, os sites estão a começar de utilizar um novo padrão chamado *Progressive Web Applications* (PWAs) que visa oferecer um alto nível de desempenho, independentemente do tipo de

dispositivo ou rede. As PWAs são aplicações web que oferecem ao utilizador uma experiência similar à das aplicações móveis nativas (A., 2018).

As PWAs podem ser utilizadas *offline*, facilitando a acessibilidade a estas por parte do utilizador. Esta característica é uma das mais relevantes e uma das suas maiores vantagens (A., 2018). Outra das suas funcionalidades é o facto de reduzir os problemas relacionados com o *download* das aplicações ou as posteriores atualizações, porque as PWAs ocupam muito menos memória quando comparadas com as aplicações nativas (Osmani, 2017).

Alguns dos maiores sites da atualidade já utilizam o conceito de PWAs nas suas aplicações web. Exemplos populares são: a rede social Pinterest (Osmani, 2017), a revista Forbes, o Twitter ou o site de compras e vendas online OLX (Zaina., 2018).

A performance e a qualidade de serviço (QoS) estão diretamente relacionadas. A QoS refere-se a qualquer tecnologia que faz o controlo e a gestão do tráfego de dados numa rede, de forma a reduzir a perda de pacotes, a latência e a instabilidade. Uma má qualidade de serviço traduz-se numa perda de utilizadores na aplicação que, conseqüentemente, origina carência no que toca a oportunidades de negócio (Menascé., 2002).

Nos últimos anos, a relação entre as características da qualidade de serviço da rede e a satisfação do utilizador tem sido alvo de estudo. No entanto, não tem sido tarefa fácil conseguir relacionar estes dois fatores (Shaikh, 2010). Um dos estudos efetuados foi, precisamente, a medição da qualidade do site da Universidade da Malásia (Dominic, 2010), avaliando assim a qualidade do serviço e a perspetiva do utilizador, no que toca à usabilidade da aplicação web.

1.2 Motivação

A principal motivação para a realização deste projeto assenta na falta generalizada de técnicas de apoio ao designer, de forma a verificar quais as interfaces que poderão ser mais sensíveis à performance de determinadas APIs. A utilização das funcionalidades do modelo KLM, no que toca à previsão do tempo que o utilizador tende a demorar a efetuar uma ação numa determinada aplicação e a relação existente entre a performance, a qualidade do serviço e o design das interfaces são outros pontos que motivaram a realização deste projeto.

Anna Bouch (Anna Bouch, 2000) constatou que, apesar das perceções dos utilizadores sobre a qualidade do serviço serem influenciadas por fatores relacionados com um determinado contexto, é possível correlacionar a qualidade do serviço na internet com ações subjetivas efetuadas pelos utilizadores, influenciando assim o design do sistema ou da aplicação web. Segundo os autores, apenas é possível maximizar a qualidade dos serviços da internet, integrando os requisitos da qualidade do serviço dos utilizadores com o design do sistema, identificando o lema que diz que “a qualidade está nos olhos de quem vê”.

Devido ao fácil acesso que os utilizadores têm aos dispositivos móveis, e ao aumento da sua utilização por parte destes, as PWAs tornaram-se, sem dúvida, uma das tecnologias mais utilizadas pelas empresas no desenvolvimento das suas aplicações web (Osmani, 2017) e

(Zaina., 2018). Devido a este facto, torna-se bastante importante conseguir relacionar o UX com a performance no contexto das PWAs.

A título de exemplo, o site Pinterest optou por recriar toda a sua aplicação web para dispositivos móveis, tornando-a uma PWA. Modificou todo o design das interfaces da sua aplicação web anterior, de modo a facilitar a interação e proporcionar uma melhor experiência de utilização, no que toca ao desempenho da aplicação, tendo como principal objetivo aumentar os registos de utilizadores no site. Com esta melhoria, obteve um aumento de 44% no que diz respeito à receita oriunda dos anúncios gerada pelos utilizadores e o tempo gasto por estes no site aumentou também em 40%, quando comparada com a antiga aplicação web para dispositivos móveis (Osmani, 2017).

1.3 Objetivos e Abordagens

Neste capítulo irão ser abordados os principais objetivos deste projeto, sendo que em cada subcapítulo seguinte serão descritos com maior detalhe a abordagem seguida para cada um deles.

- Modelo de interpretação de ações não planeadas por parte dos utilizadores: Como entrada, este modelo apenas requer uma descrição da tarefa que o utilizador irá efetuar e o URL da página web a testar. A partir daí, o KLM é aplicado e é gerada uma *string* KLM apenas considerando o objetivo a atingir pelo utilizador.
- Sistema de indicadores de desempenho para os designers: Com base na *string* KLM e nos operadores resultantes, são gerados indicadores para os designers, materializados numa ferramenta que os irá ajudar no redesenho da interface focando na redução do tempo de execução de uma tarefa por parte dos utilizadores.
- Desenvolvimento de uma ferramenta para análise de dependências entre recursos web: Com base nas interações cliente-servidor são identificados os recursos que podem ser obtidos do servidor com a antecedência necessária para evitar atrasos na execução da aplicação, assegurando a priorização e transferência proativa destes recursos.

1.3.1 Modelo de interpretação de ações não planeadas por parte dos utilizadores

Com este objetivo o que se pretende é simplificar a descrição da tarefa que o utilizador irá realizar e reduzir a carga de trabalho existente atualmente na análise dos resultados (John, 2004). Para que o utilizador tenha uma maior facilidade na perceção do que se pretende testar e de qual é o objetivo do teste, apenas é necessária a descrição da tarefa e a identificação do URL do website a testar. Pretende-se que o utilizador tenha liberdade total de interações para explorar o cenário em causa de forma a concluir o que lhe foi pedido na descrição da tarefa. O utilizador não é influenciado a seguir determinados caminhos nem lhe é exigido um

determinado tipo de interação ou de utilizador. O principal objetivo é que o utilizador efetue ações não planeadas com o objetivo de concluir a tarefa que lhe foi proposta.

Após a conclusão do cenário de interação, é aplicado o KLM não sendo mais necessária a intervenção do utilizador, pois, a *framework* gera automaticamente as estatísticas do caso em análise e as possíveis melhorias e recomendações resultantes da interação do utilizador.

1.3.2 Criar um sistema de indicadores de desempenho para os designers

O principal objetivo da criação de um sistema de indicadores de desempenho centra-se na melhoria da usabilidade e do design das aplicações web. Através dos testes com utilizadores é possível identificar possíveis melhorias e assim permitir a redução do tempo de conclusão das tarefas numa aplicação web, permitindo o aumento da performance de um website. A criação deste sistema de indicadores de desempenho baseia-se na geração de uma *string* KLM composta pelos operadores resultantes da interação do utilizador com o website. Esta *string* é analisada permitindo a identificação de determinados padrões com base na ordem dos operadores e nos respetivos tempos de cada um. Através destes padrões é criado um relatório onde são gerados indicadores de desempenho e apresentados alguns gráficos com dados estatísticos que permitem ao designer implementar melhorias na usabilidade e no design da aplicação.

1.3.3 Desenvolvimento de uma ferramenta para análise de dependências entre recursos web

Com este objetivo pretende-se facilitar a análise da performance das aplicações web, criando uma ferramenta que minimiza o esforço que habitualmente é efetuado na análise das interações cliente-servidor das aplicações web. Esta ferramenta identifica os recursos que podem ser obtidos do servidor com a antecedência necessária de forma a evitar atrasos na execução da aplicação web. Após a identificação destes recursos são calculadas as probabilidades de transferência de cada recurso num total de 20 execuções da ferramenta desenvolvida. São agrupados todos estes recursos em combinações com elevada probabilidade de futura utilização e identificadas as combinações de recursos que estão a causar atrasos no carregamento da aplicação web, de forma a aumentar a performance da mesma e do utilizador na interação com esta.

Uma outra análise é efetuada no sentido de priorizar o carregamento dos recursos. São calculadas as diferenças temporais entre cada um deles de forma a identificar e definir uma ordem para o seu carregamento na aplicação web.

1.4 Contribuições Principais

Este projeto explora a ideia de desenvolver aplicações que sejam resilientes a problemas de desempenho por meio do desenvolvimento de melhores estratégias para amortecer padrões de desempenho com alto impacto na usabilidade.

O facto de os utilizadores serem cada vez mais sensíveis a atrasos nas aplicações web (Hogan, 2014), fez com que a existência de ferramentas como o HTTP2 e as PWAs se tornassem elementos imprescindíveis na melhoria da performance das aplicações web, sendo os principais potenciadores do desenvolvimento da solução que irá ser descrita no capítulo 4.

Para atingir o conjunto de objetivos do projeto, é necessário identificar os parâmetros a analisar ao nível da aplicação. Os parâmetros definem o que vai ser analisado, ou seja, o número de interações cliente-servidor necessárias para terminar uma tarefa na aplicação, o volume de dados transferido em cada interação e o tipo da interação.

1.5 Estrutura do documento

Este documento é composto por mais 4 capítulos, para além da Introdução. Os restantes estão organizados da seguinte maneira.

Capítulo 2 apresenta o histórico de aplicações existentes, identificando aplicações e tendências atuais relacionadas com o KLM e com a performance.

O capítulo 3 começa a apresentar uma visão geral do KLM, da abordagem seguida e dos dois módulos principais que foram desenvolvidos.

O capítulo 4 apresenta a identificação do problema relacionado com a performance das aplicações web e a análise da solução implementada.

Por fim o capítulo 5, Conclusão e trabalho futuro, apresenta as reflexões acerca do trabalho desenvolvido, limitações, bem como o trabalho futuro que irá ser desenvolvido no âmbito do presente projeto.

2. Estado da arte

O modelo KLM tem sido amplamente usado para avaliar o desempenho do tempo da interação utilizador-sistema em diferentes dispositivos. Nesta secção irá ser revisto o KLM e as suas aplicações em contextos relacionados, assim como a análise da performance em diferentes casos de estudo. No artigo elaborado por Al-Megren et al. (Al-Megren, 2018), é apresentada uma pesquisa do KLM e são fornecidas recomendações para futuras aplicações. Já Raimund Schatz (Raimund Schatz, 2014) identifica que é possível aumentar a eficiência da rede ao mesmo tempo que se maximiza a satisfação do utilizador.

Neste capítulo será realizada uma revisão da literatura no contexto de: (1) modelos preditivos de interação humano-computador; (2) aplicações desses modelos preditivos; (3) performance e qualidade de serviço e (4) de aplicações onde é relacionada a performance e a usabilidade.

2.1 Modelo KLM

O modelo GOMS, modelo preditivo abordado no capítulo anterior, a par do KLM, pode ser usado para melhorar a eficiência da interação humano-computador, eliminando interações pouco úteis ou desnecessárias. Pode ser definido como uma descrição de alto nível do conhecimento que um utilizador deve ter para executar tarefas num sistema. Relaciona-se com o processo mental e com a decomposição hierárquica de objetivos pré-definidos. Para além disso, é uma forte referência no contexto da análise do comportamento dos utilizadores, no que diz respeito à previsão da usabilidade efetiva da UI (Card S. K., 1983).

O KLM modela a decomposição de uma tarefa obtida a partir do modelo GOMS, em tarefas de unidade, ao nível do dispositivo (como cliques no rato ou o pressionar de tecla). Permite prever o desempenho do utilizador, medindo o tempo necessário para um utilizador concluir uma tarefa. Proposto por Card et al. (Card S. K., 1980), uma única tarefa engloba duas partes:

aquisição da tarefa e execução da tarefa. O tempo total para completar uma única tarefa é dado pela Equação (1):

$$T_{tarefa} = T_{aquisição} + T_{execução} \quad (1)$$

Ao nível da execução, o KLM fornece operadores físicos e mentais com tempos de resposta predefinidos com determinados valores. Esses operadores são definidos por uma letra e incluem: K pressionar uma tecla, P apontar, H origem, D desenhar, M preparação mental para a ação, e R, resposta do sistema. Estes operadores estão listados na Tabela 1.

Tabela 1- Operadores KLM e previsão dos tempos de execução (Card S. K., 1980)

Tipo	Operador	Descrição	Tempo (s)
		Pressionar uma tecla	
		Ótimo digitador (135 wpm)	0.08
		Bom digitador (90 wpm)	0.12
Físicos	K	Média de um digitador experiente (55 wpm)	0.20
		Média de um digitador não experiente (40 wpm)	0.28
		Escrita de letras aleatórias	0.50
		Escrita de códigos complexos	0.75
		Mau digitador	1.20
		B	Pressionar o botão
	P	Apontar com o rato até a um alvo no ecrã	1.10
	H	Mãos no teclado ou noutro dispositivo	0.40
	D	Desenhar n segmentos em linha reta com um comprimento total de 1	
Mental	M	Preparação mental para a ação	1.35
Sistema	R(t)	Resposta do sistema em segundos	t

Segundo Card et al. (Card S. K., 1980), o operador mais utilizado é o K que representa o ato de pressionar uma tecla ou premir o botão do rato. P representa apontar para um alvo específico com o rato. O seu tempo varia em função da distância para o alvo e do próprio tamanho do alvo, de acordo com a lei de Fitts (Fitts, 1954); para simplificar, é atribuído um valor fixo. H representa a ação de trocar a mão entre o teclado e o rato. D é normalmente associado ao rato e refere-se a arrastar/desenhar. Para executar os operadores físicos, o utilizador tem normalmente de se preparar mentalmente para a execução. Essa preparação é representada pelo operador mental M. Uma vez que este operador não é observável a partir do comportamento do utilizador, é, assim, baseado no conhecimento específico das habilidades do utilizador e na colocação de M's a partir de um conjunto de heurísticas (Card S. K., 1980):

- Regra 0: Inserir M's antes de todos os K's que não fazem parte da *string* de argumento. Colocar M's antes de todos os P's que selecionam comandos.

- Regra 1: Quando o operador M é totalmente antecipado pela intenção, remover o M. Exemplo: Mover rato para carregar num botão: PMK.
- Regra 2: Quando um conjunto de MK's pertence a uma unidade cognitiva, excluir todos os M's exceto o primeiro.
- Regra 3: Quando um K é um terminador redundante no fim da unidade cognitiva, excluir o M antes do K.
- Regra 4: Quando um K segue uma sequência constante, excluir o M antes do K; no entanto, não excluir o M quando o K segue uma sequência variável.

O operador final, R refere-se ao tempo necessário para o sistema responder às ações do utilizador, e normalmente é menosprezado, uma vez que se assume que o sistema responde em tempo real. O início do processo KLM começa com uma codificação (chamada *string* KLM) que contém os operadores físicos (K, P, H e D). Depois disso, são inseridos os M's na *string* de acordo com a regra 0. As regras 1,2,3 e 4 são então aplicadas para cada M de forma a determinar se este deve ser eliminado.

O tempo de execução é a soma dos tempos de cada operador da *string* KLM final e é apresentado na Equação (2):

$$T_{execute} = T_K + T_P + T_H + T_D + T_M + T_R \quad (2)$$

Apesar das conhecidas vantagens do uso do KLM, este modelo apresenta várias limitações. Primeiro, ele considera apenas os utilizadores experientes e não tem em consideração os erros ou as falhas do utilizador. Em segundo lugar, considera tarefas lineares; no entanto, os utilizadores podem realizar várias tarefas ao mesmo tempo ou atingir um objetivo de diferentes formas. Por último, as diferenças individuais não são consideradas, uma vez que o operador mental apenas assume um único valor; a carga de trabalho mental e a fadiga são aspetos dos operadores mentais que não são contabilizados por este modelo.

O KLM foi validado a partir de modelos de observação de interações reais para determinar a precisão da previsão. Segundo Card (Card S. K., 1983), a raiz do erro quadrático médio percentual do modelo é 21%, pelo que esse valor é o melhor que se pode esperar da aplicação do KLM, o que é longe de ser uma boa previsão para um erro.

2.2 Aplicações do Modelo KLM

A definição de usabilidade requer a criação de pontes entre modelos preditivos e empíricos. Durante as últimas décadas, modelos cognitivos preditivos como o KLM têm sido aplicados e estendidos a vários domínios de aplicações, com foco em aplicações móveis e sistemas de deteção de imagens ao vivo (IVIS). Foram desenvolvidas aplicações que fornecem um modelo KLM para estimar o tempo de execução de determinadas tarefas. Algumas das mais representativas são a aplicação desenvolvida no projeto CogTool (John, 2004), que foi

estendida para formulários web através do KLM *Form Analyser* (KLM-FA) (Katsanos C. K., 2013) e do SANLab-CM (Patton, 2010).

A CogTool (John, 2004) é uma ferramenta que pode produzir previsões quantitativas baseadas em modelos de tempo de desempenho qualificado a partir de tarefas demonstradas em maquetes alusivas a *storyboards* da UI (inspirado no projeto DENIM (Lin, 2001)). Cada estado da interface é representado por um *frame* no *storyboard*. Com o *storyboard* criado são definidas as etapas necessárias para completar uma tarefa, e o KLM é usado para determinar o tempo que foi necessário para completar a mesma. O CogTool Explorer (Teo, 2008) baseia-se no CogTool para prever a interação baseada na exploração direcionada a objetivos de um utilizador num web site. No entanto, as ferramentas baseadas nos modelos disponíveis atualmente requerem trabalho manual, que não é assim tão trivial para examinar formulários. Para resolver este problema, o KLM-FA (Katsanos C. K., 2013) estende os recursos das ferramentas existentes atualmente, concentrando-se especificamente na automatização da análise de formulários web. Por um lado, o KLM-FA aumentou a automatização e eficiência de tarefas de avaliação e minimizou o esforço exigido pelo CogTool. No entanto, não foi considerado suporte e avaliação ao design, nem foi contabilizado o cálculo da lei de Fitts, tendo sido usado o seu valor fixo.

O SANLab-CM (Patton, 2010) difere dessas abordagens, pois incorpora a instabilidade como parte do seu paradigma central e oferece um conjunto de ferramentas projetadas para visualizar e comparar caminhos críticos e como eles diferem dentro e entre os modelos.

Essa instabilidade está relacionada com os tempos de execução do operador cognitivo, que podem fazer com que o caminho seguido pelo utilizador para a conclusão das tarefas flutue amplamente.

Foram também desenvolvidos modelos focados na interação humana empírica. Em vez do foco ser no comportamento preditivo, estes modelos concentram-se na perceção cognitiva dos utilizadores que empregam, até certo ponto, técnicas de Inteligência Artificial (IA). Estas ferramentas são documentadas no estudo realizado por Katsanos et. al (Katsanos C. T., 2010) e abrangem a elaboração da estrutura de informação e a avaliação da adequação do link das aplicações web. Alguns dos mais representativos são o *WebCriteria SiteProfile*, *Bloodhound* (Chi, 2003), o método para avaliar a arquitetura dos sites (MESA) (Miller, 2004), a automatização cognitiva passo a passo para a web (ACWW) (Blackmon, 2005), *InfoScent (ISEtool)* (Katsanos C. T., 2006), *Combining Link Scent with Navigation Path Relevancy (CoLiDeS+)* (Van Oostendorp, 2007) e ao nível da conformidade com os padrões, a *framework* de avaliação de usabilidade (*USEFul*) (Dingli, 2011).

WebCriteria SiteProfile emprega agentes de software que funcionam como utilizadores para testar um website e devolver várias métricas de usabilidade de ações simuladas. Esses agentes utilizam um modelo baseado em GOMS para executar ações aleatórias num website. O uso de agentes para extrair métricas de usabilidade foi posteriormente posto em causa por Chi et al. (Chi, 2003), uma vez que, os agentes não consideram o conteúdo da análise e outros aspetos relacionados com o ser humano. Também no estudo efetuado por Chi et al. (Chi, 2003), é apresentado o *InfoScent Bloodhound Simulator*. Neste trabalho é realizada uma avaliação automatizada da usabilidade de aplicações web a partir da utilização de informação e produz

relatórios de usabilidade, para o designer de forma a este identificar problemas de navegação. Este método requer três entradas: as tarefas do utilizador identificadas por palavras chave, o endereço URL de cada página web relacionada com a tarefa e o URL da página web em que a exploração começa.

Em *MESA* (Miller, 2004) é simulada a navegação numa página web modelando a interoperabilidade entre a estrutura do site, a qualidade do link e as limitações cognitivas. Esta aplicação fornece uma previsão do tempo médio de navegação e um rastreamento de todos os links. No entanto, não tem em consideração o layout das páginas web.

A *ACWW* (Blackmon, 2005) fornece uma análise das páginas web de forma a evitar problemas de navegação. Identifica quatro tipos de problemas relacionados com a usabilidade e produz uma previsão média do número total de cliques até um utilizador selecionar o link correto. O designer precisa de fornecer três tipos de entradas: uma descrição do objetivo do utilizador, classificar as regiões das páginas web e as descrições para cada região.

O *ISEtool* (Katsanos C. T., 2006) apresenta uma abordagem semelhante ao *Bloodhound* e é uma ferramenta semiautomática para ajudar na avaliação da descrição da informação das hiperligações em páginas web. Assume que os utilizadores têm um objetivo e que os seus padrões de navegação são guiados pela informação. As entradas fornecidas pelo designer são: uma descrição do objetivo do utilizador, o URL da página web e a análise da semântica latente (LSA) (Landauer, 1997). Fornece um relatório tabular com mapas de calor e links de caminhos seguidos.

O *CoLiDeS+* (Van Oostendorp, 2007) é um modelo cognitivo que oferece suporte à navegação em páginas web descrevendo, passo a passo, um processo em tempo real baseado em correlações positivas entre habilidade espacial, adequação do caminho seguido e o desempenho da tarefa, de forma a permitir que os utilizadores tomem decisões de navegação. Como entrada requer uma descrição detalhada do objetivo do utilizador, cabeçalhos das sub-regiões em cada página web e a identificação dos links em cada sub-região. Os resultados fornecem uma simulação da seleção do link e comportamento de retrocesso.

O *USEFul* (Dingli, 2011) é um protótipo privado baseado na web que avalia a usabilidade de uma aplicação web, utilizando a técnica da avaliação heurística, referente ao conjunto de 242 diretrizes de usabilidade baseadas em pesquisas compiladas pelo autor. No entanto, foram apenas consideradas as diretrizes mais simples de implementar. Os resultados foram comparados com o trabalho de Nielsen et al. (Nielsen J. T., 2002), e a sua *framework* detetou, em média, mais 28,5% de transgressões do que o referido trabalho.

Não obstante, as ferramentas acima mencionadas tenham contribuído para a automatização de tarefas, apresentam também várias desvantagens. Tanto o *Bloodhound* como o *ISEtool* não modelam a pesquisa visual e assumem uma avaliação global de todas as hiperligações numa página web. Enquanto que o *ACWW* e o *CoLiDeS+* solicitam ao designer uma análise manual da página web em sub-regiões, não especificando quais as estratégias no caso de navegações web mais complexas. Por outro lado, as ferramentas preditivas simulam a perceção visual e fornecem uma maior perceção do comportamento esperado e previsões precisas do desempenho do utilizador.

O modelo KLM tem sido explorado em diversas áreas de conhecimento. No âmbito do *Reinforcement Learning* (RL) (Katri Leino, 2019) ou para aplicações móveis, para prever o tempo médio que os utilizadores demoram a concluir uma tarefa (Paul Holleis, 2007).

Ao nível do RL, no KLM, as políticas de tarefas são criadas manualmente, sendo que uma variação substancial no comportamento do utilizador pode tornar esse trabalho bastante dispendioso, sendo necessária uma alteração das políticas de tarefas. Devido a isto, surge o contexto do RL (Katri Leino, 2019), onde o KLM é utilizado para comparar a política de tarefas por parte de um utilizador, definindo um cenário específico com determinadas ações que o utilizador deve fazer. Através deste, o RL atua como um método para aprender um modelo a nível de tarefas, não se restringindo a ambientes totalmente observáveis, pois o agente aprende com as interações.

O RL, sendo um método geral para aprender políticas de tarefas em ambientes que exigem ações sequenciais e onde o ambiente não é totalmente conhecido, um agente RL tenta descobrir o comportamento ideal por tentativa e erro, interagindo com o ambiente. Em suma, ao utilizar o KLM com o RL, pretende-se usar os custos de tempo do tipo de operador do KLM utilizado pelo utilizador, como forma de recompensar a aprendizagem do agente.

Uma outra aplicação do KLM centra-se no tempo que os utilizadores tendem a demorar para concluir uma tarefa num dispositivo móvel (Paul Holleis, 2007). Este estudo está relacionado com o facto de os telemóveis terem menor capacidade de processamento, de memória e de bateria, quando comparados com os computadores tradicionais. Assim, este tipo de ambiente força o estabelecimento de um compromisso entre usabilidade e utilização de recursos.

Segundo Alix Goguey (Alix Goguey, 2018), existem poucas ferramentas que possam ajudar os designers a prototipar rapidamente UI baseadas em toques com as mãos (AMA, 2019). De forma a ser possível combinar os benefícios das ferramentas de prototipagem rápida, com os modelos preditivos, os autores desenvolveram uma ferramenta que explora e prevê o desempenho do utilizador em interfaces em que é necessário o toque. Essa ferramenta foi avaliada num cenário experimental, onde foram comparados os seus resultados com valores temporais empíricos, para vários cenários. Através deste estudo, os autores concluíram que a ferramenta desenvolvida indicou previsões precisas, conseguindo explorar e prever o desempenho do toque do utilizador, mesmo em UIs que se encontravam numa fase inicial de desenvolvimento.

Relativamente ao modelo GOMS, Rebai Rim (Rebai Rim, 2017) com o estudo acerca das UI adaptáveis utilizou este modelo para avaliar a usabilidade dessas interfaces permitindo determinar como os utilizadores podem aprender a atingir os seus objetivos, no que toca à conclusão de um cenário experimental. Ao adaptar as interfaces consoante a aprendizagem do utilizador, recorrendo também ao modelo GOMS, chegou à conclusão de que as UI adaptáveis, permitem um maior conforto e um maior desempenho do que as UI fixas.

Himali Saitwall (Himali Saitwall, 2010), com o objetivo de avaliar a usabilidade de um registo eletrónico de saúde e sugerir áreas de melhoria na interface do utilizador, utilizou os modelos preditivos GOMS e KLM. O modelo GOMS, para avaliar a interface do utilizador da aplicação, classificando cada etapa de uma determinada tarefa, enquanto que o modelo KLM foi utilizado

para prever o tempo necessário para executar cada tarefa. Após a análise de 14 tarefas realizadas por utilizadores, chegou à conclusão que um utilizador necessita de, em média, 106 etapas para concluir uma tarefa específica. E que para executar as 14 tarefas propostas, eles gastaram 22 minutos (tempo este independente do tempo do sistema) apenas para a entrada de dados no sistema, sendo que, desses 22 minutos, 11 foram gastos em operadores mentais.

De acordo com Chi et al. (Chi, 2003) existem dois tipos de ferramentas que avaliam automaticamente a usabilidade: as que fazem uso da conformidade com padrões e as que tentam prever o uso de um website. A *framework* proposta neste projeto enquadra-se em ambas as categorias, uma vez que aplica heurísticas de usabilidade como recomendações para os designers, com base numa análise KLM preditiva que fornece uma *string* KLM que é analisada, extraíndo as métricas preditivas. Esta *framework* será apresentada no capítulo 5.

2.3 Performance e QoS

Uma outra perspetiva associada a este projeto é a performance de aplicações, em particular, as aplicações web. No contexto deste tema, Jean Bolot (Jean-Chrysostome Bolot, 1996), com o estudo acerca da performance, conseguiu provar que o tempo de acesso aos dados localizados num determinado servidor, depende de vários parâmetros, entre eles, a carga no servidor e a carga na rede. Essas sobrecargas são evitadas ao dimensionar corretamente o servidor, efetuando balanceamento de carga e usando mecanismos de *caching* para minimizar os pedidos cliente-servidor.

Com a existência de todos estes fatores, como a carga no servidor ou na rede, a performance tem um impacto bastante significativo na usabilidade de uma aplicação. Estas restrições fazem com que o designer de interfaces tenha de adaptar a UI de forma a precaver-se de possíveis atrasos na comunicação com o servidor, utilizando *widgets*, como por exemplo, barras de carregamento, com a típica palavra ‘*Loading*’. Este tempo de espera de carregamento de uma interface ou componente, irá afetar o KLM, pois o tempo previsto para um utilizador concluir uma ação irá ser previsivelmente maior. Toda esta situação influencia o KLM, fazendo com que os resultados finais não tenham o mesmo nível de confiabilidade que era suposto terem.

Os dados necessários para a análise da performance das interações entre o cliente e o servidor são muitas vezes fornecidos em formatos normalizados como o *HTTP Archive* (HAR). Estes ficheiros armazenam os dados em formato JSON com a informação de cada recurso que é descarregado pelo browser, assim como informações temporais relacionadas com o carregamento desses recursos (Odvarko, s.d.). Através destes ficheiros, é possível verificar o tempo de processamento e de comunicação de cada browser com o servidor.

Tomando novamente como exemplo o estudo efetuado por Jean Bolot (Jean-Chrysostome Bolot, 1996) é possível concluir que a qualidade dos serviços web depende do tempo que é necessário para recuperar todos os dados, pois, esse tempo está diretamente relacionado com a carga que possa existir na rede ou no servidor.

Uma análise mais recente realizada por Raimund Schatz (Raimund Schatz, 2014), identifica precisamente que é possível maximizar a eficiência da rede e a economia de recursos, maximizando ao mesmo tempo a satisfação do utilizador e, consecutivamente, a sua performance através da relação entre a gestão da rede baseada na qualidade de utilização por parte do utilizador, fazendo uso da memória cache, e funções padronizadas de manutenção e administração de redes.

Relativamente às PWAs, é possível, através do software da Google chamado *Lighthouse* (Audit web apps with Lighthouse., 2019), perceber quais são os componentes responsáveis pelo atraso no carregamento da página e definir uma ordem ou sequência para o seu carregamento. No estudo relacionado com PWAs, Andreas B. Hansen (Andreas B. Hansen, 2017) conseguiu identificar duas diferenças bastantes significativas entre uma PWA, uma aplicação web tradicional baseada em *React Native* e uma aplicação híbrida. A tabela 1 mostra exatamente essas diferenças.

Tabela 2- Diferenças de performance entre 3 tipos de aplicações

Casos de estudo	Aplicação híbrida	Aplicação web baseada em React	PWA
Tamanho do pacote de instalação	4.53MB	16.39MB	104KB
Tempo de carregamento	860ms	246ms	230ms

Com o estudo apresentado na tabela anterior, a diferença mais significativa é, sem dúvida, a maior performance que as PWA fornecem em comparação com as restantes aplicações.

2.4 Performance e Usabilidade

A capacidade de resposta da interação efetuada pelos utilizadores representa um dos principais motivos pelos quais os utilizadores finais instalam aplicações em dispositivos móveis em vez de usarem a página web equivalente. No entanto, o desempenho da interação com o utilizador ganhou um elevado estatuto com o recente aumento da popularidade das *Progressive Web Applications* (PWA). Esta nova classe de aplicações web procura melhorar os níveis de desempenho das aplicações móveis sem o custo de: (1) desenvolver aplicações dependentes da plataforma (por exemplo, web, iOS e Android); (2) instalar aplicações de uma loja; (3) desperdício de quantidades significativas de espaço de armazenamento com aplicações que raramente são usadas.

A análise do desempenho de interação em aplicações web cobre dois períodos do ciclo de vida de execução da aplicação: (1) carregamento da aplicação; e (2) interação do utilizador. As métricas de desempenho que cobrem a fase de carregamento da aplicação permitem medir quanto tempo a aplicação leva para ser útil aos utilizadores. Esta classe de métricas inclui:

- *First Contentful Paint* (FCP) - tempo necessário para processar o primeiro elemento da página web, após ser carregado;
- *Largest Contentful Paint* (LCP) - tempo necessário para renderizar o maior objeto visível na janela de visualização de uma página web;
- *Time to Interactive* (TTI) - o tempo que uma página web demora a carregar antes que o utilizador possa interagir com ela.
- *First Input Delay* (FID) - o momento em que um utilizador interage pela primeira vez com uma página até o momento em que o navegador responde a essa interação.

Ferramentas como o *Lighthouse* (Audit web apps with Lighthouse., 2019), que já foi falado anteriormente, usam métricas para analisar o desempenho das páginas web do ponto de vista da prontidão como estas podem ser usadas pelos utilizadores finais.

Uma das formas de controlar os recursos armazenados em cache é a utilização dos *service workers*. Os *service workers* são *scripts* que os navegadores executam em segundo plano de forma a manipular recursos que não precisam da interação do utilizador ou do carregamento de uma página web para serem descarregados, permitindo a interceção e manipulação de interações cliente-servidor, assim como, a gestão da memória cache (Gaunt, 2020).

Uma outra ferramenta que permite o aumento da performance das aplicações web é o protocolo HTTP2 que trouxe melhorias significativas ao nível da eficiência, velocidade e segurança na comunicação cliente-servidor, sendo suportado pela maioria dos navegadores atuais. Este protocolo permite a priorização de recursos e o carregamento destes em simultâneo. Maarten Wijnants (Maarten Wijnants, 2018) constatou que a priorização de recursos web e a ordem do seu carregamento permitem o aumento na performance de uma aplicação web, sendo essas melhorias mais evidentes nas páginas web em que é exigido um maior processamento de dados, sendo por isso páginas mais pesadas. Através desta ferramenta e dos *service workers* é possível o aumento da performance das aplicações web devido, essencialmente, à utilização da memória cache e à definição de uma ordem para o carregamento de recursos web, focados na prioridade de cada um.

No contexto da relação entre performance e usabilidade nas aplicações, Rachel Bellamy (Rachel Bellamy, 2011), com o estudo realizado para aplicações desktop, verificou que o modelo KLM permitiu concluir que um determinado cenário era mais lento do que outro, mas que o que realmente interferia com o desempenho da aplicação era o tempo de resposta do servidor. Logo, o foco dos programadores deveria estar na melhoria dos tempos de resposta do sistema e não no redesenhar da aplicação criando um novo cenário para o utilizador.

Já Amit Mishra (Amit Mishra, 2017), através do estudo da usabilidade de uma aplicação web, conseguiu medir o nível de usabilidade do site de uma universidade utilizando fatores como a organização e legibilidade da informação, a navegação, o design da UI e o desempenho e eficácia da aplicação web, identificando os pontos fortes e fracos do site e se este conseguiu atender às necessidades dos utilizadores. Posto isto, foi possível concluir que a usabilidade permitiu identificar interfaces do site onde os níveis de performance eram mais baixos, quando

comparados com outras UIs do site e que devido a isso a usabilidade é um dos fatores que, sem dúvida, tem de ser levado em conta no desenvolvimento de uma aplicação web.

Com o estudo relacionado com testes de regressão de desempenho humano, Amanda Swearngin (Amanda Swearngin, 2013) constatou que apesar das ferramentas de modelação cognitiva, como o CogTool, fornecerem alguma ajuda na previsão de desempenho dos utilizadores, estas ainda exigem algum esforço manual na descrição da interface e das tarefas do utilizador. Devido a este facto, os autores desenvolveram uma ferramenta que permite reduzir o esforço necessário na geração de modelos de desempenho humano dos sistemas já existentes. Com este estudo, concluíram que amostras que ocupam cerca de 50% do tempo de execução perdem no máximo 10% de desempenho previsto.

2.5 Conclusão

Um dos principais desafios que se prendem com o desenvolvimento de uma aplicação web é a realização dos testes com utilizadores, pois em muitas situações, estes apenas são realizados quando a aplicação já está concluída ou quando esta se encontra num estado de desenvolvimento bastante adiantado. A ferramenta proposta foca exatamente nas limitações do utilizador quando este interage com uma aplicação web, tornando as interfaces mais adequadas e adaptáveis para ele. Identifica causas que originem atrasos no carregamento de páginas web, que tenham um enorme impacto tanto na usabilidade como na performance da mesma.

Tendo em conta as aplicações existentes atualmente, e na literatura realizada, não foram encontradas ferramentas que pudessem orientar o designer de uma forma tão simples e intuitiva, com um modelo de entrada tão acessível a qualquer tipo de tarefa que seja necessária efetuar de forma a orientar o designer para as possíveis falhas que a sua aplicação possa ter. Não foram identificadas aplicações que conseguissem transparecer tão facilmente para o designer, as fragilidades da sua aplicação web e como este a poderia tornar mais direcionada para os utilizadores. A ferramenta proposta neste documento permite exatamente isso, a sugestão de indicadores de desempenho que visam a melhoria dos websites, tanto ao nível da usabilidade como ao nível da performance.

Tal como referido, a *framework* desenvolvida neste projeto apresenta uma abordagem diferente das já existentes, pelo que facilitará o trabalho do designer e tornará mais simples a implementação de melhorias nas aplicações web em causa.

3. Interação não planeada usando KLM

Uma ação não planeada corresponde a uma ação do utilizador em que este tem um objetivo a atingir na interface, mas a sequência de ações é definida pelo próprio, não tendo que seguir um caminho predefinido. Neste capítulo será descrito o processo que permite converter uma ação deste tipo em indicadores de desempenho para o designer. Será feita a descrição de alto nível da arquitetura da aplicação a implementar e, de seguida, serão detalhados os processos associados a cada componente dessa arquitetura.

3.1 Descrição de alto nível da *framework* não planeada

Antes de ser proposta a solução que visa obter indicadores de desempenho para melhorar a usabilidade e o design das aplicações web, irá ser descrito o modelo de alto nível que se encontra por trás da *framework* desenvolvida.

Neste modelo, o primeiro pressuposto está relacionado com o designer de interfaces que idealiza a aplicação, a desenha e a disponibiliza para testes. De seguida, o utilizador executa interações não planeadas na aplicação (associadas a um determinado objetivo). A execução de ações por parte do utilizador permite a extração de métricas acerca da interação realizada. Através destas métricas são obtidos indicadores de desempenho da tarefa realizada pelo utilizador que permitem ao designer a implementação de melhorias no design da aplicação. É expectável que os indicadores de desempenho possam ser diferentes para utilizadores distintos, uma vez que poderão seguir caminhos potencialmente diferentes para concluir a mesma tarefa. Isto permite obter diferentes indicadores em função das diferentes interações. Para se obterem os indicadores de desempenho, o modelo é composto por dois componentes principais: o módulo de implementação e o módulo de análise, conforme apresentado na Figura 1 - Visão geral da *framework* com ações não planeadas por parte do utilizador.

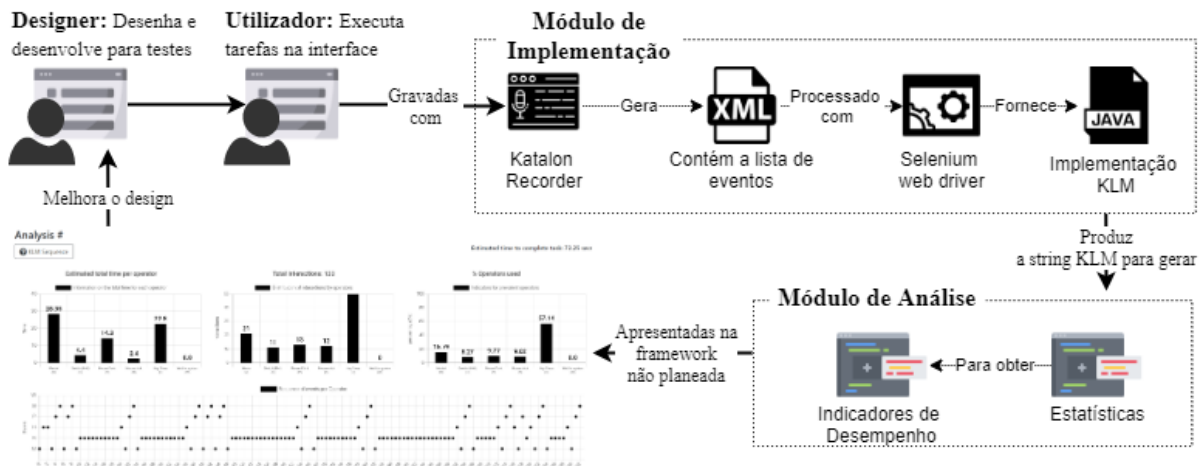


Figura 1 - Visão geral da *framework* com ações não planeadas por parte do utilizador

O módulo de implementação captura eventos de interação do utilizador ao executar uma tarefa numa aplicação web usando o *Katalon* (Katalon, 2020). Esta aplicação armazena todas as interações num ficheiro XML que é processado usando a linguagem de programação JAVA e a API do *Selenium WebDriver* (Selenium., 2020), que atuam sobre o ficheiro XML para extração de informação. De seguida, o KLM (Card S. K., 1980) é aplicado e é gerada a correspondente *string* KLM, configurando a fase de pré-processamento da *framework*. Esta *string* KLM contém todas as interações que o utilizador executou na interface mapeadas numa sequência de operadores.

O módulo de análise atua na *string* KLM para extrair estatísticas dos operadores usados na interação. Tal como referido anteriormente, uma vez, que a interação do utilizador não é planeada, os utilizadores podem atingir um objetivo de interação seguindo caminhos possivelmente diferentes. Portanto, numa primeira fase, estas estatísticas permitem determinar o tipo de interação do utilizador (baseado no teclado ou no rato), e de seguida, são fornecidos indicadores de desempenho ao designer de forma a que este consiga melhorar o design da aplicação web. Nas seguintes secções, são apresentados detalhes da implementação de cada módulo.

3.2 Extração da *string* KLM

A extração da *string* KLM corresponde ao *output* do módulo de implementação, que será descrito nesta secção. De forma a atingir esse objetivo, são realizadas as seguintes etapas: armazenar a interação efetuada pelo utilizador, processar o ficheiro XML e, no final, gerar a *string* KLM. Na primeira etapa, a interação que o utilizador executa na aplicação web é processada através da ferramenta *Katalon Recorder* (Katalon, 2020) que é uma extensão do navegador que regista ações e captura os elementos da aplicação web, guardando-os num ficheiro XML. A estrutura deste ficheiro XML é composta pelo seguinte conjunto de *tags* que

estão associadas a um evento (identificadas como `<selenese>`): o propósito da ação, `<command>`, contém uma chave ou caminho que permite identificar o elemento web presente na interação, `<target>`, que é obtido através da tag `<value>`. A título de exemplo, considere-se o XML apresentado na Figura 2 que se refere à interação com uma caixa de texto de email. O primeiro evento está associado ao clique na caixa de texto para ativar o elemento, e o segundo relaciona-se com a escrita de texto nesse mesmo elemento.

```
<selenese>
  <command>click</command>
  <target><![CDATA[name=registrationEmail]]></target>
  <value><![CDATA[]]></value>
</selenese>
<selenese>
  <command>type</command>
  <target><![CDATA[name=registrationEmail]]></target>
  <value><![CDATA[dfghjkljsdf@gmail.com]]></value>
</selenese>
```

Figura 2 - Tags XML da caixa de texto email: o utilizador clica na caixa de texto e insere o email

O primeiro evento, o `<command>` retorna o valor `click` que corresponde ao clique do utilizador no componente antes de escrever o email. No segundo evento, o valor do `<command>` é `type` que se refere ao uso do teclado no elemento web indicado no `<target>`.

Assim que o ficheiro XML é gerado através do *Katalon Recorder*, a segunda etapa é iniciada. Nesta etapa, o XML é interpretado na implementação JAVA e são extraídas as *tags* para o processamento do KLM. Este processo é realizado com a ajuda da API do *Selenium Web Driver* (Selenium., 2020). Esta API, fornece funções adequadas para extrair informações do ficheiro XML, associar essa informação aos operadores do KLM (Secção 2.1) e construir a respetiva *string* KLM. Este processo é ilustrado na Figura 3 e é apresentado um exemplo da aplicação.

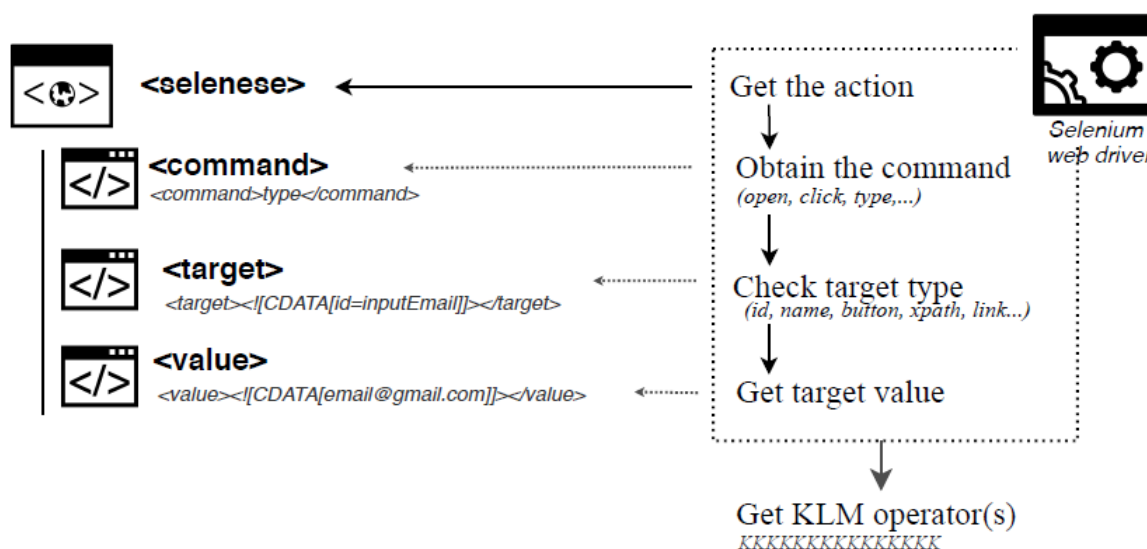


Figura 3 - Conversão do XML: do ficheiro XML para a *string* KLM

Cada evento `<selenese>` hierárquico é processado para obter o `<value>` associado a uma ação específica. Logo, quando um utilizador executa um `<command>` `type` num `<target>` do tipo `Id`, então é interpretado como o utilizador estar a interagir com uma caixa de texto. De seguida, o `<value>` `email@gmail.com` é obtido e guardado. Este `<value>` é do tipo texto e é analisado no KLM como sendo uma sequência de 15 operadores K (valor do KLM para a ação de pressionar uma tecla). Esta sequência é adicionada à *string* KLM e o próximo evento `<selenese>` do ficheiro XML irá ser analisado para posteriormente ser também inserido na *string* KLM. No caso do próximo evento `<selenese>` executar um `<command>` `click` num `<target>` do tipo `Id`, significa que o utilizador interagiu com um elemento web clicando sobre ele. Como um `click` não tem texto associado, o elemento `CDATA` associado à tag `<value>` não terá valor, contrariamente ao caso associado à escrita exemplificado na Figura 3. A *framework* consegue detetar quais os operadores KLM que estão associados ao valor presente no `<command>`, adicionando esses operadores à *string* KLM. Neste caso iria adicionar os operadores HPB associados, respetivamente, ao movimento da mão do utilizador do teclado até ao rato, apontar o rato até ao elemento web e o pressionar o botão do rato.

Quando todos os eventos forem processados, é criada a *string* KLM inicial. A *string* KLM final é gerada aplicando as regras de inserção do operador mental, M, descritas na Secção 2.1.

Para validar a precisão da implementação do KLM, foram comparadas as previsões não planeadas com os dados de teste do utilizador para o formulário de registo da *Cable News Network* (CNN). Foram consideradas duas estratégias de interação: (a) baseada no rato, no qual um utilizador interage com o formulário com o rato, exceto para as entradas de texto, e (b) baseada no teclado, em que os campos do formulário são alcançados usando a tecla `tab`.

Foi solicitado a dez utilizadores que, como tarefa, realizassem as etapas necessárias para o registo na referida aplicação. Não foi definida nenhuma sequência de passos, exceto para o objetivo da interação. Desta forma, foram utilizadas as seguintes suposições na *framework*: (a)

o tempo de resposta do sistema, W é negligenciado (isto é, $W=0$), (b) a mão do utilizador começou no teclado antes da execução, (c) foram habilitados os cálculos da lei de Fitts.

A Tabela 3 apresenta os tempos de execução da tarefa efetuada pelo utilizador e os tempos previstos da *framework* não planeada, junto com a taxa de erro e o coeficiente de covariação do RMSE (CV-RMSE). Os resultados mostram que o erro médio das previsões foi de 8,43%, que se encontra dentro da margem de erro de 20% relata por Ivory and Hearst (Ivory, 2001).

Tabela 3- Resultados não planeados mostrando médias e, entre parêntesis, desvios padrão

Interação*	Tempo tarefa participante (ms)	Tempo previsto Framework (ms)	Taxas de erro previsões (%)	CV-RMSE previsões (%)
Baseada no teclado	24743 (1902.48)	25020	1.12%	7.68%
Baseada no rato	25679 (3522.02)	29720	15.74%	18.04%

*formulário registo da CNN, identificado como [S01] na Tabela 7 Apêndice A

No geral, a *framework* não planeada, é uma implementação precisa do KLM que sobrestima o tempo da tarefa na interação baseada no rato e subestima a interação baseada no teclado.

3.3 Estatísticas e Detecção do tipo de interação

As recomendações de design são de extrema importância para os designers na criação de uma aplicação web. Estas, resultam de interações do utilizador, nas quais um utilizador tem um objetivo a atingir numa aplicação web e executa uma sequência não planeada de interações. Assim, cada utilizador executa uma tarefa livremente, de forma a conseguir atingir um objetivo. Ao fornecer esse grau de liberdade ao utilizador, torna-se importante que os designers entendam o resultado do seu comportamento. Este projeto foca-se em três tipos de estratégias de interação: (a) o utilizador, com base nas interações do rato, navega na página até que o objetivo seja alcançado; (b) o utilizador usando o teclado; ou (c) combinação de ambos.

O módulo de análise consiste na extração de indicadores de desempenho da *string* KLM para gerar recomendações de design que melhorem a usabilidade da aplicação web. Neste projeto, não se obtêm mais informações sobre a aplicação web HTML do que as que são extraídas para gerar a *string* KLM. É apresentado um algoritmo baseado em regras que utiliza os valores dos operadores na *string* KLM para realizar cálculos que fornecem indicadores do desempenho da interação. Esses indicadores são usados no algoritmo baseado em regras para fornecer recomendações de design. Ao nível dos indicadores de desempenho, é difícil para os designers identificarem, a partir de uma *string* KLM, potenciais problemas numa interação do utilizador. Portanto, a *string* KLM é analisada em três representações visuais: o número de operadores usados na interação, o tempo por operador e a linha do tempo dos eventos por operador. Para isso ser alcançado, a *string* KLM é convertida numa matriz e são realizadas operações de

3 – Interação não planeada usando KLM

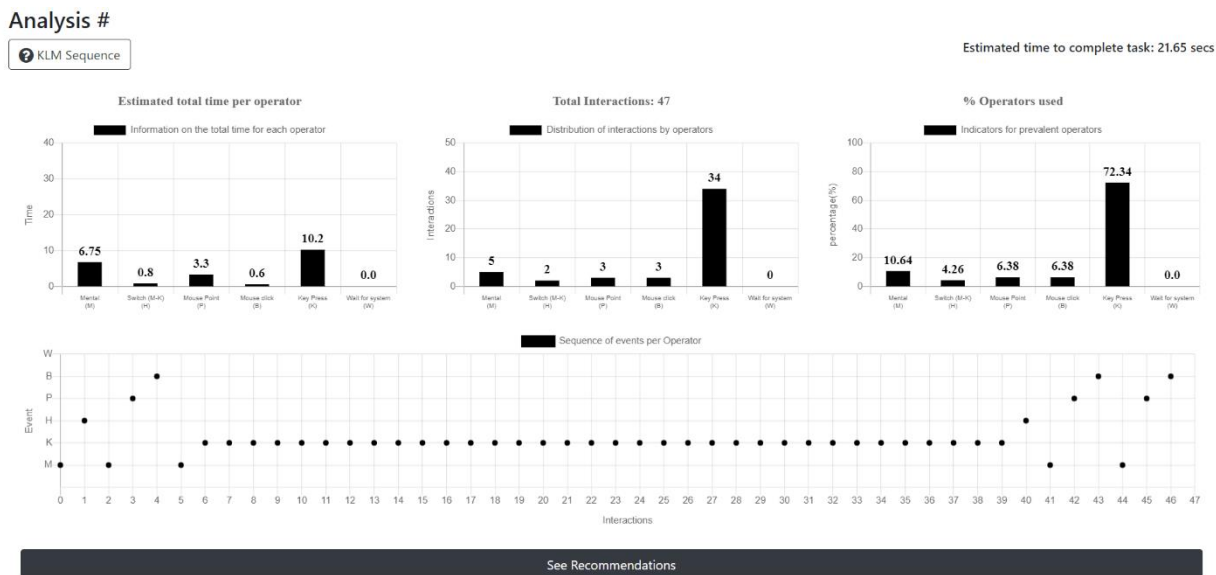


Figura 5 - Transformação *string* KLM em estatísticas com uma interação baseada no teclado

Como a Figura 4 em comparação com a Figura 5 diferem no tipo de interação efetuada, já não se verifica a interrupção na sequência de K's no gráfico de pontos. Este facto deve-se à utilização da tecla TAB na passagem entre caixas de texto, algo que não se verificou na interação baseada no rato e teclado, pois esta passagem foi efetuada com o clique em cada caixa de texto.

Para implementar um método baseado em regras é necessário, primeiramente, determinar os padrões da estratégia de interação do utilizador. A Tabela 4 e a Figura 6 apresentam os resultados obtidos aplicando as estratégias de interação a catorze aplicações web (identificadas na Tabela 7 do Apêndice A).

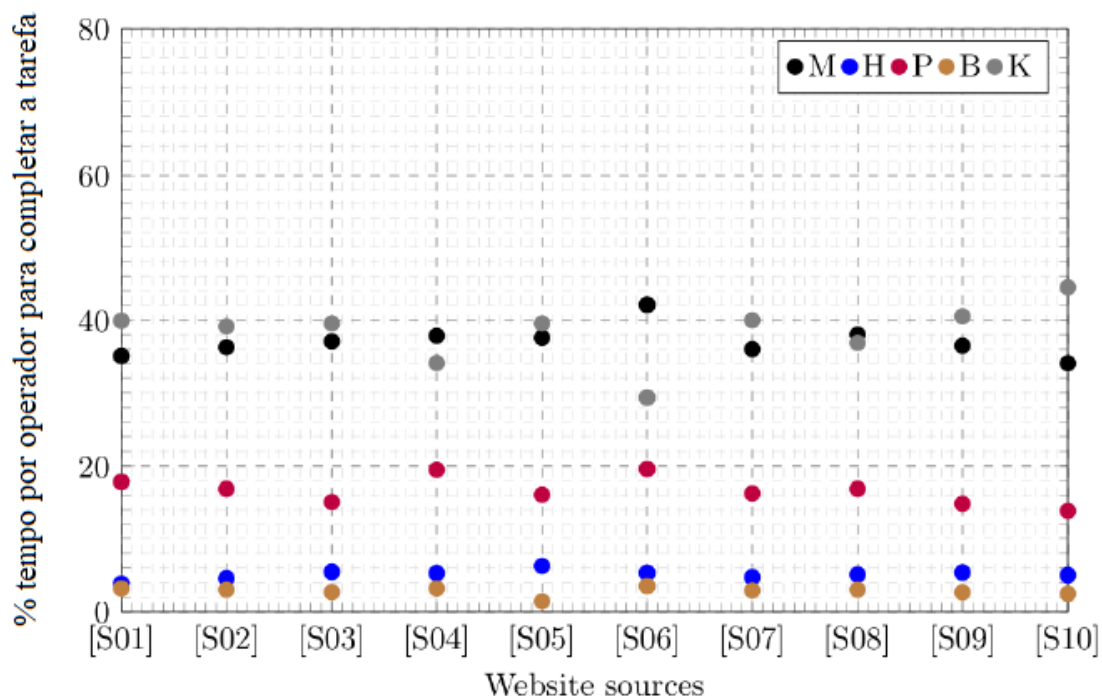
Tabela 4 - Estratégias de interação seguidas em dez fontes, e percentagens correspondentes de tempo gasto e número de operadores usados

Fontes*	% de tempo por operador (% de operadores usados)					
	M	H	P	B	K	
Registo (Rato e Teclado)	[S01]	35.06 (12.90)	3.89 (4.84)	17.85 (8.06)	3.24 (8.06)	39.93 (66.14)
	[S02]	36.27 (13.46)	4.60 (5.77)	16.89 (7.69)	3.07 (7.69)	39.15 (65.39)
	[S03]	37.08 (13.70)	5.49 (6.85)	15.10 (6.85)	2.74 (6.85)	39.56 (65.75)
	[S04]	37.85 (14.84)	5.31 (7.03)	19.48 (9.38)	3.24 (8.59)	34.09 (60.16)
	[S05]	37.58 (13.77)	5.27 (6.52)	16.11 (7.24)	1.46 (7.25)	39.56 (65.11)
	[S06]	42.09 (17.50)	5.34 (7.50)	19.59 (10.00)	3.56 (10.00)	29.35 (55.00)
	[S07]	36.00 (13.24)	4.74 (5.88)	16.29 (7.35)	2.96 (7.35)	40.00 (66.18)
	[S08]	38.02 (14.47)	5.12 (6.58)	16.90 (7.79)	3.07 (7.89)	36.87 (63.17)
	[S09]	36.48 (13.32)	5.40 (6.67)	14.86 (6.67)	2.70 (6.67)	40.54 (66.67)
	[S10]	34.06 (11.94)	5.04 (5.97)	13.88 (5.97)	2.52 (5.97)	44.47 (70.15)
	Média	37.05 (13.92)	5.02 (6.36)	16.70 (7.71)	2.86 (7.63)	38.36 (64.38)
Registo (teclado - Tab)	[S01]	29.67 (10.00)	2.93 (3.33)	16.11 (6.67)	2.93 (6.67)	48.35 (73.33)
	[S02]	31.17 (10.64)	3.69 (4.26)	15.24 (6.38)	2.77 (6.38)	47.11 (72.34)
	[S03]	22.13 (6.45)	3.27 (3.22)	9.01 (3.23)	1.63 (3.23)	63.93 (83.87)
	[S04]	29.78 (10.28)	4.01 (4.67)	17.65 (7.48)	2.80 (6.54)	45.73 (71.03)
	[S05]	23.19 (6.93)	2.94 (2.97)	10.79 (3.96)	1.96 (3.96)	61.10 (82.18)
	[S06]	34.06 (12.31)	3.78 (4.62)	17.35 (7.69)	3.15 (7.69)	41.64 (67.69)
	[S07]	31.87 (10.94)	4.04 (4.68)	14.83 (6.25)	2.69 (6.25)	46.54 (71.88)
	[S08]	32.19 (11.11)	4.08 (4.76)	14.99 (6.35)	2.72 (6.35)	45.99 (71.43)
	[S09]	29.34 (9.52)	4.34 (4.76)	11.95 (4.76)	2.17 (4.76)	52.17 (76.20)
	[S10]	26.73 (8.47)	3.16 (3.39)	13.06 (5.08)	2.37 (5.08)	54.65 (77.98)
	Média	29.01 (9.67)	3.62 (4.01)	14.10 (5.79)	2.52 (5.70)	50.72 (74.80)
Navegação (Rato)	[S011]	53.54 (35.00)	2.26 (5.00)	37.39 (30.00)	6.79 (30.00)	0.0 (0.0)
	[S06]	54.65 (35.71)	3.23 (7.14)	35.62 (28.57)	6.47 (28.57)	0.0 (0.0)
	[S12]	53.54 (35.00)	2.26 (5.00)	37.39 (30.00)	6.79 (30.00)	0.0 (0.0)
	[S13]	54.65 (35.71)	3.23 (7.14)	35.62 (28.57)	6.47 (28.57)	0.0 (0.0)
	[S03]	54.65 (35.71)	3.23 (7.14)	35.62 (28.57)	6.47 (28.57)	0.0 (0.0)
	[S07]	54.00 (35.29)	2.66 (5.88)	36.66 (29.41)	6.66 (29.41)	0.0 (0.0)
	[S14]	54.65 (35.71)	3.23 (7.14)	35.62 (28.57)	6.47 (28.57)	0.0 (0.0)
	[S01]	55.67 (36.36)	4.12 (9.09)	34.02 (27.27)	6.18 (27.27)	0.0 (0.0)
	[S02]	52.56 (34.38)	1.41 (3.13)	38.93 (31.25)	7.07 (31.25)	0.0 (0.0)
	[S10]	57.44 (37.50)	5.67 (12.5)	31.20 (25.00)	5.67 (25.00)	0.0 (0.0)
	Média	57.44 (35.65)	3.13 (6.92)	35.81 (28.72)	6.51 (28.72)	0.0 (0.0)

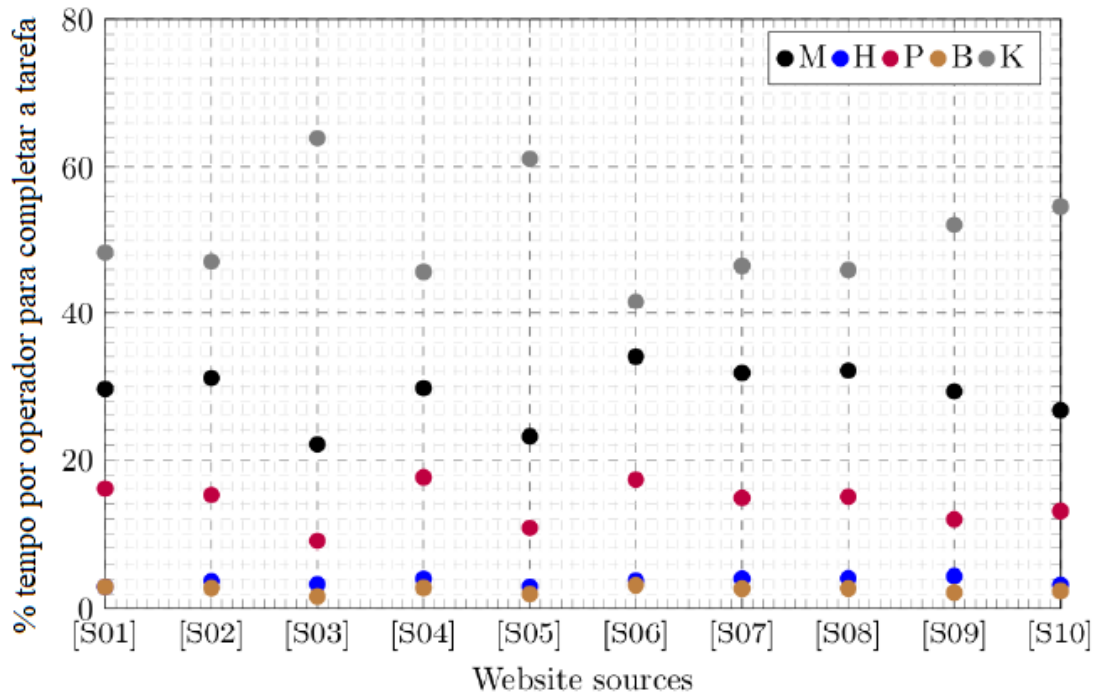
*Lista de fontes disponível na Tabela 7 do Apêndice 1

Os resultados mostram que a porcentagem de tempo utilizada por cada operador varia dependendo das estratégias de interação. Na estratégia de interação baseada no rato, os operadores predominantes são o M e o P com 57.44% e 35.81%, em média, considerando o tempo total despendido na interação, respetivamente. Na interação baseada no teclado, são os operadores M (29.01%) e K (50.72%), e na interação baseada na combinação de rato e teclado, os operadores predominantes são o M (37.05%) e K (38.36%). No entanto, podem ser encontradas semelhanças na estratégia baseada no teclado e na estratégia baseada no rato e teclado, pois o aumento de M é compensado por uma diminuição em K, complementado com um aumento em H e P.

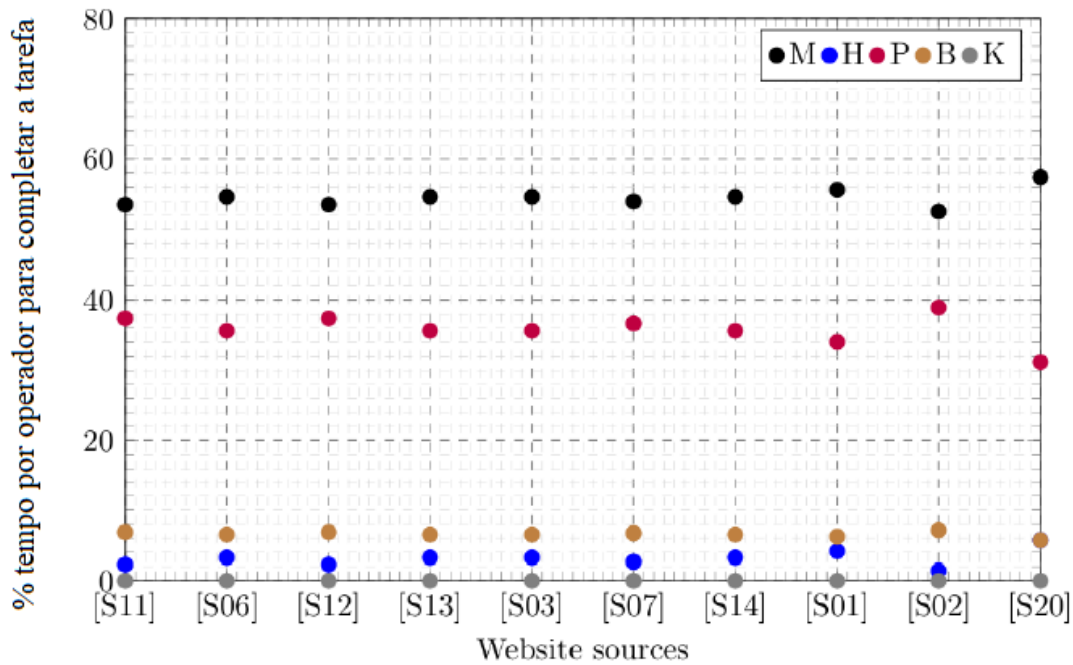
Para efeito de comparação, considerou-se também o número de operadores utilizados (materializado na porcentagem de operadores). Os resultados mostram que não é fornecido um bom indicador no que toca ao tempo que o utilizador despende, uma vez, que uma elevada porcentagem de operadores não implica um aumento de tempo (primeiro cenário, onde o K é usado em 64.38% dos operadores e, em tempo despendido, corresponde quase ao mesmo tempo gasto que o operador M, com apenas 13,92% dos operadores utilizados). Considerando os padrões, a Figura 6 representa a porcentagem de tempo usada por cada operador em cada estratégia de interação, e o desvio padrão, σ , de cada operador dentro do seu contexto.



(a) Interação Rato-Teclado ($\sigma_M=2.05$, $\sigma_H=0.46$, $\sigma_P=1.79$, $\sigma_B=0.55$, $\sigma_K=3.89$)



(b) Interação com teclado ($\sigma_M=3.69$, $\sigma_H=0.49$, $\sigma_P=2.69$, $\sigma_B=0.45$, $\sigma_K=6.82$)



(c) Interação com rato ($\sigma_M=1.26$, $\sigma_H=1.1$, $\sigma_P=2.0$, $\sigma_B=0.36$, $\sigma_K=0.0$)

Figura 6 - Formulários de registo usados para testes em diferentes cenários de interação

De maneira a prever o intervalo de valores, admissíveis, dos operadores para cada tipo de interação, a regra geral foi definida da seguinte maneira: Seja $O \in \{M, H, P, B, K\}$ um operador genérico resultante de uma interação do utilizador, e O_i , $i = 1..n$, o valor do operador apresentado na Tabela 5. A regra adotada consiste em determinar $\min\{O_i\}$ e o $\max\{O_i\}$ e verificar se $O \in [\min_{O_i} - \sigma_O, \max_{O_i} + \sigma_O]$. Caso isto ocorra, é estabelecido um padrão e o tipo de interação prevista, conforme apresentado na Tabela 5.

Tabela 5 – Padrões detetados consoante o tipo de interação

Operadores $\in [\min_{O_i} - \sigma_O, \max_{O_i} + \sigma_O]$					
Tipo de Interação	M	H	P	B	K
Teclado	[18.44, 37.76]	[2.44, 4.83]	[6.33, 20.34]	[1.19, 3.60]	[34.82, 70.76]
Rato	[51.31, 58.71]	[0.31, 6.78]	[29.20, 40.94]	[5.31, 7.44]	[0.0, 0.0]
Rato e Teclado	[32.02, 44.15]	[3.4, 5.96]	[12.09, 21.39]	[0.92, 4.11]	[25.51, 48.37]

Com base nestes padrões, foram testados os três tipos de interação em 10 websites diferentes dos testados anteriormente, sendo os resultados apresentados na Tabela 6. Nesta tabela são apresentados os resultados obtidos, partindo do que foi a interação pedida aos utilizadores.

Tabela 6 – Padrões obtidos em 10 sites diferentes

Fontes*	M	H	P	B	K	Resultado Esperado	Resultado Obtido
[S15]	23.28	3.45	9.48	1.72	62.07	Teclado	Teclado
[S16]	54.66	3.24	35.63	6.48	0.0	Rato	Rato
[S17]	38.03	5.63	15.49	2.82	38.03	Rato e Teclado	Rato e Teclado
[S18]	30.86	4.57	12.57	2.29	49.71	Teclado	Teclado
[S19]	54.66	3.24	35.63	6.48	0.0	Rato	Rato
[S20]	39.21	4.98	18.26	3.32	34.23	Rato e Teclado	Rato e Teclado
[S21]	27.05	3.21	13.23	2.40	54.11	Teclado	Teclado
[S22]	38.93	5.24	17.30	3.15	35.39	Rato e Teclado	Rato e Teclado
[S23]	26.73	2.64	14.52	2.64	53.47	Teclado	Teclado
[S24]	55.67	4.12	34.02	6.19	0.0	Rato	Rato

* Lista de fontes disponível na Tabela 8 do Apêndice 1

Os resultados coincidiram com os padrões detetados para cada operador, encontrando-se entre o intervalo de valores apresentado. Assim, conclui-se que o modelo sugerido é adequado para obter indicadores de tipos de interação, usando o KLM. A Figura 7 mostra um exemplo de aplicação do website da *National Football League* (NFL), em que o cenário de interação efetuado foi a pesquisa de notícias num cenário de interação baseado no Rato.

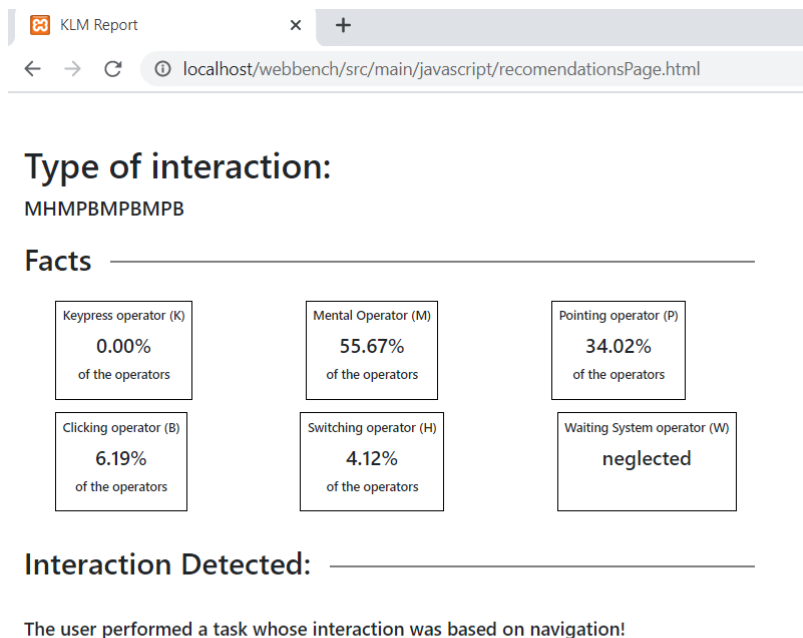


Figura 7 – Indicação do tipo de interação efetuada pelo utilizador no website [S24]

Como é possível constatar através das percentagens disponibilizadas, estas encontram-se dentro do intervalo de valores padrão de cada operador para a estratégia de interação baseada no rato, pelo que, a *framework* não planeada consegue detetar esses valores padrão e indicar ao designer qual foi o tipo de interação efetuada pelo utilizador.

4. Performance

A previsão fornecida pelo KLM pode ser baseada não só no tempo necessário para o utilizador despoletar os eventos na aplicação, mas também, no tempo de resposta da aplicação. Numa aplicação web, esse tempo de resposta depende de recursos acessíveis através de uma rede de computadores sujeitos a problemas de performance na rede ou servidor.

A recente popularidade das PWAs como soluções alternativas às aplicações nativas tradicionais para dispositivos móveis, forçou a constrangimentos rígidos ao nível da performance das aplicações web. Estas aplicações têm de apresentar níveis de performance similares às PWAs. O protocolo HTTP2 é um potenciador de performance no processo de transferência de recursos entre o cliente e o servidor, pois através deste protocolo é possível a requisição de recursos em paralelo, existindo a possibilidade do carregamento de conjuntos de recursos praticamente em simultâneo. Este protocolo trouxe melhorias significativas ao nível da eficiência, velocidade e segurança na comunicação cliente-servidor, sendo suportado pela maioria dos navegadores atuais.

De forma a respeitar os constrangimentos de performance das aplicações web atuais e potenciar o aumento da performance ao nível da comunicação entre o cliente e o servidor utilizando técnicas e protocolos de última geração, desenvolveu-se neste trabalho uma ferramenta para a análise de dependências entre recursos utilizados pelas aplicações web, de forma a assegurar a priorização e transferência proativa destes.

4.1 Identificação do problema

No contexto da *framework* não planeada apresentada no capítulo anterior, foi possível constatar que os desafios do desenho de uma aplicação web prendem-se, não só, com a usabilidade do website, como também com a sua performance, que depende fortemente da resposta do sistema

ao fazer chamadas a *APIs*, ficheiros ou outros recursos que se encontrem alojados no servidor. Se tomarmos como exemplo o trabalho apresentado no capítulo 3, a usabilidade de um website pode estar a seguir todas as recomendações indicadas e não ter mais nenhum tipo de melhoria relevante a implementar, mas se o tempo de resposta do sistema não corresponder às expectativas, dificilmente essa aplicação irá oferecer uma boa experiência de utilização.

A performance como fator potenciador da qualidade na experiência de utilização serviu de motivação para o trabalho de análise do desempenho de diferentes websites, como por exemplo, o número de recursos que muitas vezes são carregados em duplicado ou triplicado podendo estes ser guardados, por exemplo, em cache e evitando-se a exposição a eventos externos, como falhas na rede ou completa inexistência de comunicação, que irão atrasar o carregamento da página.

Os *service workers* permitem o controlo dos recursos a colocar em cache e o momento da sua transferência. Um recurso pode ser transferido por um *service worker* mesmo antes de ser requisitado pelo browser. Esta característica permite requisitar proativamente os recursos que são necessários pela aplicação.

O HTTP2 permite descarregar combinações desses recursos de forma a rentabilizar a sobrecarga imposta pelo processo de comunicação entre o cliente e o servidor.

O caso de estudo apresentado neste capítulo pretende dar resposta às seguintes perguntas:

- Como identificar os recursos que são recorrentemente necessários num serviço web?
- Como agrupar combinações de recursos para transferência entre o servidor e o cliente de forma a maximizar a sua probabilidade de utilização e reduzir o número de interações com o servidor?
- Como identificar os recursos que podem ser obtidos do servidor com a antecedência necessária para evitar atrasos na execução da aplicação?

A performance de um website está dependente das possíveis chamadas ao servidor, e, devido a esse facto, foi efetuado um estudo que identifica quais os recursos mais utilizados em determinados websites, identificando o momento certo para o carregamento desses recursos num contexto de dois cenários de interação: (1) pesquisa no website e (2) registo de um novo utilizador.

4.2 Processo de Recolha de Dados

A *API* do *Selenium WebDriver* (Selenium., 2020) não só consegue extrair toda a informação existente no ficheiro XML acerca dos passos seguidos pelo utilizador durante a sua interação com o website (Secção 3.1), como também tem acesso aos *logs* provenientes do navegador. Esses *logs* fornecem toda a informação/dados resultantes da interação entre o navegador e o servidor. Através destes *logs* é possível obter os eventos relacionados com a performance do website, como as interações cliente-servidor e as suas respetivas *timelines*.

Habitualmente, a informação relativa a cada recurso pode ser consultada nos ficheiros HAR (Odvariko, s.d.). Estes ficheiros são usados para exportar os dados capturados nas interações cliente-servidor e têm uma estrutura *json* (HAR Analyser, 2020). Optou-se pela criação de ficheiros HAR a partir dos *logs*, incluindo, por exemplo, os dados da prioridade, o URL do pedido (*request URL*) ou o tipo de recurso (*XHR, Fetch, Javascript*, etc). Estes dados vão permitir fazer a análise da performance de cada website a testar. A Figura 8 apresenta a arquitetura da solução desenvolvida, baseada na *framework* não planeada abordada nos capítulos anteriores.

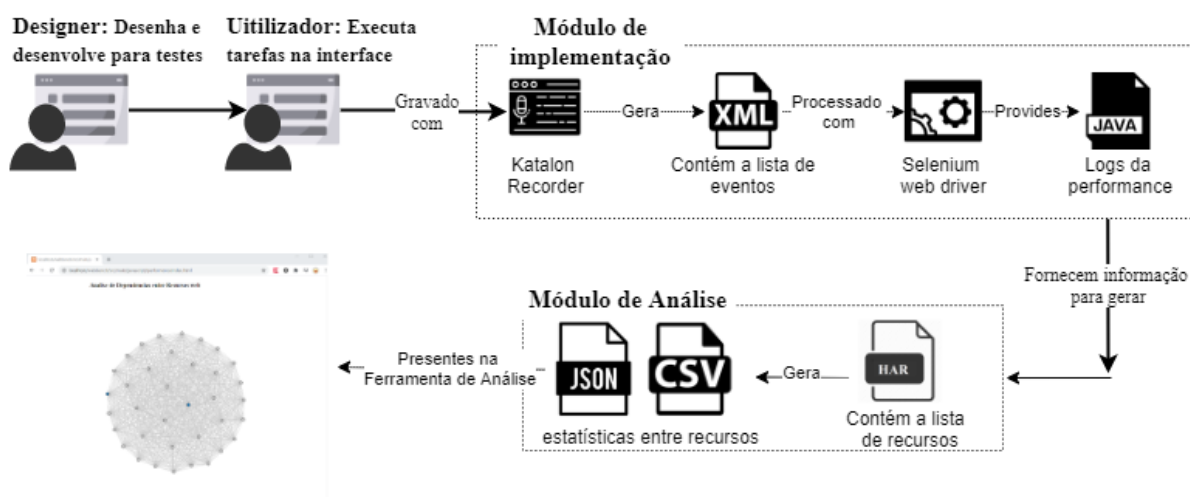


Figura 8 - Visão geral da solução implementada ao nível da performance

Os ficheiros HAR são utilizados para gerar estruturas de dados com resultados da comparação de tempos entre recursos através de um conjunto de estatísticas acerca da relação entre eles. Como o número de recursos em muitos casos de teste era bastante significativo, optou-se pela filtragem por aqueles que têm uma maior prioridade (*VeryHigh* ou *High*) e os que eram armazenáveis em cache. Isto porque os recursos com elevada prioridade fazem parte do caminho crítico de renderização, provocando tempos de espera do utilizador proporcionais ao atraso no *download* destes recursos. Por outro lado, que os recursos que têm a capacidade de ser armazenáveis em cache, são recursos cujo conteúdo não depende de eventos futuros e como tal podem ser descarregados com antecedência. Este aspeto é de extrema importância para este projeto, pois ao se conseguir identificar os recursos que têm esta característica é possível distinguir os recursos que podem ser carregados antecipadamente na página, evitando tempos de espera associados a cada ação do utilizador.

De forma a captar o maior número de recursos possível para cada website, foram efetuadas 20 execuções da aplicação desenvolvida, associadas às interações efetuadas pelo utilizador em cada cenário de teste. Como cenários de teste, foram efetuadas pesquisas de notícias e/ou produtos, assim como, o registo de novos utilizadores em diferentes websites, capturando todos os recursos associados às interações do utilizador.

4.3 Ferramenta de Análise de Dependências

De forma a facilitar a análise dos recursos foi implementada uma ferramenta de análise de dependências entre recursos. Esta ferramenta tem toda a informação alusiva às interações cliente-servidor efetuadas no cenário de interação que se pretender estudar, pois lê um ficheiro JSON gerado a partir dos ficheiros HAR onde é possível obter a informação de cada recurso. Como forma de representação foi criado um grafo onde os nodes são os recursos que irão ser analisados e os links são as relações entre eles. A Figura 9 representa o caso de estudo associado à pesquisa de notícias no website da *British Broadcasting Corporation (BBC)*.

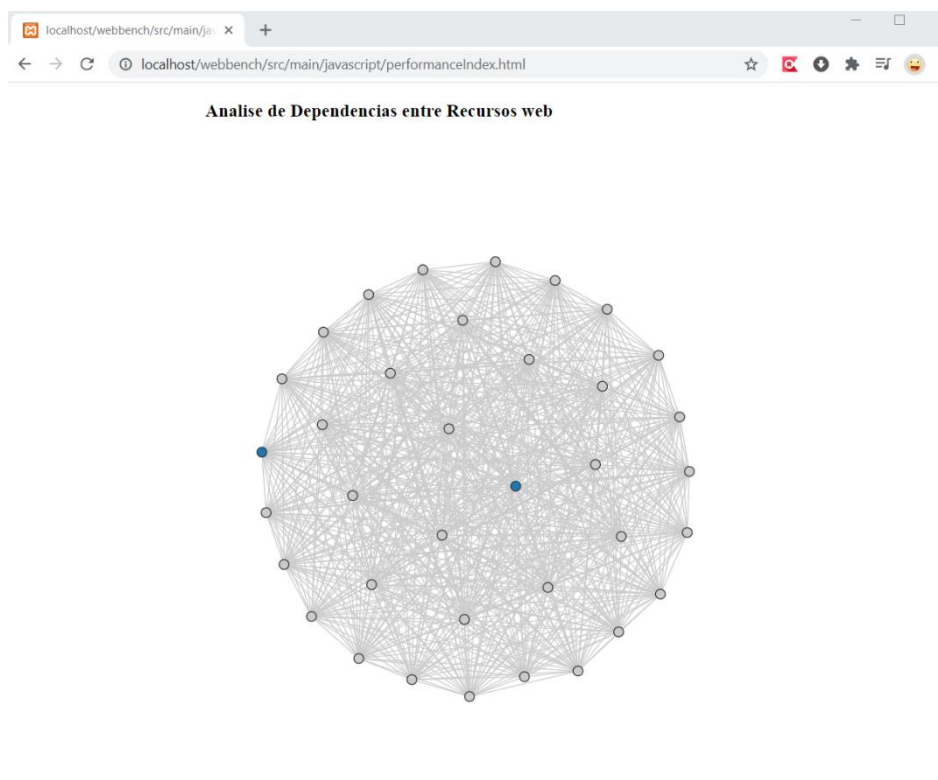


Figura 9 - UI da Ferramenta de Análise de Dependências

Este cenário de interação obteve um total de 36 recursos, sendo representados na ferramenta 36 nodes, dos quais 2 são do tipo *XMLHttpRequest (XHR)* pois estão identificados com a cor azul. Uma das características da ferramenta desenvolvida é a capacidade de mostrar informação ao ser passado o rato por cima de cada node, como é mostrado na Figura 10.

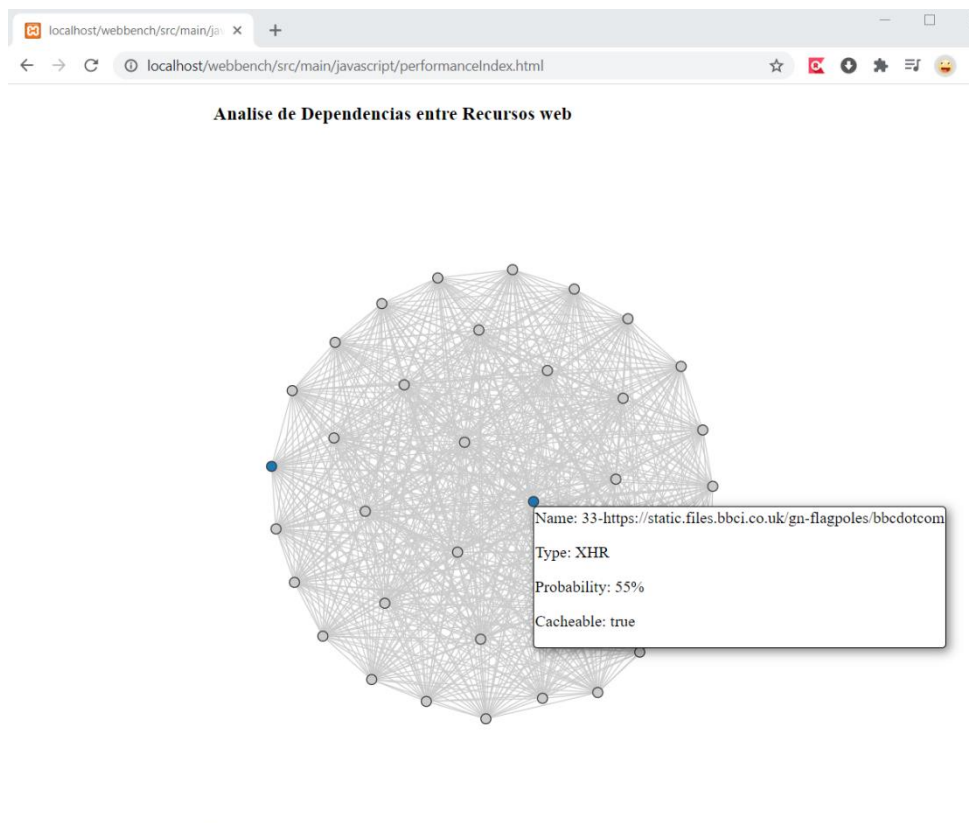


Figura 10 - Informação acerca de cada recurso na Ferramenta de Análise de Dependências

Cada *node* fornece informação útil acerca de cada recurso, como é o caso do URL do pedido, o tipo de recurso, o facto de ser armazenável em cache e a probabilidade desse recurso existir. Uma outra funcionalidade é a filtragem de todos os *nodes* que estão ligados a um determinado *node*. Cada *node* tem a característica de ser clicável, ou seja, se a tecla do lado esquerdo do rato for pressionada duas vezes quando o ponteiro do rato se encontrar em cima de um *node*, apenas são mostrados os nodes que se relacionam com esse *node* em específico. Na Figura 11 é apresentada essa funcionalidade.

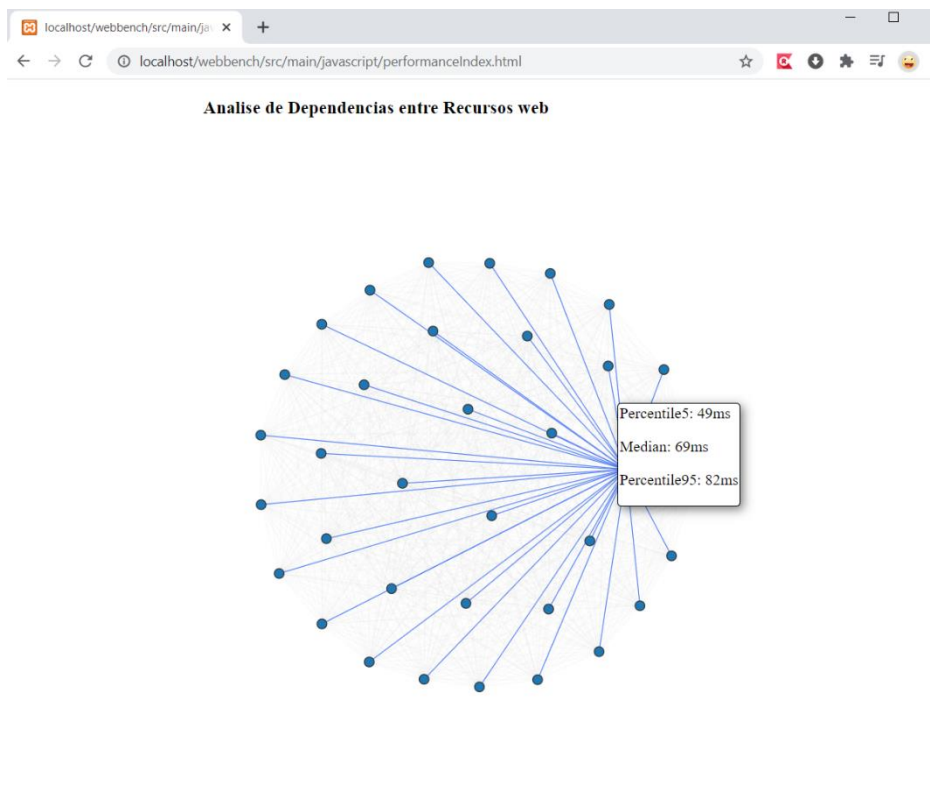


Figura 11 - Relação entre dois recursos na Ferramenta de Análise de Dependências

A ligação entre dois *nodes* fornece informação acerca destes, como o Percentil 5, Percentil 95 e a Mediana. Estes três cálculos fornecem dados estatísticos acerca das interações temporais entre recursos de forma a definir uma ordem para o carregamento destes e priorizando a sua transferência (Secção 4.5).

4.4 Análise de Dependências entre Recursos

De forma a identificar os recursos web para os quais deve ser antecipado o seu carregamento, é realizada uma análise de todos os recursos, calculando as probabilidades de utilização desses recursos no cenário de interação efetuado pelo utilizador. São também efetuadas combinações entre todos os recursos de maneira a calcular a probabilidade destas. No caso de uma combinação ou um recurso isolado terem uma probabilidade de utilização bastante elevada, deve ser antecipado o seu carregamento, e assim, evitar possível falhas de rede e problemas na interação do utilizador com a aplicação web.

4.4.1 Análise Individual de Recursos

Através da identificação dos recursos que são armazenáveis em cache, é possível a obtenção de todos os recursos que foram necessários no cenário de interação utilizado. A grande mais valia destes recursos é que podem ser descarregados em qualquer altura da aplicação, sendo possível ao programador tirar partido desse facto e descarregá-los, por exemplo, no primeiro

carregamento desta, armazenando-os na memória cache e fazendo uso deles consoante as ações efetuadas pelo utilizador no cenário de interação proposto. Posto isto, com base nestes recursos e no número total de execuções da aplicação foi calculada a probabilidade do aparecimento de cada recurso para 20 execuções, através da Equação (3):

$$P_{Rx} = \text{ocorrência}_{Rx} \times 100 / \text{número}_{runs} \quad (3)$$

Assim, a partir da probabilidade de cada recurso é possível verificar se este está a ser constantemente necessário no cenário de interação e averiguar se é um recurso essencial para a interação do utilizador e se a aplicação web ou o navegador não se encontra a tirar partido da cache para o armazenamento desse recurso.

Foram efetuados vários testes em diferentes aplicações web, de forma a ser possível perceber se, consoante o cenário de interação, existiam recursos que estivessem a atrasar o carregamento dos dados na aplicação, contribuindo assim para a diminuição da performance das aplicações web em estudo. A Figura 12 representa a probabilidade de cada recurso no cenário de interação do registo de um novo utilizador no website da *BBC*.

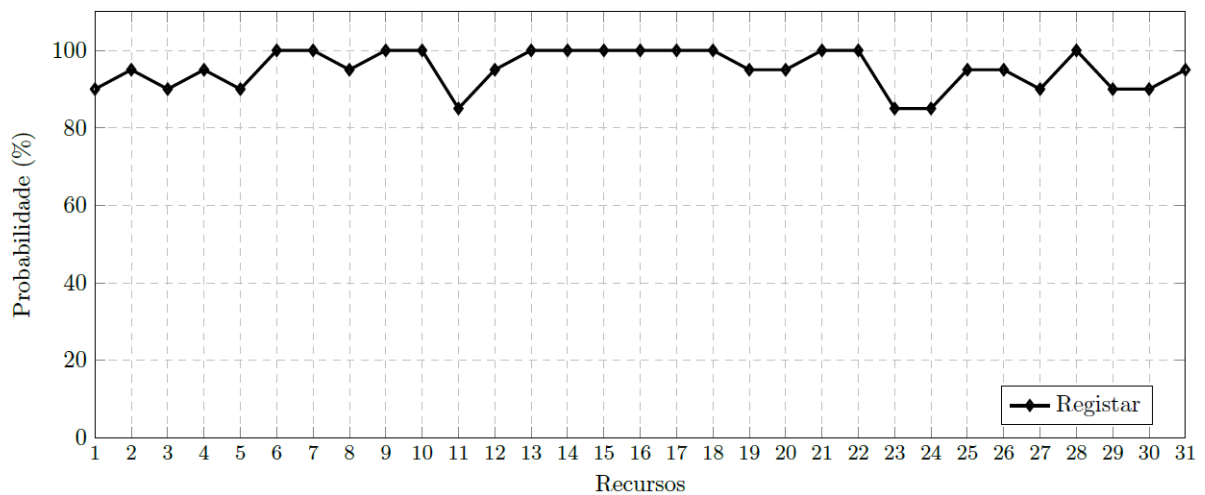


Figura 12 - Registo de um novo utilizador no website *BBC*

Foram identificados um total de 31 recursos, onde existem vários com a probabilidade de 100%, querendo assim dizer que estiveram presentes em cada uma das 20 execuções da aplicação efetuadas. Todos os recursos estão identificados, através do respetivo URL, na Tabela 9 do Apêndice 1. Este tipo de registo é uma situação excecional, pois o utilizador necessita de efetuar mudanças de página para concluir o mesmo, o que significa que irá ter um número mais elevado de recursos do que um registo dito normal, como é o exemplo demonstrado na Figura 13, onde apenas foram identificados 5 recursos.

4 - Performance

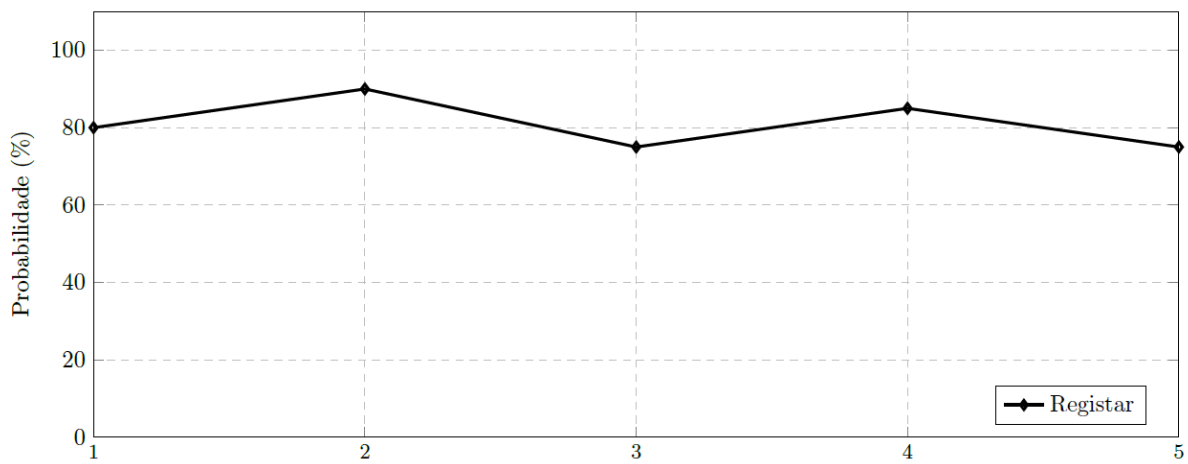


Figura 13 - Registo de um novo utilizador no website de compras online *Walmart*

Foram efetuados mais dois testes num cenário de interação de pesquisa nos mesmos dois websites (*BBC* e *Walmart*). A Figura 14 e a Figura 15 representam essa pesquisa para os dois websites, respetivamente.

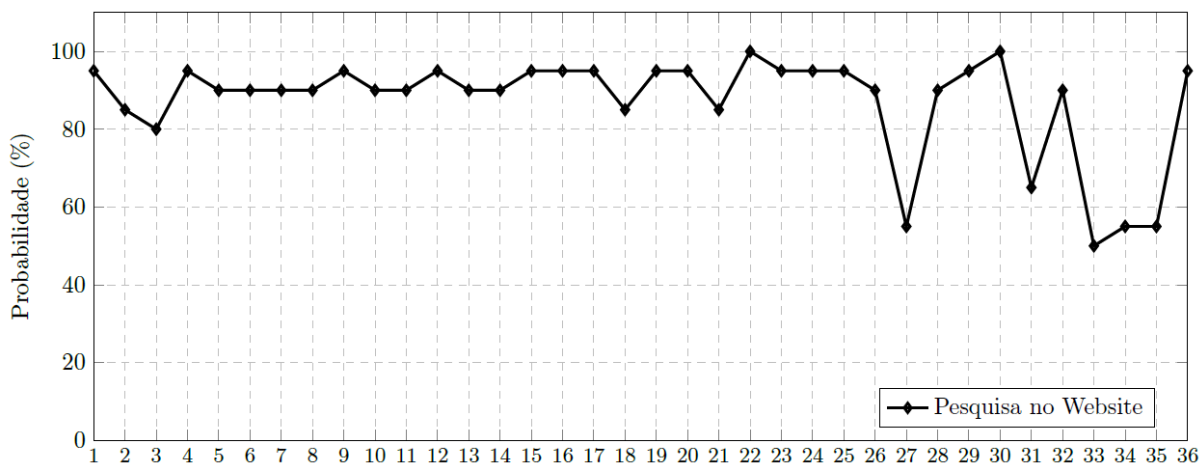
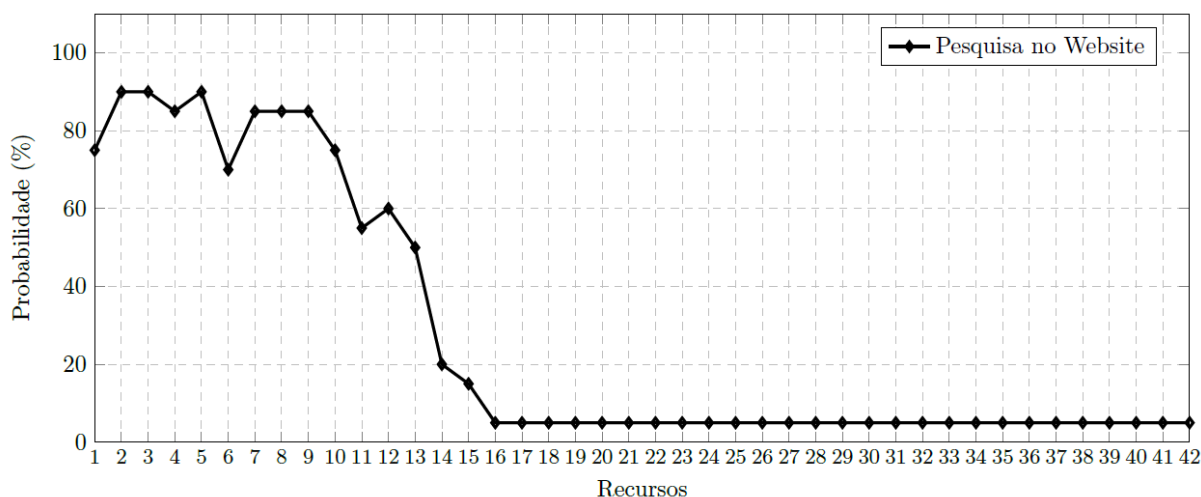


Figura 14 - Pesquisa de notícias no website *BBC*

Figura 15 - Pesquisa de produtos na loja online *Walmart*

Dos 42 recursos envolvidos, apenas 13 têm uma probabilidade de serem requisitados superior a 50%. Os recursos com baixa probabilidade de serem requisitados pelo navegador estarão a beneficiar da antecipação do seu *download*, daí a probabilidade de 5% a partir do recurso 16. No entanto, serão os recursos com elevada probabilidade de serem requisitados ao servidor que beneficiarão do *download* proativo.

Através dos resultados anteriores relativos aos 4 testes efetuados, é possível constatar a existência de um maior número de recursos e uma maior variação na probabilidade dos mesmos, no cenário de interação de pesquisa na aplicação web relativamente ao cenário de interação de registo de um novo utilizador. Isso deve-se à constante mudança de contexto pois, quando uma nova página é carregada, novos recursos são descarregados, aumentando o número total destes na interação efetuada pelo utilizador.

De salientar que durante os 20 *runs*, a cache do navegador não foi limpa, pelo que todos os recursos com probabilidade de 100% foram descarregados do servidor em todas as execuções da aplicação, mesmo sendo recursos passíveis de serem armazenados em cache. Este pressuposto vai permitir identificar os recursos que ainda não estão a ser armazenados em cache pela aplicação ou navegador.

4.4.2 Agrupamento de Recursos

Em muitas situações existem recursos que apenas são necessários no momento da interação com o utilizador e por isso são descarregados precisamente nessa altura. Esses recursos podem beneficiar de um *download* proativo, se o tempo de carregamento deles for bastante significativo para a performance da aplicação, devido à quantidade de dados, tamanho dos recursos ou às falhas existentes na rede. Ao serem novamente requisitados, apenas seria necessário ir à cache e fazer a requisição deles, evitando possíveis atrasos no carregamento do elemento web associado e, conseqüentemente, problemas relacionados com a interação do utilizador.

Para cada execução são analisados e guardados os recursos existentes, sendo que após a análise de todas as execuções é possível ter como resultado uma lista de todos os recursos utilizados com a respectiva informação acerca de cada um, como por exemplo o tamanho, o tempo de resposta, o tipo de recurso e uma lista com a identificação das execuções onde esse recurso foi descarregado. Através desta informação são efetuadas combinações de n elementos para todos os recursos, sendo contado o número de ocorrências de cada combinação em cada execução, verificando se cada recurso existente na combinação foi descarregado em cada uma das execuções. Através desta contagem é calculada a probabilidade do aparecimento de cada combinação através da Equação (4):

$$Prob_{comb_{[Rx,Ry,Rz,Rt]}} = (ocorrências_{comb_{[Rx,Ry,Rz,Rt]}} \times 100) / número_{execuções} \quad (4)$$

De forma a ser feita uma análise mais detalhada acerca de cada combinação, é também calculado o seu tamanho, somando o tamanho de cada recurso, através da Equação (5):

$$Tam_{comb_{[Rx,Ry,Rz,Rt]}} = Tam_{Rx} + Tam_{Ry} + Tam_{Rz} + Tam_{Rt} \quad (5)$$

Para o estudo em causa interessa analisar todas combinações resultantes e identificar padrões em que a probabilidade de cada combinação seja elevada. Se a combinação de determinados recursos ou a sequência de combinações, tiver uma probabilidade elevada, o programador deve carregá-los previamente de forma a amortecer o alto impacto que o não carregamento desses recursos irá trazer para a aplicação. A Figura 16 apresenta as probabilidades das combinações de N recursos em que $2 \leq N \leq 6$. Foi seguido exatamente o mesmo cenário de teste da secção anterior, pelo que o número total de recursos se mantém nos 36 para o website da *BBC*.

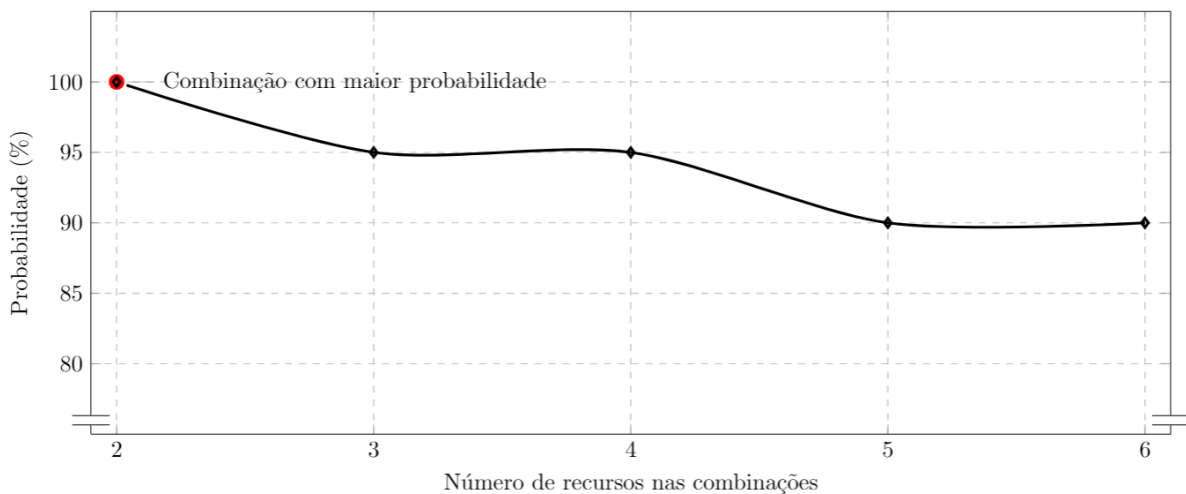


Figura 16 - Análise da probabilidade relativamente à dimensão das combinações

- (2) R_{22}, R_{30} (3) R_{16}, R_{19}, R_{22} (4) $R_{16}, R_{19}, R_{22}, R_{30}$
 (5) $R_4, R_{15}, R_{17}, R_{22}, R_{30}$ (6) $R_{12}, R_{16}, R_{19}, R_{20}, R_{22}, R_{30}$

Foram escolhidas as combinações com probabilidades mais elevadas consoante cada N, tendo em conta a soma dos tamanhos de cada recurso presentes na combinação. Ou seja, no caso de existirem combinações exatamente com a mesma probabilidade, deve ser escolhida a combinação com menor tamanho. Se fosse escolhida a combinação com maior tamanho iria implicar um maior tempo de download de cada recurso, pelo que no final não seria tão vantajoso o *download* dessa combinação.

Como é possível verificar através da Figura 16, quanto maior o valor do N, menor o valor da probabilidade. Como o N representa o número de recursos na combinação, é de esperar que quanto maior o número de recursos, maior a probabilidade de aparecerem na combinação recursos provenientes das últimas execuções da aplicação, que não foram descarregados nas primeiras execuções, pelo que este facto resulta no decréscimo da probabilidade refletida na Figura 16. O conjunto de recursos presente em cada combinação pode ser consultado na Tabela 10 do Apêndice 1, que mostra a informação de todos os recursos presente no cenário de interação de pesquisa para o website *BBC*.

Através desta análise é possível concluir que o sucesso da interação efetuada pelo utilizador está inteiramente dependente do carregamento dos recursos 16, 19, 20, 22 e 30, pois aparecem em todas as combinações. Estes recursos devem ser carregados como um todo de modo a serem carregados em simultâneo.

4.5 Priorização de Transferência de Recursos

Conseguir identificar em que momento o programador deve carregar os recursos de uma página web, assim como a ordem de carregamento dos mesmos, é de extrema importância para a performance de uma aplicação web. A performance de um website depende em grande parte das interações efetuadas pelo utilizador. Do ponto de vista do designer de aplicações, seria bastante útil conseguir antecipar a utilização dos recursos, definindo uma ordem para o carregamento destes e assim evitar atrasos na execução da aplicação e por consequência, na interação do utilizador.

Uma outra análise que foi efetuada, encontra-se relacionada com os tempos de cada recurso e na ordem ou sequência com que estes são carregados na aplicação web. Tal como referido anteriormente, são guardados em ficheiros HAR todos os dados alusivos a cada recurso. Alguns desses dados encontram-se relacionados com o tempo que é necessário para que o recurso seja descarregado e, por consequência, que a informação seja disponibilizada ao utilizador. Esse tempo é calculado através da seguinte Equação (6):

$$T_{recurso} = T_{request} + T_{resposta} \quad (6)$$

Este cálculo é efetuado para cada recurso em cada execução da aplicação, sendo os valores resultantes guardados iterativamente. Com base no tempo total de cada recurso é calculada a

diferença temporal entre o número de recursos existentes, ou seja, se existirem 5 recursos no total, o cálculo é efetuado da seguinte maneira:

$$\begin{aligned}
 T_{[R1,R2]} &= T_{R1} - T_{R2} & T_{[R1,R3]} &= T_{R1} - T_{R3} & T_{[R1,R4]} &= T_{R1} - T_{R4} & T_{[R1,R5]} &= T_{R1} - T_{R5} \\
 T_{[R2,R3]} &= T_{R2} - T_{R3} & T_{[R2,R4]} &= T_{R2} - T_{R4} & T_{[R2,R5]} &= T_{R2} - T_{R5} \\
 T_{[R3,R4]} &= T_{R3} - T_{R4} & T_{[R3,R5]} &= T_{R3} - T_{R5} \\
 T_{[R4,R5]} &= T_{R4} - T_{R5}
 \end{aligned}$$

Com o objetivo de ser feita uma análise detalhada acerca da diferença de tempos entre recursos, no final das 20 execuções da aplicação são calculados o percentil 5, a mediana e o percentil 95 para cada conjunto de recursos com base nos tempos calculados em cada execução. O cálculo do Percentil 5 das diferenças temporais entre recursos fornece a informação mais útil para o estudo em causa, pois quanto menor a diferença temporal ente recursos mais sentido faz o carregamento do segundo recurso a seguir ao carregamento do primeiro.

Foi testado o cenário de interação de pesquisa numa aplicação web. Devido ao facto de o cenário de interação relacionado com o registo de um novo utilizador ter uma quantidade muito escassa de recursos, como foi demonstrado nos capítulos anteriores com o exemplo da Figura 13, chegou-se à conclusão que não faria sentido a sua análise. Assim, foi efetuada uma análise novamente no website da *BBC*, sendo os resultados apresentados na Figura 17. Tal como referido anteriormente, apenas são mostrados os resultados que obtiveram as diferenças temporais entre recursos com um menor percentil 5, encontrando-se no Anexo 2 todos os resultados obtidos.

	R1	R2	R11	R27	R29	R34	R35
R1		6		0		19	20
R3				21			
R4				15			
R5							
R6					16		
R7			20				
R32						16	20

Figura 17 - Análise das diferenças temporais entre recursos

É possível verificar que a diferença temporal entre o recurso 1 e o recurso 27 é 0, podendo-se concluir que existindo a necessidade de priorizar recursos num pedido ao servidor, os recursos mais próximos obteriam uma maior probabilidade de serem carregados em simultâneo, assumindo-se com a maior prioridade. No entanto essa probabilidade teria de ser combinada com a análise das combinações.

5. Conclusões e Trabalho Futuro

Neste capítulo são apresentadas as principais conclusões retiradas com a elaboração deste projeto, bem como potenciais trabalhos a desenvolver baseados no trabalho realizado.

5.1 Conclusões

A existência de uma ferramenta para auxiliar o designer na avaliação da implementação de interfaces com uma elevada performance e que sejam ao mesmo tempo usáveis para o utilizador, é, já há alguns anos uma necessidade evidente. A grande maioria das ferramentas existentes, apesar de implementarem o modelo KLM, não conseguem indicar melhorias para as interfaces das aplicações em análise, nem forma de aumentar o desempenho do utilizador na interação com estas. Para além disso, pressupõem a existência de um caminho predefinido para a conclusão de uma tarefa, retirando ao utilizador a liberdade de execução de tarefas numa dada interface.

A *framework* proposta difere das demais existentes na literatura por utilizar o modelo KLM no cálculo dos tempos de interação do utilizador com as aplicações web e na extração de indicadores de desempenho do utilizador que visam a melhoria das interfaces desenhadas. Através da simplificação do modelo de entrada alusivo à tarefa orientada para o utilizador, apenas lhe é indicado o objetivo da tarefa a ser executada, tendo ele que executar tarefas não planeadas na aplicação web de forma a atingir o objetivo proposto. Para além disso, é detetado automaticamente o tipo de interação que o utilizador realiza na página web em que está a interagir. Embora fora do âmbito deste trabalho de mestrado, esta funcionalidade permitirá agilizar a implementação de um sistema de recomendações ao designer, baseado em standards para a web.

Através da ferramenta de análise das dependências entre recursos são analisadas as interações cliente-servidor resultantes da interação do utilizador com a aplicação web, sendo examinados os tempos de descarregamento dos recursos na rede e na maneira como estes podem influenciar o desempenho de uma aplicação web. Assim, é definida uma ordem para o carregamento destes recursos de forma a priorizar a sua transferência. São indicados ao designer, através da mostragem das probabilidades, os recursos que poderão estar a atrasar o carregamento dos elementos web, assim como aqueles que poderão estar mais vulneráveis a atrasos e/ou falhas na rede. Através do cálculo das combinações de recursos é possível definir um conjunto de recursos que devem ser carregados imperativamente no carregamento da página, de forma a poderem ser utilizados quando assim for necessário, através do uso da memória cache.

5.2 Trabalho Futuro

O trabalho subjacente a uma tese de mestrado permite encontrar a solução para um problema, mas também abre novas perspectivas de trabalho que assentam na solução apresentada para esse problema. Assim, o trabalho realizado nesta tese abre portas a trabalho a realizar em diversas áreas:

- Criação de sistema de recomendações baseado nos indicadores de desempenho obtidos;
- Proceder à captura de emoções dos utilizadores durante a realização de tarefas e sobrepor essa *timeline* à *timeline* extraída pelo KLM de forma a medir a relação entre os operadores KLM e as emoções do utilizador.
- Analisar o impacto do operador W no tempo de execução de tarefas. É um aspeto não explorado na literatura associada ao KLM, uma vez que este, habitualmente, é negligenciado

Relativamente à criação de um sistema de recomendações, o principal objetivo centra-se na análise das estatísticas e dos indicadores de desempenho obtidos de forma a identificar determinados padrões que possam ajudar o designer de interfaces a melhorar o design da aplicação web. Efetivamente, quando detetado um tipo de interação por parte do utilizador, as *guidelines* variam, o que permite analisar a *string* KLM nesse contexto. Para atingir este objetivo, é necessário, em primeiro lugar construir uma base de dados que contenha a informação do conjunto (vasto) de *guidelines* existentes para páginas web, usando algoritmos de recomendação híbridos (Burke, 2002) e usando os indicadores de desempenho implementados nesta tese. Uma outra funcionalidade que se está planeada é a extração de emoções do utilizador de forma a conseguir analisar o seu estado de espírito no momento da interação com a página web. O principal objetivo é verificar se as emoções do utilizador interferem com o tempo calculado pelo KLM comparando a *timeline* das emoções com a *timeline* extraída pelo KLM. É espectável que emoções negativas estejam associadas a transições de rato-teclado e apontar no ecrã. Tal irá permitir inferir sobre o impacto de cada

operador KLM nas emoções do utilizador na execução das tarefas. Para tal, terão que ser usadas técnicas de Machine Learning para a captura e processamento de emoções e a construção de um modelo que relacione a sequência de operadores KLM com a emoção manifestada pelo utilizador. Tal irá permitir medir o valor da emoção de cada operador KLM num contexto de interação.

Por último, pretende-se analisar o impacto do operador W do modelo KLM no tempo de execução de tarefas por parte do utilizador. Num contexto ideal, uma interface é 100% usável e o tempo de execução de uma tarefa é o mínimo possível, tendo em conta a utilização ótima de operadores. No entanto, como referido, um fator muito importante não é considerado nos modelos existentes e, na literatura, sugere-se que o tempo de resposta do sistema, W (denotado por $R(t)$ na Tabela 1), seja “medido”. Tipicamente assume-se que esse tempo é curto e não mensurável. No entanto, nem sempre tal acontece, e W poderá passar a ter um impacto bastante grande na interação considerada perfeita. Este estudo permitirá determinar de que forma alterações de performance influenciam a produtividade do utilizador, identificando padrões de desenho que permitam reduzir a sensibilidade da UX a variações de performance. O principal objetivo é o desenvolvimento de uma ferramenta que permite testar essa sensibilidade relativamente a variações de performance que estão evidenciadas no operador W do KLM, ou seja, no tempo de resposta do sistema. De forma a atingir o objetivo é necessário identificar os pressupostos de falha e as métricas de análise. Os pressupostos de falha centram-se na degradação de performance e na completa ausência de comunicação (inexistência de rede). As métricas de análise estão relacionadas com a utilização do tempo previsto pelo KLM na iminência de condições de falhas aleatórias, com menor ou maior grau de severidade e de degradação de performance permanente.

REFERÊNCIAS

- A., L. (2018). SLMA and Novel Software Technologies for Industry 4.0. . *Advances in Intelligent Systems and Computing*, vol 889, 170-184.
- Alan Dix, J. F. (2004). *Human-Computer Interaction Third Edition*. Pearson Education Limited.
- Alix Goguey, G. C. (2018). Storyboard-Based Empirical Modeling of Touch Interface Performance. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*.
- Al-Megren, S. &.K. (2018). A Systematic Review of Modifications and Validation Methods for the Extension of the Keystroke-Level Model. *Advances in Human-Computer Interaction*.
- AMA, A. p. (2019). *Usabilidade.gov.pt*. Retrieved from <https://usabilidade.gov.pt/toque>
- Amanda Swearngin, M. B. (2013). Human performance regression testing. *Proceedings of the 2013 International Conference on Software Engineering*.
- Amit Mishra, A. A. (2017). The Importance of Usability of a Website. . *The IUP Journal of Information Technology*, Vol. XIII.
- Andreas B. Hansen, T. A.-M. (2017). Progressive Web Apps: The Possible Web-native Unifier for Mobile Development. *Scitepress Science and Technology Publications*.
- Anna Bouch, A. K. (2000). Quality is in the eye of the beholder: meeting users' requirements for Internet quality of service. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*,, 297-304.
- Audit web apps with Lighthouse*. (2019). Retrieved from <https://developers.google.com/web/tools/lighthouse>
- Blackmon, M. H. (2005). Tool for accurately predicting website navigation problems, non-problems, problem severity, and e. *In Proceedings of the sigchi conference on human factors in computing systems*, 31-40.
- Brad A. Myers, J. S. (2016). Improving API Usability. *Communications of the ACM*.
- Burke, R. (2002). *Hybrid Recommender Systems: Survey and Experiments 1*. Retrieved from <http://www.google.com>
- Card, S. K. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7),, 396-410.
- Card, S. K. (1983). The psychology of human-computer interaction. *Lawrence Erlbaum Associates*, 49-51.
- Chi, E. H. (2003). *In Proceedings of the sigchi conference on human factors in computing systems*, 55-512.
- Dingli, A. &. (2011). Useful: A framework to mainstream web site usability through automated evaluation. *International Journal of Human Computer Interaction (IJHCI)*, 2 (1),, 10.

REFERÊNCIAS

- Dominic, P. &. (2010). Evaluation method of Malaysian university website: Quality website using hybrid Method. *Interaction Symposium on Information Technology (ITSim)*.
- Fitts, P. M. (1954). The information capacity of the human motor system in controlling the the amplitude of movement. *ournal of experimental psychology*, 47(6),, 381.
- Gaunt, M. (2020). *Google Web Fundamentals*. Retrieved from Service Workers: an Introduction: <https://developers.google.com/web/fundamentals/primers/service-workers>
- HAR Analyser*. (2020). Retrieved from Google Admin Toolbox: https://toolbox.googleapps.com/apps/har_analyzer/
- Himali Saitwall, X. F. (2010). Assessing performance of an Electronic Health Record (EHR) using Cognitive Task Analysis. . *International Journal of Medical Informatics*.
- Hogan, L. (2014). *Designing for Performance*. O'Reilly Media.
- Ivory, M. Y. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33 (4),, pp. 470-516.
- Jean-Chrysostome Bolot, P. H. (1996). Performance engineering of the World Wide Web: Application to dimensioning and cache design. *Computer Networks and ISDN Systems*, 1397-1405.
- John, B. E. (2004). Predictive human performance modeling made easy. *In Proceedings of the sigchi conference on human factors in computing systems*, 455-462.
- Katalon. (2020). *Katalon recorder v3.4*:. Retrieved from A powerful selenium ide alternative.: <https://www.katalon.com/resources-center/blog/katalon-recorder-v3-4/>
- Katri Leino, A. O. (2019). Automating Keystroke-level Modeling with Reinforcement Learning. *24th International Conference on Intelligent User Interfaces (IUI '19)*.
- Katsanos, C. K. (2013). Klm form analyzer: automated evaluation of web form filling tasks using human performance models. *In Ifip conference on human-computer interaction*, 530-537.
- Katsanos, C. T. (2006). Infoscent evaluator: a semi-automated tool to evaluate semantic appropriateness of hyperlinks in a web site. *In Proceedings of the 18th australia conference on computer-human interaction: Design: Activities, artefacts and environments*, 373-376.
- Katsanos, C. T. (2010). A survey of tools supporting design and evaluation of websites based on models of human information interaction. *nternational Journal on Artificial Intelligence Tools*, 19(06),, 755-781.
- Landauer, T. K. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2), , 211.
- Lin, J. N. (2001). Denim: an informal tool for early stage web site design. *In Chi'01 extended abstracts on human factors in computing systems*, 205-206.
- Maarten Wijnants, R. M. (2018). HTTP/2 Prioritization and its Impact on Web Performance. *In Proceedings of the 2018 World Wide Web Conference (WWW '18)*, 1755–1764.

REFERÊNCIAS

- Menascé., D. A. (2002). Load testing, benchmarking, and application performance management for the web. *Computer Measurement Group (CMG) Conference*,.
- Michael J.Franklin, M. J. (1997). Transactional client-server cache consistency: alternatives and performance. . *ACM Transactions on Database Systems* 22, 3, 315-363.
- Michael O. Leavitt, B. S. (2006). Research-Based Web Design & Usability Guidelines. *Background and Methodology*.
- Miller, C. S. (2004). Modeling information navigation: Implications for information architecture. *Human-computer interaction*, 19(3),, 225-271.
- Moulin, S. D. (2002). User interfaces and help systems: from helplessness to intelligent assistance. *Artificial Intelligence Review*, 18 (2), 117-157.
- Nadeem Ahmad, U. S. (2015). Usability of Online Assistance From Semiliterate Users' Perspective. *International Journal of Human-Computer Interaction*, 31 (1), 55-64.
- Nielsen, J. (2012). *Usability 101: Introduction to Usability*.
- Nielsen, J. T. (2002). Homepage usability: 50 websites deconstructed (Vol. 50). *New Riders Indianapolis, IN*.
- Odvarko, J. (. (n.d.). *Arquivos HAR*. Retrieved from <http://www.softwareishard.com/blog/har-12-spec/>
- Osmani, A. (2017). A Pinterest case Progressive Web App Performance Case Study.
- Patton, E. W. (2010). Sanlab-cm: A tool for incorporating stochastic operations into activity network modeling. *Behavior research methods*, 42(3),, 877-883.
- Paul Holleis, F. O. (2007). Keystroke-level model for advanced mobile phone interaction. *CHI '07 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.
- Pommeranz, A. B. (2012). Designing interfaces for explicit preference elicitation: a user-centered investigation of preference representation and elicitation process. *User Modeling and User-Adapted Interaction*,22(4-5), pp. 357–397.
- Rachel Bellamy, B. J. (2011). Deploying CogTool: Integrating Quantitative Usability Assessment into Real-World Software Development. *ICSE '11 Proceedings of the 33rd International Conference on Software Engineering*.
- Raimund Schatz, M. F.-K. (2014). QoE-Based Network and Application Management. *Springer International Publishing*.
- Ratcliff, C. (2019, October). *12 fascinating UX and user research stats to help your business case*.
- Rebai Rim, M. M. (2017). Evaluation Method for an Adaptive Web Interface: GOMS Model. *Intelligent Systems Design and Applications*.
- Selenium. (2020). *Selenium webdriver*. Retrieved from Retrieved 2020-09-23,; <https://www.selenium.dev/projects/>
- Shaikh, J. F. (2010). Quality of Experience from user and network perspectives. *Ann. Telecommun.* 65,, 47-57.
- Shitkova, M. H. (2015). Towards Usability Guidelines for Mobile Websites and Applications. *Wirtschaftsinformatik*.

REFERÊNCIAS

- Szymański J.M., S. J. (2015). Eye Tracking in Gesture Based User Interfaces Usability Testing. *Springer, Cham*.
- Teo, L. &. (2008). Cogtool-explorer: towards a tool for predicting user interaction. *In Chi'08 extended abstracts on human factors in computing systems*, 2793-2798.
- Van Oostendorp, H. &. (2007). Using a cognitive model to generate web navigation support. *International journal of human-computer studies*, 65(10),, 887-897.
- Zaina., G. d. (2018). Analyzing User Experience in Mobile Web, Native and Progressive Web Applications: A User and HCI Specialist Perspectives. *In Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems*, 1-11.

APÊNDICE 1

Catorze aplicações web foram usadas para realizar os testes especificados na Secção 3.3 e são identificados na Tabela A: (a) interações do rato apenas com base na navegação; (b) interações de teclado entre campos usando a tecla Tab; ou (c) combinação de interações de rato e teclado.

Tabela 7 – Websites usados nos testes e correspondentes contextos de interação

Identificador	Website	(a)	(b)	(c)
[S01]	CNN (https://edition.cnn.com/)	x	x	x
[S02]	New York Times (https://www.nytimes.com/)	x	x	x
[S03]	Walmart (https://www.walmart.com/)	x	x	x
[S04]	Spotify (https://www.spotify.com/)		x	x
[S05]	Lion of Porches (https://www.lionofporches.com/)		x	x
[S06]	BBC (https://www.bbc.com/)	x	x	x
[S07]	Reddit (https://www.reddit.com/)	x	x	x
[S08]	The Guardian (https://theguardian.com/)		x	x
[S09]	LinkedIn (https://www.linkedin.com/)		x	x
[S10]	GitHub (https://github.com/)	x	x	x
[S11]	Ebay (https://www.ebay.co.uk/)	x		
[S12]	Stack Overflow (https://stackoverflow.com/)	x		
[S13]	Forbes (https://www.forbes.com/)	x		
[S14]	Reuters (https://www.reuters.com/)	x		

Tabela 8 – Websites utilizados para os testes alusivos à identificação de padrões

Identificador	Website
[S15]	Farfetch (https://www.farfetch.com/)
[S16]	Fox News (https://www.foxnews.com/)
[S17]	The Wall Street Journal (https://www.wsj.com/)
[S18]	The Telegraph (https://www.telegraph.co.uk/)
[S19]	Puma (https://eu.puma.com/)
[S20]	Logitech (https://www.logitech.com/)
[S21]	Levis (https://www.levi.com/)
[S22]	Gant (https://gant.com/)
[S23]	El Corte Ingles (https://www.elcorteingles.com/)
[S24]	NFL (https://www.nfl.com/)

APÊNDICE 1

Identificação de todos os recursos presentes no cenário de interação do registo de um novo utilizador no website *BBC*.

Tabela 9 – Recursos estudados para o registo no website *BBC news*

Identificador	URL	Tipo
[R1]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/js/require.min.js	Script
[R2]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/css/orb-ltr.min.css	Stylesheet
[R3]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Rg.woff2	Font
[R4]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Lt.woff2	Font
[R5]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Bd.woff2	Font
[R6]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Bdlt.woff2	Font
[R7]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Lt.woff2	Font
[R8]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Ltlt.woff2	Font
[R9]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_ExBd.woff2	Font
[R10]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_ExBdlt.woff2	Font
[R11]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Md.woff2	Font
[R12]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Mdlt.woff2	Font
[R13]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Rg.woff2	Font
[R14]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Lt.woff2	Font
[R15]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Bd.woff2	Font
[R16]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Bdlt.woff2	Font
[R17]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Lt.woff2	Font

APÊNDICE

[R18]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Ltlt.woff2	Font
[R19]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_ExBd.woff2	Font
[R20]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_ExBdlt.woff2	Font
[R21]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Md.woff2	Font
[R22]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Mdlt.woff2	Font
[R23]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/js/api-forge-free.min.js	Script
[R24]	https://static.files.bbc.co.uk/account/id-cta/1.51.14/style/id-cta.css	Stylesheet
[R25]	https://nav.files.bbc.co.uk/orbit-webmodules/0.0.2-556.7ef1b51/cookie-banner/cookie-library.min.js	Script
[R26]	https://static.files.bbc.co.uk/account/id-account/1999/css/app.min.css	Stylesheet
[R27]	https://static.files.bbc.co.uk/account/id-account/1999/js/react-bundle.min.js	Script
[R28]	https://static.bbc.co.uk/frameworks/requirejs/lib.js	Script
[R29]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/js/orb.min.js	Script
[R30]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/js/nav.min.js	Script
[R31]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/js/redirect.bundle.js	Script

Identificação de todos os recursos presentes no cenário de interação de pesquisa de notícias no website *BBC*.

Tabela 10 - Recursos estudados para a pesquisa no website *BBC news*

Identificador	URL	Tipo
[R1]	https://www.bbc.co.uk/programmes/m000pbgl	Document
[R2]	https://static.bbc.co.uk/frameworks/jquery/0.4.1/shared/modules/jquery-1.9.1.js	Script
[R3]	https://programmes.files.bbc.co.uk/programmes-frontend/js/rv-bootstrap-5faf60a64a.js	Script

APÊNDICE 1

[R4]	https://programmes.files.bbc.co.uk/programmes-frontend/css/ds2013-a2111b537b.css	Stylesheet
[R5]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/js/require.min.js	Stylesheet
[R6]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/css/orb-ltr.min.css	Script
[R7]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Rg.woff2	Font
[R8]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Rg.woff2	Font
[R9]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_It.woff2	Font
[R10]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_It.woff2	Font
[R11]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Bd.woff2	Font
[R12]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Bd.woff2	Font
[R13]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_BdIt.woff2	Font
[R14]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_BdIt.woff2	Font
[R15]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Lt.woff2	Font
[R16]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Lt.woff2	Font
[R17]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_LtIt.woff2	Font
[R18]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_LtIt.woff2	Font
[R19]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_ExBd.woff2	Font
[R20]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_ExBd.woff2	Font
[R21]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_ExBdIt.woff2	Font
[R22]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_ExBdIt.woff2	Font

APÊNDICE

[R23]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_Md.woff2	Font
[R24]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_Md.woff2	Font
[R25]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSans_W_MdIt.woff2	Font
[R26]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/font/BBCReithSerif_W_MdIt.woff2	Font
[R27]	https://nav.files.bbc.co.uk/orbit/ece9cb048f668bbc4cc6214737a6cdb6/js/api-forge-free.min.js	Script
[R28]	https://static.files.bbc.co.uk/account/id-cta/1.51.14/style/id-cta.css	Stylesheet
[R29]	https://nav.files.bbc.co.uk/searchbox/f8f1f2fe025bd744351a195a6b0840ee/css/main.css	Stylesheet
[R30]	https://nav.files.bbc.co.uk/orbit-webmodules/0.0.2-556.7ef1b51/cookie-banner/cookie-library.min.js	Script
[R31]	https://branding.files.bbc.co.uk/branding/live/css/d9b019ce480d50fd24a16d874bc1682d.css	Stylesheet
[R32]	https://www.bbc.co.uk/userinfo	Fetch
[R33]	https://static.files.bbc.co.uk/gn-flagpoles/gnlops	XHR
[R34]	https://static.files.bbc.co.uk/gn-flagpoles/bbcdotcom	XHR
[R35]	https://nav.files.bbc.co.uk/orbit-webmodules/0.0.2-556.7ef1b51/cookie-banner/cookie-prompt/bbccookies.min.css	Stylesheet
[R36]	https://gn-web-assets.api.bbc.com/bbcdotcom/assets/20201111-082934-7aeb88999-4.9.0/style/bbcdotcom-async.css	Stylesheet

Identificação de todos os recursos presentes no cenário de interação de registo no website *Walmart*.

Tabela 11 - Recursos estudados para o registo no website *Walmart*

Identificador	URL	Tipo
[R1]	https://i5.wal.co/dfw/63fd9f59-7a1b/1399b31b-559f-4202-b08a-1e41342cedba/v1/BogleWeb_subset.css-644f112a80a85e5f115b7411a063d07c2a69418a-br.css	Stylesheet
[R2]	https://i5.wal.co/dfw/63fd9f59-fffe/114ef1b5-cfd3-459a-8933-b77eeac7476/v1/min_style.style.12c79963c12b3e072eac.css-ef549b1cc113d71bd97142b752d443e0bb1212ae-br.css	Stylesheet
[R3]	https://i5.walmartimages.com/dfw/63fd9f59-a78c/fcfae9b6-2f69-4f89-beed-f0eeb4237946/v1/BogleWeb_subset-Regular.woff2	Font
[R4]	https://i5.wal.co/dfw/63fd9f59-fffe/114ef1b5-cfd3-459a-8933-b77eeac7476/v1/fd827841624904d4b8f51d20174fa3a4.woff2	Font
[R5]	https://i5.walmartimages.com/dfw/63fd9f59-a78c/fcfae9b6-2f69-4f89-beed-f0eeb4237946/v1/BogleWeb_subset-Bold.woff2	Font

Identificação de todos os recursos presentes no cenário de interação de pesquisa no website *Walmart*.

Tabela 12 - Recursos estudados para a pesquisa no website *Walmart*

Identificador	URL	Tipo
[R1]	https://i5.walmartimages.com/dfw/63fd9f59-2fd9/37326403-98ab-4ea7-80ff-7285478bcaec/v1/desktop.style.e1dcc0032c16fc8ecba0.css	Stylesheet
[R2]	https://i5.walmartimages.com/dfw/63fd9f59-b85f/619fd54e-d5aa-434d-8b1d-5e23da816d68/v1/all-css.069df478.css	Stylesheet
[R3]	https://i5.walmartimages.com/dfw/63fd9f59-a78c/fcfae9b6-2f69-4f89-beed-f0eeb4237946/v1/BogleWeb_subset-Bold.woff2	Font
[R4]	https://i5.walmartimages.com/dfw/63fd9f59-a78c/fcfae9b6-2f69-4f89-beed-f0eeb4237946/v1/BogleWeb_subset-Regular.woff2	Font
[R5]	https://i5.walmartimages.com/dfw/63fd9f59-b85f/619fd54e-d5aa-434d-8b1d-5e23da816d68/v1/header-js.es.5c632484.js	Script
[R6]	https://i5.walmartimages.com/dfw/63fd9f59-2fd9/37326403-98ab-4ea7-80ff-7285478bcaec/v1/6e03da290d64bf9563b86a26f3c0dfa3.woff2	Font

APÊNDICE

[R7]	https://i5.walmartimages.com/dfw/63fd9f59-b85f/619fd54e-d5aa-434d-8b1d-5e23da816d68/v1/common-chunks.es.c5e5be5a.js	Script
[R8]	https://i5.walmartimages.com/dfw/63fd9f59-b85f/619fd54e-d5aa-434d-8b1d-5e23da816d68/v1/footer-js.es.f15e8b37.js	Script
[R9]	https://www.walmart.com/globalnav/vanilla-fragments/mobile/get-all	XHR
[R10]	https://i5.walmartimages.com/dfw/63fd9f59-b85f/619fd54e-d5aa-434d-8b1d-5e23da816d68/v1/index.es.80979ea3.js	Script
[R11]	https://i5.walmartimages.com/dfw/63fd9f59-b85f/619fd54e-d5aa-434d-8b1d-5e23da816d68/v1/account-menu.es.91e64a50.js	Script
[R12]	https://i5.walmartimages.com/dfw/63fd9f59-b85f/619fd54e-d5aa-434d-8b1d-5e23da816d68/v1/index.es.68af380c.js	Script
[R13]	https://pagead2.googlesyndication.com/getconfig/sodar?sv=200&tid=gpt&tv=2020111701&st=env	XHR
[R14]	https://pagead2.googlesyndication.com/getconfig/sodar?sv=200&tid=gpt&tv=2020111801&st=env	XHR
[R15]	https://pagead2.googlesyndication.com/getconfig/sodar?sv=200&tid=gpt&tv=2020111901&st=env	XHR
[R16]	https://i5.wal.co/dfw/63fd9f59-161f/f0d7612f-54fb-4910-a9c7-20c65d03f878/v1/fd827841624904d4b8f51d20174fa3a4.woff2	Font
[R17]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/main.style.dade7a9e5a86fc33800e.css	Stylesheet
[R18]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/deals-SearchProductResult.dade7a9e5a86fc33800e.chunk.js	Script
[R19]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/deals-app.dade7a9e5a86fc33800e.chunk.js	Script
[R20]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/vendors~benefits-banner~category-app~category-navbar~category-navigation-enhanced~category-shop-by-c~d270bd3f.dade7a9e5a86fc33800e.chunk.js	Script

APÊNDICE 1

[R21]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/1.dade7a9e5a86fc33800e.chunk.js	Script
[R22]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/vendors~benefits-banner~category-app~category-top-nav~collection-highlights~curated-category-carouse~fe31db1c.dade7a9e5a86fc33800e.chunk.js	Script
[R23]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/vendors~category-app~collection-highlights~curated-category-carousel~deals-listing~editorial-product~e147b47f.dade7a9e5a86fc33800e.chunk.js	Script
[R24]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/category-app.dade7a9e5a86fc33800e.chunk.js	Script
[R25]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/breadcrumbs-v2.dade7a9e5a86fc33800e.chunk.js	Script
[R26]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/vendors~skinny-banner.dade7a9e5a86fc33800e.chunk.js	Script
[R27]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/skinny-banner.dade7a9e5a86fc33800e.chunk.js	Script
[R28]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/main.bundle.dade7a9e5a86fc33800e.js	Script
[R29]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/vendors~department-listing-container~redux-visual-facet.dade7a9e5a86fc33800e.chunk.js	Script
[R30]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/redux-visual-facet.dade7a9e5a86fc33800e.chunk.js	Script
[R31]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/vendors~deals-listing.dade7a9e5a86fc33800e.chunk.js	Script

APÊNDICE

[R32]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/deals-listing.dade7a9e5a86fc33800e.chunk.js	Script
[R33]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/vendors~deals-LeftHandNavigation~deals-SortRefineMenu.dade7a9e5a86fc33800e.chunk.js	Script
[R34]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/deals-LeftHandNavigation~deals-SortRefineMenu.dade7a9e5a86fc33800e.chunk.js	Script
[R35]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/deals-LeftHandNavigation.dade7a9e5a86fc33800e.chunk.js	Script
[R36]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/vendors~deals-SearchProductResult~deals-SearchResultMobile.dade7a9e5a86fc33800e.chunk.js	Script
[R37]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/deals-SearchProductResult~deals-SearchResultMobile.dade7a9e5a86fc33800e.chunk.js	Script
[R38]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/deals-listing-container.dade7a9e5a86fc33800e.chunk.js	Script
[R39]	https://i5.walmartimages.com/dfw/63fd9f59-9649/b55020d7-c8b5-492e-808e-de7317291551/v1/6e03da290d64bf9563b86a26f3c0dfa3.woff2	Font
[R40]	https://i5.walmartimages.com/dfw/63fd9f59-1701/2ac8400d-8d4f-4cc5-970c-e0de5520d8f8/v1/new_york_css.style.8481c4a9b602d7fca475.css	Stylesheet
[R41]	https://i5.walmartimages.com/dfw/63fd9f59-1701/2ac8400d-8d4f-4cc5-970c-e0de5520d8f8/v1/6e03da290d64bf9563b86a26f3c0dfa3.woff2	Font
[R42]	https://www.walmart.com/typeahead/v2/complete?term=car&cat=0&zipcode=94066	XHR

ANEXO 1 – FORMULÁRIOS DE REGISTO DAS APLICAÇÕES WEB

The New York Times

Create your free account

 Continue with Google

 Continue with Facebook

 Continue with Apple

Or use your email

Email Address

Password

 Show

You agree to receive updates and offers from The Times. You may opt out anytime.

By creating an account, you agree to the [Terms of Service](#) and acknowledge our [Privacy Policy](#).

Create Account

Already have a Times account? [Log in](#)

This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.
© 2020 The New York Times Company [Privacy Policy](#) [Help](#) [Contact Us](#) [California Notices](#)



Create a CNN Account

Be a part of the new CNN.

Already have an account? [Log In](#)

Email address*

Password*

- Yes! CNN and WarnerMedia News & Sports may use my email address along with my interests, interactions, and device data to send me tailored email and other offers through social media or other sites and apps.
- Yes! CNN and WarnerMedia News & Sports may also share those details with [Affiliates](#) so they may send me tailored email and other offers. CNN and WarnerMedia News & Sports are requesting this consent on behalf of their affiliates, which will use the information under their respective privacy policies.

Opt-out by email to WMPPrivacy@warnermediagroup.com, mail to "WarnerMedia Privacy Office, 4000 Warner Blvd., Bldg. 160, Burbank, CA 91522," or by the unsubscribe link in the email.

By clicking Register you confirm you have read and agree to our [Terms and Conditions](#) and acknowledge our [Privacy Policy](#).

*Required



Create your Walmart account

* required field

[Show](#)

Keep me [signed in](#)
Uncheck if using a public device.

Email me about Rollbacks, special pricing, hot new items, gift ideas and more.

By clicking Create Account, you acknowledge you have read and agreed to our [Terms of Use](#) and [Privacy Policy](#).

Create account



Sign up for free to start listening.

SIGN UP WITH FACEBOOK

I agree to the [Spotify terms & conditions](#) and [Privacy Policy](#).

or

Sign up with your email address

What's your email?

Enter your email.

Confirm your email

Enter your email again.

Create a password

Create a password.

What should we call you?

Enter a profile name.

This appears on your profile.

What's your date of birth?

Day

Month

Year

DD

Month

YYYY

What's your gender?

Male Female Non-binary

Please send me Spotify marketing messages.

Share my registration data with Spotify's content providers for marketing purposes. Note that your data may be transferred to a country outside of the EEA as described in our privacy policy.

I agree to the [Spotify terms & conditions](#) and [Privacy Policy](#).

I'm not a robot



SIGN UP

Have an account? [Log in](#).

LION OF PORCHES

Já é cliente registado?

A forma mais rápida de rever as suas seleções, endereços e submeter o seu pedido é escolher esta opção.

Introduza o seu endereço de email ou o seu código de utilizador *

Introduza a sua palavra-chave ou o seu PIN *

[ESQUECEU A PALAVRA-CHAVE?](#)

INICIAR SESSÃO

Criar conta cliente?


Crie a sua conta e comece já a beneficiar de grandes vantagens e de um acompanhamento das suas encomendas.

Endereço de email *

CONTINUAR

OU ACEDA ATRAVÉS DA SUA REDE SOCIAL

 ENTRAR COM FACEBOOK

 ENTRAR COM GOOGLE

LION OF PORCHES

Já é cliente registado?

A forma mais rápida de rever as suas seleções, endereços e submeter o seu pedido é escolher esta opção.

Introduza o seu endereço de email ou o seu código de utilizador *

Introduza a sua palavra-chave ou o seu PIN *

[ESQUECEU A PALAVRA-CHAVE?](#)

INICIAR SESSÃO

Criar conta cliente?

Crie a sua conta e comece já a beneficiar de grandes vantagens e de um acompanhamento das suas encomendas.

Endereço de email *

País *

Nome *

Morada *

Max. 28

Morada (continuação)

Max. 28

Código Postal *

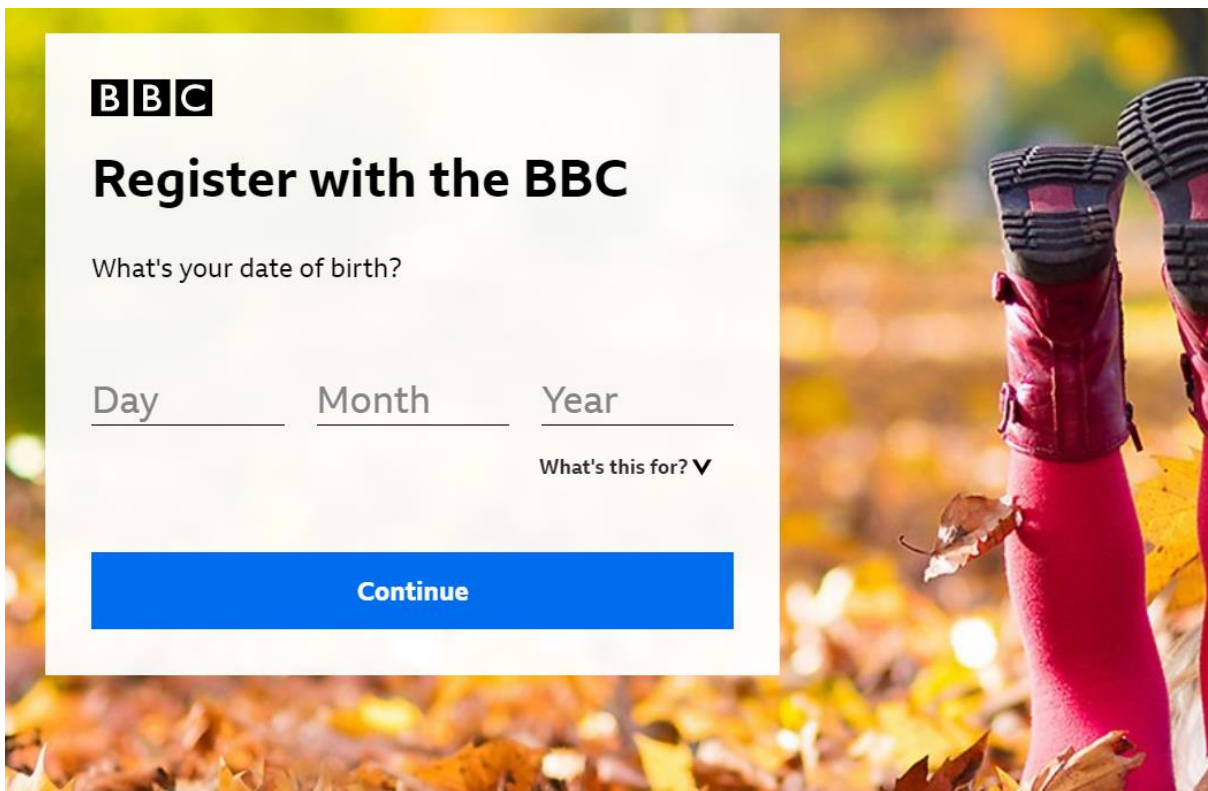
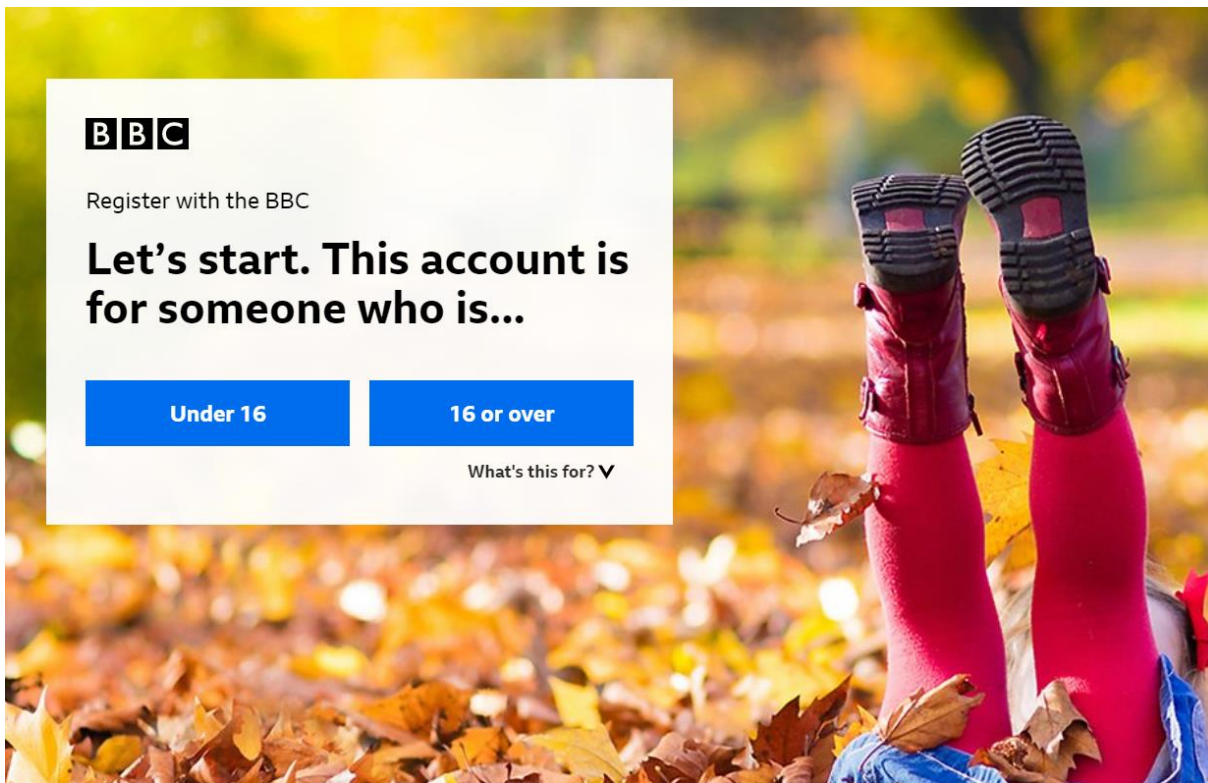
Localidade *

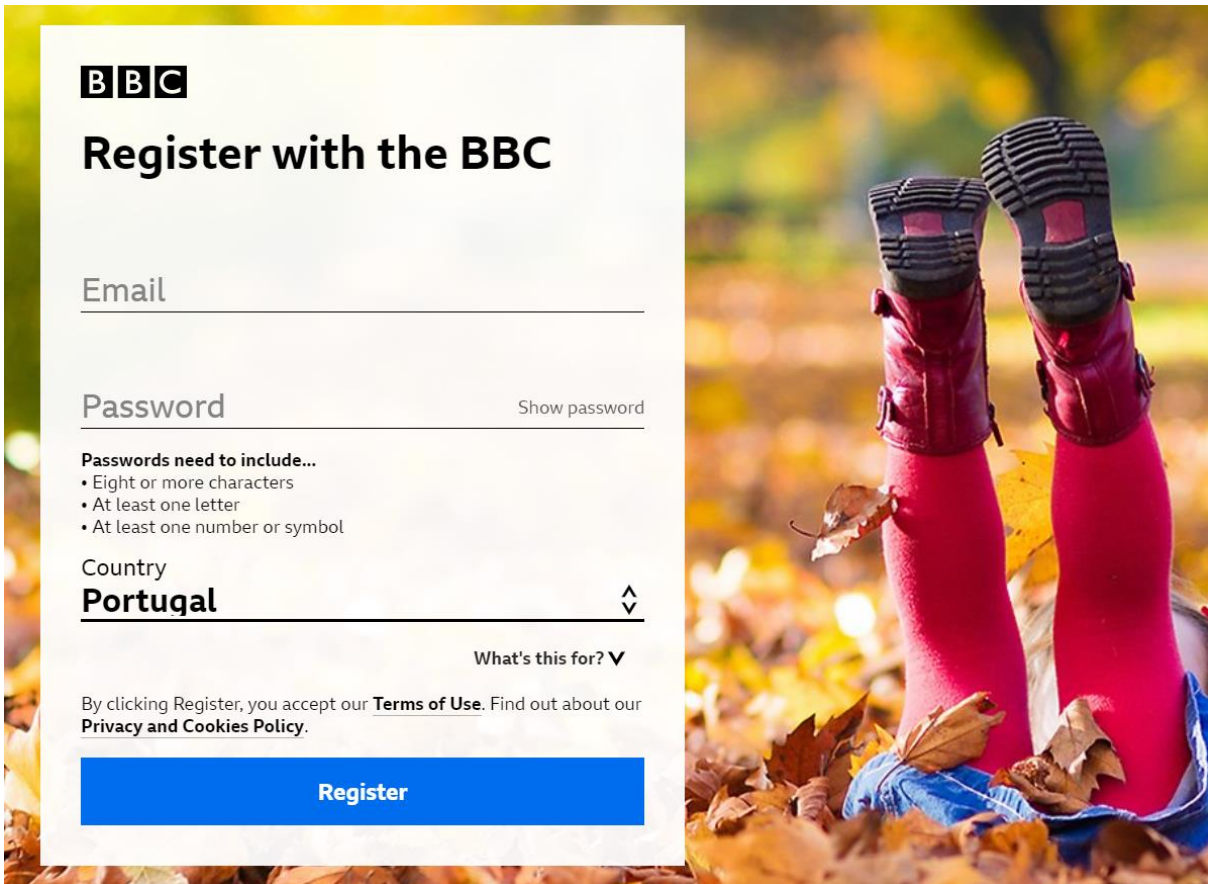
Introduza uma palavra-chave *

Confirme a palavra-chave *

* Ao enviar estes dados está a aceitar a nossa [POLÍTICA DE PRIVACIDADE](#)

FINALIZAR





BBC


Register with the BBC


Email

Password [Show password](#)

Passwords need to include...

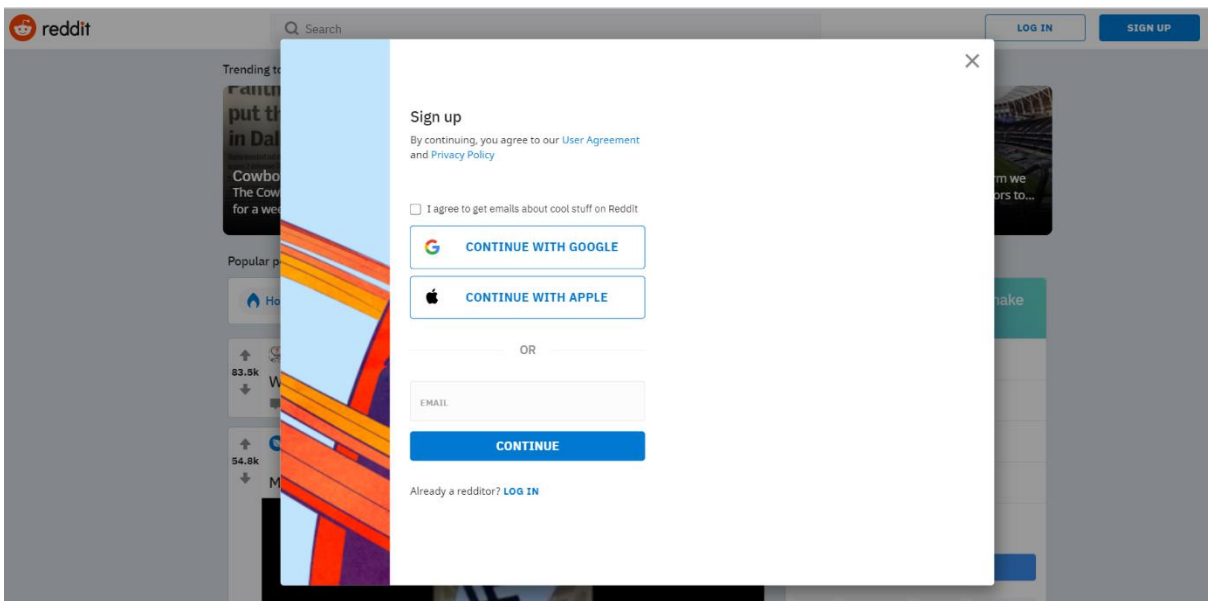
- Eight or more characters
- At least one letter
- At least one number or symbol

Country **Portugal** 

[What's this for?](#) 

By clicking Register, you accept our [Terms of Use](#). Find out about our [Privacy and Cookies Policy](#).

Register



reddit

Search

LOG IN SIGN UP

Trending to

put the
in Dal

Cowbo
The Cow
for a we

Popular p

Ho

83.5k

W


54.8k


M

Sign up

By continuing, you agree to our [User Agreement](#) and [Privacy Policy](#)

I agree to get emails about cool stuff on Reddit

 **CONTINUE WITH GOOGLE**

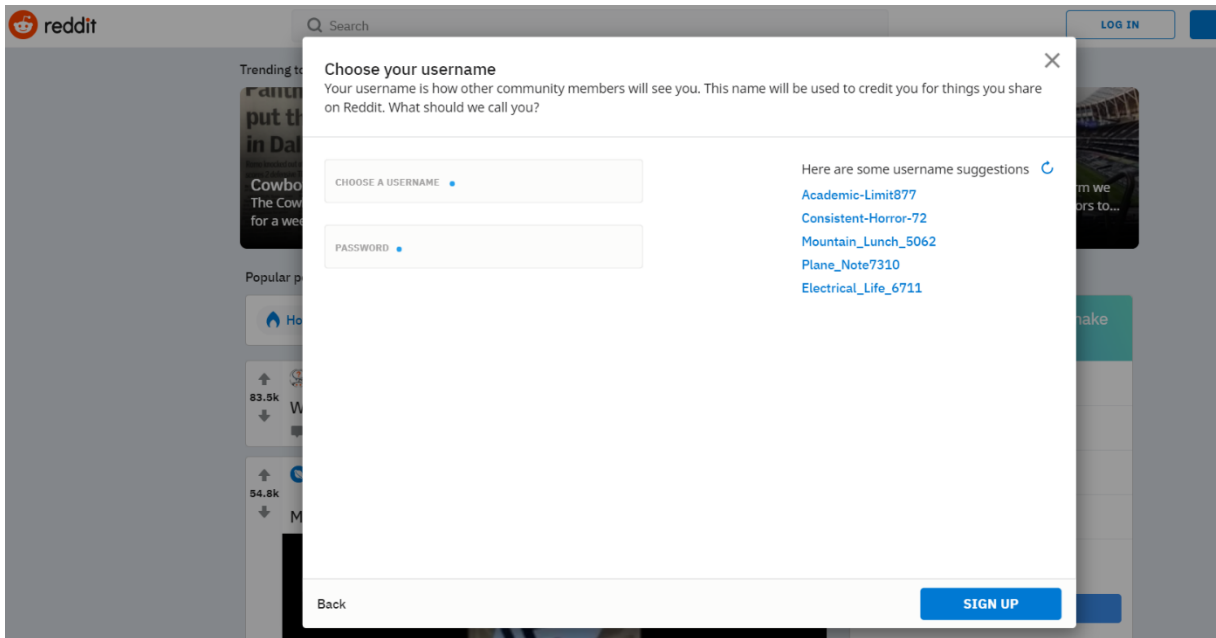
 **CONTINUE WITH APPLE**

OR


EMAIL


CONTINUE


Already a redditor? [LOG IN](#)



Sign in or register

 **Continue with Google**

 **Continue with Facebook**

 **Continue with Apple**

or

Enter your email address

Next →

By proceeding, you agree to our [Terms & Conditions](#).

You also confirm that you are 13 years or older, or that you have the consent of your parent or a person holding parental responsibility.

To find out what personal data we collect and how we use it, please visit our [Privacy Policy](#).

This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.



Register

Looks like you haven't registered with The Guardian yet. Enter your details below to get started.

Email [change](#)

sdgdgdfgdf@gmail.com

First name

Last name

Password

8 to 72 characters

× At least 8 characters

[← Back](#) [Register →](#)

By proceeding, you agree to our [Terms & Conditions](#).

You also confirm that you are 13 years or older, or that you have the consent of your parent or a person holding parental responsibility.

To find out what personal data we collect and how we use it, please visit our [Privacy Policy](#).

This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.



Make the most of your professional life

Email

Password (6 or more characters)

By clicking Agree & Join, you agree to the LinkedIn [User Agreement](#), [Privacy Policy](#), and [Cookie Policy](#).

Agree & Join

or



Join with Google

Already on LinkedIn? [Sign in](#)

Join GitHub

Create your account

Username *

Email address *

Password *

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

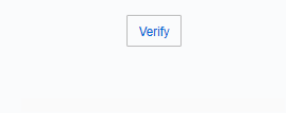
Email preferences

Send me occasional product updates, announcements, and offers.

Verify your account

Please solve this puzzle so we know you are a real person

[Verify](#)



[Create account](#)

ANEXO 2 – RESULTADOS DA DIFERENÇA TEMPORAL ENTRE RECURSOS

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	R26	R27	R28	R29	R30	R31	R32	R33	R34	R35	R36			
R1		6	-51	-59	-507	-1837	-663	-343	-343	-362	-320	-366	-681	-693	-693	-698	-360	-581	-703	-735	-642	-695	-709	-701	-733	-640	0	-563	-482	-553	-41	-183	-1052	19	20	-1116			
R2			-95	-95	-715	-2045	-679	-359	-359	-378	-371	-571	-697	-709	-709	-714	-376	-670	-719	-751	-731	-711	-725	-717	-749	-729	-17	-652	-690	-761	-75	-423	-1063	-5	-2	-1122			
R3				-26	-663	-1993	-615	-295	-295	-314	-281	-519	-633	-645	-645	-650	-312	-580	-655	-687	-641	-647	-661	-653	-685	-639	21	-562	-638	-709	-15	-342	-1017	51	66	-1065			
R4					-657	-1987	-617	-297	-297	-316	-276	-513	-635	-647	-647	-652	-314	-575	-657	-689	-636	-649	-663	-655	-687	-634	15	-557	-632	-703	-23	-328	-1010	57	66	-1070			
R5						-1330	-368	-148	-191	-198	-185	-199	-386	-398	-398	-403	-187	-187	-408	-440	-179	-400	-414	-406	-438	-176	148	-385	-147	-290	65	-231	-915	153	164	-981			
R6							-233	68	51	44	47	43	-43	-55	-55	-60	55	53	-65	-97	47	-57	-71	-63	-95	-13	378	-255	16	-58	228	-14	-673	385	396	-749			
R7								100	67	60	20	-154	-172	-30	-30	-35	71	-279	-40	-72	-340	-32	-46	-38	-70	-338	343	-261	-273	-344	295	-33	-657	377	382	-742			
R8									-138	-145	-86	-362	-380	-350	-350	-355	-134	-382	-360	-392	-443	-352	-366	-358	-390	-441	174	-371	-481	-552	115	-215	-862	197	206	-877			
R9										-127	-110	-362	-380	-350	-350	-355	-89	-409	-360	-392	-470	-352	-366	-358	-390	-468	176	-391	-481	-552	140	-211	-724	208	210	-859			
R10											-82	-235	-319	-331	-331	-336	-84	-381	-341	-373	-442	-333	-347	-339	-371	-440	160	-395	-354	-425	141	-199	-717	216	213	-857			
R11												-314	-465	-477	-477	-482	-144	-299	-487	-519	-360	-479	-493	-485	-517	-358	173	-337	-433	-504	136	-173	-831	205	204	-796			
R12													-315	-327	-327	-332	-82	-412	-337	-369	-473	-329	-343	-335	-367	-471	226	-394	-203	-205	139	-175	-716	221	232	-856			
R13														-75	-12	-37	-110	-412	-137	-137	-473	-119	-126	-126	-133	-471	221	-394	-201	-205	134	-181	-838	216	227	-856			
R14															-58	-69	-115	-380	-142	-142	-441	-124	-131	-131	-138	-439	172	-362	-349	-420	143	-184	-843	225	225	-880			
R15																-42	-141	-418	-168	-168	-479	-150	-157	-157	-164	-477	211	-400	-347	-418	144	-178	-869	208	214	-879			
R16																	-116	-416	-143	-143	-477	-125	-132	-132	-139	-475	217	-398	-347	-418	145	-189	-844	227	232	-886			
R17																		-416	-343	-375	-477	-335	-349	-341	-373	-475	213	-398	-392	-463	126	-187	-728	208	219	-887			
R18																			-350	-382	-124	-342	-356	-348	-380	-91	201	-317	-482	-553	135	-178	-879	198	204	-802			
R19																				-74	-507	-87	-32	-45	-42	-505	208	-428	-364	-435	126	-198	-701	208	219	-887			
R20																					-482	-40	-98	-73	-70	-480	205	-403	-363	-434	128	-201	-701	210	220	-884			
R21																							-334	-348	-340	-372	-81	205	-302	-358	-429	150	-201	-862	210	215	-884		
R22																								-87	-62	-59	-508	196	-431	-459	-530	137	-164	-719	186	191	-936		
R23																									-30	-47	-513	140	-436	-395	-466	121	-167	-712	196	193	-917		
R24																										-55	-489	144	-412	-459	-530	125	-175	-712	200	197	-913		
R25																										-487	126	-410	-404	-475	107	-169	-705	182	179	-936			
R26																												152	-322	-494	-565	133	-167	-831	195	205	-849		
R27																														-659	-696	-767	-87	-406	-1080	-10	-5	-1145	
R28																															-201	-272	330	47	-632	390	391	-723	
R29																																-304	62	-260	-925	139	141	-995	
R30																																	67	-256	-898	140	142	-691	
R31																																		-351	-1014	49	54	-1082	
R32																																						-1090	
R33																																						-1149	
R34																																						-6	-1153
R35																																							-1147