

# Web Performance

Mário Mesquita e F. Jorge F. Duarte

Instituto Superior de Engenharia do Porto  
1150517@isep.ipp.pt; fjd@isep.ipp.pt

**Resumo.** Este artigo resulta de um projeto que teve como objetivo desenvolver um processo de avaliação da performance de uma aplicação de uma empresa e a aplicação de técnicas que permitissem aperfeiçoar o desempenho da mesma. Para isso, foram estudadas várias ferramentas que permitiam avaliar o seu estado inicial permitindo qualificá-la e reunir o máximo de informação relativamente à sua performance. No entanto, para se entender os valores obtidos no estudo do caso, foram pesquisadas e definidas métricas. Depois de agrupada toda a informação, prosseguiu-se para a análise e revisão de código com o objetivo de encontrar pontos críticos que afetassem o comportamento da aplicação. Existem técnicas próprias para lidar com os problemas mais frequentes neste tipo de aplicações, e que, quando aplicadas corretamente, apresentam resultados que podem ser essenciais na sua resolução. Da aplicação dessas técnicas, verificou-se uma melhoria nos tempos de abertura em todas as páginas analisadas.

**Palavras-chave:** Performance, Otimização Web.

## 1 Introdução

A performance das soluções é cada dia mais importante e um fator decisivo na compra e venda de produtos de software [1].

Está comprovado que um segundo de atraso na resposta de uma página reduz a satisfação de um cliente em 16% [2]. Existem várias evidências que demonstram que o desempenho das aplicações é motivo de preocupação e neste documento serão abordados temas como: a avaliação da eficiência dos produtos; a identificação dos problemas que causam o seu mau desempenho; a correção dos problemas identificados; e os benefícios obtidos pela correção dos problemas identificados.

Com a constante evolução das tecnologias e serviços de software, depressa aumentou o seu número e a comercialização de novos produtos. Este enorme aumento de novos produtos, fez com surgissem cada vez menos ideias inovadoras. Com a originalidade a desvanecer-se, outros fatores foram postos em cima da mesa. Era necessário algo que distinguisse um produto dos restantes do mesmo ramo, que tornasse a escolha dos utilizadores mais objetiva. Uma das atitudes foi o começo da valorização cada vez maior do desempenho das aplicações. Os utilizadores depositam mais confiança e preferem visitar páginas cujo desempenho é superior [3]. Existem casos comprovados onde a melhoria de desempenho apresentou um aumento significativo nas visitas e inscrições em páginas [4], e onde o baixo desempenho afetou negativamente alguns negócios [5].

ISBN: 978-989-54758-6-5

Em casos de maior escala, o baixo desempenho pode resultar em grandes perdas financeiras. Milhões de euros são perdidos anualmente devido a problemas de performance na Web [6]. Para grandes aplicações, uns simples milissegundos podem traduzir-se em milhares de euros de prejuízo. A Amazon, uma empresa mundialmente reconhecida pela venda e distribuição de produtos, admitiu uma perda de 1% do valor das suas vendas por cada 100 milissegundos extra no carregamento da sua página. Estes valores são enormes e podem chegar aos 1.6 mil milhões de euros anualmente, apenas com o aumento de 1 segundo [6]. A Google, multinacional americana de tecnologia especializada em serviços e produtos relacionados com a internet e o motor de busca mais utilizado no mundo [7], afirmou que com o aumento de apenas 4 décimas de segundo nos resultados das procuras, perderia aproximadamente 8 milhões de euros por dia, que resultava em milhões de anúncios online não veiculados [6]. Estes exemplos, apesar de reais, são impensáveis para empresas de menor dimensão, sendo, no entanto, essenciais na consciencialização deste problema.

## **2 Estudo**

O objetivo deste trabalho passou por fazer uma avaliação da performance de uma aplicação usando ferramentas existentes no mercado que permitam avaliar o seu estado atual. Ao submeter essa aplicação a processos de teste contínuos, são identificados os pontos mais débeis e que podem ser fonte de lacunas no desempenho destas.

Pretendia-se que fosse realizada uma pesquisa para determinar quais as melhores formas de resolver os problemas relacionados com a performance da aplicação e que estas fossem aplicadas, de acordo com as necessidades apresentadas pelos resultados da primeira avaliação.

Por fim, depois de aplicar estes processos adequadamente, a aplicação foi submetida a testes semelhantes aos da primeira fase, para garantir que os problemas não constituem mais uma adversidade, e esperava-se verificar uma melhoria significativa no desempenho da mesma.

## **3 Técnicas**

Na presente secção serão apresentadas certas técnicas que foram consideradas importantes no desenvolvimento do projeto. A utilização de uma metodologia não invalida as restantes, uma vez que estas podem ser aplicadas simultaneamente e podem complementar-se.

O problema relacionado com a performance não é uma novidade no mundo do desenvolvimento de programas e aplicações. Com a constante luta contra a falta da performance desejada, foram encontradas algumas técnicas para tentar debelar essa situação.

Antes de apresentar essas técnicas, é necessário introduzir certos conceitos para que se possa entender melhor o funcionamento destas.

“The perception of performance is just as effective as actual performance in many cases” [8].

O primeiro conceito a introduzir será o de Perceived Performance [9]. O desempenho percecionado refere-se ao quão rápido um utilizador pensa que uma aplicação é rápida. Não está necessariamente ligado aos valores apresentados nas estatísticas medidas por outros softwares. Quando o objetivo é a otimização de um website é o que o utilizador pensa, que vai fazer a diferença [9].

O segundo e o terceiro conceito a introduzir estão interligados e caracterizam-se por espera ativa e espera passiva [10]. A primeira aparenta ser menos tempo de espera porque significa que a pessoa está ativamente envolvida numa ação ou momento. Consequentemente, o tempo percecionado por ela será menor do que a realidade, uma vez que esta se encontrava concentrada no que está a fazer e não se apercebe do tempo decorrido. O caso da espera passiva consiste precisamente no contrário, já que o tempo de espera será percecionado como maior do que é na realidade. A pessoa está somente focada no tempo que está a passar, não desviando a sua atenção para outros interesses. Com o aumento desta espera, vão aumentar os sintomas de impaciência, frustração e ansiedade da pessoa em questão. Pode-se concluir que, para suavizar o tempo de espera de um utilizador e melhorar a sua perceção de tempo, deve-se reduzir o tempo de espera passiva e dilatar o tempo de espera ativa.

Apresentados os conceitos fundamentais de perceção temporal, são descritas de seguida, as técnicas que ajudam a idealizar abordagens e soluções para problemas como o que é apresentado neste documento.

### 3.1 Preemptive Start

A técnica *preemptive start* [11] consiste em entrar num processo com uma fase ativa e permanecer neste estado o máximo de tempo possível antes de sujeitar o utilizador a uma espera passiva. Como referido anteriormente, o tempo de espera ativa por vezes, faz com que o utilizador nem se aperceba que está, efetivamente, à espera de algo. Consequentemente, para o cérebro humano, um início preemptivo significa a mudança do ponto de início do processo para um ponto mais perto do fim, que dará a ilusão ao utilizador que a duração do processo foi mais curta.

Esta metodologia pode ser usada na Web nas funções de procura, por exemplo. Assumindo que um campo de procura está presente em todas as páginas de uma aplicação, mas os resultados das procuras necessitam de passar por um processo trabalhoso e complexo de ordenação e filtragem. Pode-se então começar um pré-carregamento da informação assim que o utilizador começa a escrever no campo de procura. Isto vai anteceder que a informação já esteja praticamente pronta a ser apresentada quando o utilizador entra no painel ou página que contenha os resultados.

### 3.2 Conclusão antecipada

Em oposição ao início preemptivo, que se foca em aproximar o ponto inicial de um processo mais perto do seu ponto final, a conclusão antecipada [12] tem como objetivo aproximar o ponto final de um processo do seu ponto inicial. O uso mais comum desta metodologia verifica-se nos serviços de transmissão de vídeo. Quando um utilizador pretende visualizar um vídeo, não espera que este termine de carregar totalmente. Assim que uma pequena parte esteja disponível para ver, o vídeo começa. Assim, o ponto final do processo aproxima-se do ponto inicial. O utilizador encontra-se num estado de

espera ativa, pois o vídeo ainda não terminou de carregar, mas já é possível vê-lo enquanto a restante parte é carregada em segundo plano.

Esta metodologia pode ser aplicada em aplicações Web nos momentos em que uma página está a carregar. O intervalo entre o envio de um pedido até à apresentação do resultado no browser corresponde a uma espera passiva do utilizador, portanto pretende-se que este seja o mais curto possível. Uma das soluções passa por apresentar as partes essenciais da página logo que esta esteja pronta. Não há necessidade de esperar que todos os elementos da página estejam carregados, caso não afete a performance da página. Pode-se definir que a informação que não esteja imediatamente visível no início da página, seja apresentada depois dos elementos que o utilizador vai ver em primeiro lugar. Existem bons exemplos de páginas reconhecidas que utilizam este método para oferecer ao utilizador uma experiência fluida e de resposta rápida, como é o caso dos sites da Amazon e do eBay. Depois de efetuar inúmeros testes ao carregamento da página da Amazon, foi possível observar que eram necessários à volta de sete segundos para a página estar completamente carregada. No entanto, graças a aplicações como GTmetrix [13] e PageSpeed Insights [14], é possível observar que, no primeiro segundo, o topo do site já se encontra visível, ficando o utilizador numa espera ativa enquanto o resto da página é carregada, sem se aperceber. Dar prioridade ao conteúdo visível de uma página, é um bom exemplo de conclusão antecipada. Apresentar algo de essencial aos utilizadores assim que for possível, transmite a ideia de que o carregamento da página é realizado mais rapidamente.

### 3.3 Pré-Procure, Pré-renderização e Pré-Carregamento de conteúdo

A pré-procura [15] consiste em notificar o navegador para que este procure certos recursos antes de serem necessários. Serve como uma navegação futura do utilizador. Se uma página solicitar um pedido de pré-procura de recursos para a página seguinte, os recursos e pedidos críticos podem ser concluídos em paralelo. [16] O browser armazena esses recursos em cache permitindo uma entrega mais rápida dos dados assim que forem solicitados.

A pré-renderização [15] é uma prática recente, que consiste em dar indicação ao navegador para descarregar os recursos e renderizar uma página inteira em segundo plano. Esta capacidade pode ser importantíssima quando aplicada na situação certa. Como exemplo, no caso de um utilizador se encontrar a ler os termos e condições de uma página, é muito provável que vá aceitar e terá de ser transferido para uma nova página. Enquanto este faz a leitura dos termos e condições, a segunda página já poderá estar a ser descarregada e quando o utilizador aceitar os termos e condições, a nova página é apresentada quase instantaneamente. Esta metodologia pode ser usada neste caso, pois é muito provável que o utilizador passe à página seguinte. Existem outros casos em que esta prática pode ser aplicada, como por exemplo, quando uma página consome muitos recursos e demora muito a ser carregada. No entanto, estes casos nem sempre se verificam e por vezes a rede disponível ao utilizador pode ser limitada, havendo a necessidade de utilizar a técnica com ponderação e se o seu uso for justificado.

O pré-carregamento [15] tem uma funcionalidade semelhante à metodologia previamente descrita, com a diferença de atuar apenas em determinadas partes de uma página (em vez da página toda), como imagens ou scripts que não estão a ser utilizados, mas

que poderão ser dentro em breve. O tamanho dos sites atualmente aumenta significativamente devido à abundância de imagens e scripts enormes, logo, se for possível obter a vantagem de descarregá-los antes de serem necessários, verificar-se-á uma melhoria nas respostas do site e na abertura de novas páginas.

### 3.4 Tree Shaking

A metodologia Tree Shaking [17] é usada para eliminar o código que a aplicação não emprega. Quando se importam e exportam módulos em JavaScript, na maior parte dos casos existe uma quantidade significativa de código que não é utilizado. Ao eliminar este código garante-se que a aplicação não vai ter módulos supérfluos na sua compilação. Assim, reduz-se significativamente o tamanho da mesma.

De forma semelhante, também podem ser removidos “selectors” de CSS que não sejam utilizados, o que é bastante comum em grandes aplicações.

### 3.5 Lazy Loading

Com o critério Lazy Loading [18] é possível identificar imagens ou listas e ordená-las por importância. Assim, o browser consegue reconhecer quais são as mais importantes e carregá-las primeiro. É mais eficiente em páginas que contêm muitas imagens. A sua correta aplicação permite poupar tempo e consumo de memória, uma vez que a transferência de conteúdo é otimizada.

### 3.6 Minificação

A minificação corresponde ao processo de remoção de elementos desnecessários e de reescrita de código para reduzir o tamanho de um ficheiro [19].

É frequentemente usada em ficheiros HTML, CSS e JavaScript. Ao reduzir o tamanho dos ficheiros, estes são transferidos mais rapidamente e, naturalmente as páginas são carregadas mais rapidamente. Há várias formas de minificar informação, sendo as mais básicas, a remoção de comentários, espaços desnecessários e quebras de linha. Os desenvolvedores devem ter o código indentado e devidamente comentado para cumprir as regras de boa programação, no entanto para o navegador essas linhas e caracteres são ignorados. Ao utilizar esta metodologia, todas essas regras vão ser desconsideradas e podem até ser alterados os nomes sugestivos das variáveis para um único carácter. Em ambiente de desenvolvimento, o resultado da aplicação desta metodologia poderia ser desastroso pois o desenvolvedor podia deixar de conseguir compreender convenientemente o código que produziu, sendo ainda pior para outros desenvolvedores que não tinham participado na produção desse código e que teriam que mais tarde analisar e alterar esse código. Assim, a minificação deverá acontecer somente em ambiente de produção, onde os ficheiros são separados do ambiente de desenvolvimento. Isto torna a alteração do código mais segura e permite aos desenvolvedores manter as boas práticas de programação e compreensão do seu trabalho.

## 4 Resultados

Como o trabalho incide na performance Web, o seu foco incidiu na otimização das suas páginas e gestão dos seus recursos, de forma a obter uma melhor resposta dos seus conteúdos. Inicialmente, submeteu-se a aplicação a testes de performance para promover a descoberta de problemas.

Depois de analisado o estado da aplicação em questão, verificou-se uma má gestão dos recursos em certas páginas da mesma. Desta forma, foram aplicadas as técnicas acima descritas com o objetivo de melhorar a performance da aplicação, tendo se seguido diminuir os seus tempos de resposta.

**Tabela 1.** Resultados dos tempos de abertura de algumas páginas.

Páginas	Resultado inicial	Resultado final
Página inicial - Index	2.08	1.31
CallTicketBar	0.854	0.081
AppointmentFilters	3.8	0.854
AppointmentSearchAdvanced	0.803	0.012
PartialCockpit	3.21	1.06
ExtendedHeader	1.77	1.05
Edição Ficha Doente - Index	6.18	2.02
PartialBilling	2.63	0.163

Na Tabela 1 encontra-se informação sobre algumas páginas da aplicação que foram submetidas às técnicas estudadas e, como se pode verificar, os problemas foram sendo resolvidos e atenuados. Todas as páginas registaram melhorias nos seus tempos de abertura. O resultado final da aplicação com as diversas técnicas aplicadas, apresentava sinais positivos de melhoria, que foram confirmados com uma nova sessão de testes.

## 5 Conclusões

Deste estudo pode concluir-se que os segundos retirados à abertura de cada página influenciam diretamente a performance da aplicação, fazendo com que o utilizador obtenha respostas mais rápidas. Como referido anteriormente, o atraso na resposta de uma página traduz-se numa perda de interesse e redução de satisfação do utilizador.

Quando se obteve acesso ao código da aplicação aqui tratada, logo se verificou que carecia de cuidados a nível de desempenho, tanto no seu frontend como no seu backend. Como o trabalho incidiu na performance Web, o foco foi a otimização das suas páginas e gestão dos seus recursos, de forma a obter uma melhor resposta dos seus conteúdos. Inicialmente, submeteu-se a aplicação a testes de performance para promover a descoberta de problemas. Uma vez encontrados, e como já havia conhecimento de técnicas a aplicar, os problemas foram sendo resolvidos e atenuados. O resultado final da aplicação com as diversas técnicas aplicadas apresentou sinais positivos de melhoria que foram confirmados com uma nova sessão de testes.

Para o autor, o desenvolvimento deste projeto revelou-se muito importante para o crescimento pessoal e profissional do mesmo. Espera-se que com o conhecimento aqui adquirido seja possível transformar mais aplicações e obter mais resultados positivos. É da intenção do autor continuar a investigar sobre este assunto e procurar sempre novas formas de enfrentar os impasses e solucioná-los de forma exemplar.

## Referências Bibliográficas

- [1] “Why Performance Matters | Web Fundamentals | Google Developers.” [Online]. Available: <https://developers.google.com/web/fundamentals/performance/why-performance-matters>. [Accessed: 05-Nov-2019].
- [2] “15 Web Performance Stats You Should Know.” [Online]. Available: <https://cdnify.com/blog/15-web-performance-stats/>. [Accessed: 11-Feb-2020].
- [3] “Why Performance Matters | Web Fundamentals | Google Developers.” [Online]. Available: [https://developers.google.com/web/fundamentals/performance/why-performance-matters#performance\\_is\\_about\\_retaining\\_users](https://developers.google.com/web/fundamentals/performance/why-performance-matters#performance_is_about_retaining_users). [Accessed: 13-Jan-2020].
- [4] “Driving user growth with performance improvements - Pinterest Engineering Blog - Medium.” [Online]. Available: <https://medium.com/pinterest-engineering/driving-user-growth-with-performance-improvements-cfc50dafadd7>. [Accessed: 13-Jan-2020].
- [5] “How the BBC builds websites that scale | Creative Bloq.” [Online]. Available: <https://www.creativebloq.com/features/how-the-bbc-builds-websites-that-scale>. [Accessed: 13-Jan-2020].
- [6] “How One Second Could Cost Amazon \$1.6 Billion In Sales.” [Online]. Available: <https://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales>. [Accessed: 14-Jan-2020].
- [7] “Why Google is the Best Search Engine & Why You Should Care | Tower.” [Online]. Available: <https://www.towermarketing.net/blog/google-best-search-engine/>. [Accessed: 14-Jan-2020].
- [8] “Basic Performance Tips.” [Online]. Available: [https://developer.apple.com/library/archive/documentation/Performance/Conceptual/PerformanceOverview/BasicTips/BasicTips.html#apple\\_ref/doc/uid/TP40001410-CH204-BCIFDFAA](https://developer.apple.com/library/archive/documentation/Performance/Conceptual/PerformanceOverview/BasicTips/BasicTips.html#apple_ref/doc/uid/TP40001410-CH204-BCIFDFAA). [Accessed: 20-Jan-2020].
- [9] “An Introduction to Perceived Performance | Treehouse Blog.” [Online]. Available: <https://blog.teamtreehouse.com/perceived-performance>. [Accessed: 20-Jan-2020].
- [10] “Why Performance Matters, Part 2: Perception Management — Smashing Magazine.” [Online]. Available: <https://www.smashingmagazine.com/2015/11/why-performance-matters-part-2-perception-management/>. [Accessed: 06-Nov-2019].
- [11] “Why Performance Matters, Part 2: Perception Management — Smashing Magazine.” [Online]. Available: <https://www.smashingmagazine.com/2015/11/why-performance-matters-part-2-perception-management/#preemptive-start>. [Accessed: 06-Nov-2019].
- [12] “Why Performance Matters, Part 2: Perception Management — Smashing Magazine.” [Online]. Available: <https://www.smashingmagazine.com/2015/11/why-performance-matters-part-2-perception-management/#early-completion>. [Accessed: 06-Nov-2019].
- [13] “GTmetrix | Website Speed and Performance Optimization.” [Online]. Available: <https://gtmetrix.com/>. [Accessed: 23-Feb-2020].
- [14] “PageSpeed Insights.” [Online]. Available: <https://developers.google.com/speed/pagespeed/insights/?hl=pt-PT>. [Accessed: 23-Feb-2020].
- [15] “Prefetching/Preloading and prerendering content with HTML.” [Online]. Available:

- <https://jack.ofspades.com/prefetching-preloading-and-prerendering-content-with-html/>. [Accessed: 23-Jan-2020].
- [16] “Preload, Prefetch And Priorities in Chrome - reloading - Medium.” [Online]. Available: <https://medium.com/reloading/preload-prefetch-and-priorities-in-chrome-776165961bbf>. [Accessed: 23-Jun-2020].
- [17] “Tree Shaking | webpack.” [Online]. Available: <https://webpack.js.org/guides/tree-shaking/>. [Accessed: 23-Feb-2020].
- [18] “Lazy Loading Images and Video | Web Fundamentals | Google Developers.” [Online]. Available: <https://developers.google.com/web/fundamentals/performance/lazy-loading-guidance/images-and-video>. [Accessed: 23-Feb-2020].
- [19] “The Illusion of Speed - Paul Bakaus’ blog.” [Online]. Available: <https://paulbakaus.com/tutorials/performance/the-illusion-of-speed/>. [Accessed: 23-Jan-2020].