

Sistema Iris - Análise e desenvolvimento de um sistema para análise e monitoramento de *downtime* de páginas web

Weslei Alvim de Tarso Marinho¹, José Augusto de Lima Prestes¹, José Geremonte Garcia¹,
Rafael Lucas Silva¹

¹ Facti - Fundação de Apoio à Capacitação em Tecnologia da Informação
Campinas, SP – Brasil
{weslei.marinho, jose.prestes, jose.garcia, rafael.silva}@facti.com.br

RESUMO

Falhas e *downtime* em *software* e sistemas, junto à exploração de vulnerabilidades de segurança, trazem grandes prejuízos para as organizações e empreendedores que comercializam os seus produtos e serviços por meio do comércio eletrônico. Como resultado de uma atividade interna de Pesquisa, Desenvolvimento e Inovação (PD&I), obteve-se o Sistema Iris, *software* para a análise e monitoramento de páginas web, detectando *downtime*, emitindo alertas e possibilitando a análise de páginas web com indicadores associados a temas como segurança, desempenho, otimização para motores de busca e responsividade da página web em dispositivos móveis. Além disto, o sistema também permite a realização de uma análise comparativa entre duas páginas web, abrangendo diversos elementos e aspectos. Os resultados obtidos foram então validados junto a potenciais usuários, demonstrando a utilidade do sistema e possibilitando melhorias em sua interface com o usuário.

Palavras-chave: sistema de monitoramento, *software*, monitoramento e análise, *website*.

Agradecimentos: Agradecimentos à Facti - Fundação de Apoio à Capacitação em Tecnologia da Informação, que apoiou a execução deste trabalho interno de Pesquisa, Desenvolvimento e Inovação (PD&I), e a outros colegas e ex-colegas de trabalho que contribuíram para o desenvolvimento e validação do sistema desenvolvido, em especial, André Feitoza Roque, Ana Paula Oliveira Ribeiro, Janaina Lemos Rocha, Karina Manoel Furiatto, Vera Ueda Pimental e Wellington Silva.

1 INTRODUÇÃO

De acordo com um recente levantamento, foi estimado que falhas em *software* causaram perdas financeiras na ordem de 1,7 trilhões de reais e um total acumulado de *downtime* de 268 anos em 2017 (Matteson, 2018). Em outro estudo, Gagnaire (2012) apresenta que serviços em nuvem apresentaram, em um estudo preliminar, em média, 7.738 horas de indisponibilidade por ano, impactando inúmeras aplicações que utilizam estes serviços.

Outro aspecto relevante e também relacionado com *downtime* são falhas de segurança em *software*, especialmente em aplicações web. Em um relatório publicado pelo Reino Unido, foi informado que houve um aumento no uso não autorizado de computadores, redes ou servidores por estranhos (UK, 2018).

Diante do cenário em questão, este trabalho apresenta o resultado de um trabalho de PD&I da Unidade de Monitoramento e Análise em Inteligência de Dados da Facti (UMA/Facti) de um sistema de *software* web para o monitoramento de *downtime* e análise de segurança de *websites*, um esforço que auxilia no combate ao *downtime* e na melhoria da segurança de páginas web.

O sistema permite, além da detecção e alertas de quando um determinado website está off-line, a geração de relatórios sobre a segurança, usabilidade, desempenho e outras características da página analisada. Também é permitido ao usuário realizar uma análise de *benchmark* dos parâmetros analisados com eventuais concorrentes

Os resultados obtidos foram validados junto a 16 potenciais usuários e as respostas obtidas validaram a utilidade do sistema e possibilitaram melhorias na interface com o usuário.

Este trabalho se encontra organizado nesta seção introdutória, pesquisas relacionadas identificadas pelos autores, a metodologia utilizada, os resultados obtidos com a validação junto aos potenciais usuários e as considerações finais.

2 PESQUISAS RELACIONADAS

Em uma busca por pesquisas relacionadas foram identificadas três iniciativas associadas ao monitoramento de *downtime*.

A escolha do monitoramento de *downtime* se deu essencialmente por conta de observações empíricas da grande evolução no consumo por meio de comércio eletrônico e diversas notícias abordando os montantes de perdas financeiras associadas. Verificou-se, então, que o acompanhamento de *uptime/downtime* de maneira eficaz e que permitisse às empresas e empreendedores rapidamente tomar ciência da condição e tomar as medidas necessárias para o restabelecimento do serviço poderia se transformar em um diferencial competitivo.

Com este objetivo, nas pesquisas realizadas foi encontrada uma patente para análise de *downtime* de servidores (Young, 2016). Esta patente trata de um dispositivo de hardware que realiza o monitoramento da energia provida e análises dos sinais de desligamento.

Por seu turno, Ramon e Derek (2017) apresentam uma aplicação que utiliza o protocolo ping para análise de *downtime* de servidores. Esta aplicação não é acessível via um navegador e se limita à análise dos resultados obtidos pela aplicação do protocolo ping.

Finalmente, Poper (2008) apresenta uma ferramenta para monitoramento de servidores com base no código HTTP de resposta de requisições realizadas a estes servidores.

Foram realizadas, também, pesquisas por eventuais serviços ofertados com propósitos similares, o que, na ocasião, não se encontrou.

3 PESQUISA, DESENVOLVIMENTO & INOVAÇÃO

Com base no problema de detecção de *downtime* e análise de aspectos de segurança da página web, foi realizada uma atividade de PD&I para identificar ferramentas similares existentes. Ferramentas que são invasivas (como, por exemplo, aquelas que necessitam ser instaladas no servidor onde a página web está instalada ou que necessita de um usuário e senha do servidor) foram desconsideradas.

Podem ser encontrados alguns sistemas internacionais e apenas um serviço nacional focado na detecção de *downtime*, não realizando análises de aspectos de segurança na página web monitorada - corroborando os resultados das buscas iniciais.

Sendo assim, definiu-se que o sistema a ser desenvolvido deveria ser o mínimo intrusivo possível; detectar e avisar o usuário quando seu website estivesse fora do ar; e fornecer relatórios sobre a qualidade da página (envolvendo aspectos de segurança, usabilidade e desempenho, avaliando e comparando o website analisado ao final do processo).

Foi feita então uma seleção de plataformas tecnológicas que serviriam como base para determinar que linguagem de desenvolvimento seria a mais adequada. Na sequência, montou-se uma tabela de avaliação para as opções Java, Javascript, PHP e Python, considerando-se aspectos como a familiaridade da equipe, recursos disponíveis e produtividade. As tabelas apresentadas a seguir contêm a análise realizada:

Tabela 1 – Tabela comparativa de linguagens para desenvolvimento (Fonte: elaboração própria)

| Identificação | | Popularidade | | | | Familiaridade da Equipe | | | | |
|---------------|---|--------------|---------------|---------------|-----------|-------------------------|----|----|----|-----------|
| Nome | Website | # StackOverf | Ranking TIOBE | Ranking da Wi | Pontuação | JG | VU | WS | WM | Pontuação |
| Java | https://www.java.com/ | 1384671 | 1 | 1 | 3 | 0 | 1 | 1 | 3 | 1,3 |
| Javascript | https://nodejs.org/ | 1573737 | 8 | 2 | 3 | 1 | 1 | 0 | 3 | 1,3 |
| PHP | https://secure.php.net/ | 1175993 | 7 | 2 | 2 | 2 | 2 | 0 | 2 | 1,5 |
| Python | https://www.python.org/ | 911220 | 4 | 2 | 2 | 1 | 0 | 0 | 1 | 0,5 |

Tabela 2 – Tabela comparativa de linguagens tecnológicas com pontuação final (Fonte: elaboração própria)

| Ident | Recursos | | | | | Produtividade | | | Eficiência | | | Resultado |
|------------|------------|-----------|------------|---------|-----------|---------------|------------|-----------|------------|------------|-----------|-----------|
| | Curva de A | IDEs (Amb | Biblioteca | Documen | Pontuação | Configuraçã | Velocidade | Pontuação | Desempenh | Uso de Mer | Pontuação | |
| Java | 1 | 3 | 3 | 3 | 2,5 | 2,0 | 1,0 | 1,5 | 1,0 | 1,0 | 1,0 | 9,3 |
| Javascript | 3 | 2 | 2 | 2 | 2,3 | 1,0 | 3,0 | 2,0 | 3,0 | 2,0 | 2,5 | 11,0 |
| PHP | 3 | 2 | 2 | 3 | 2,5 | 2,0 | 2,0 | 2,0 | 2,0 | 2,0 | 2,0 | 10,0 |
| Python | 2 | 1 | 2 | 2 | 1,8 | 1,0 | 1,0 | 1,0 | 1,0 | 3,0 | 2,0 | 7,3 |

Com base nas notas obtidas, a linguagem Javascript foi adotada para a totalidade do desenvolvimento. Isto foi escolhido por se tratar de linguagem com baixa curva de aprendizagem, que possui grande quantidade de recursos disponíveis e apresenta bons resultados de desempenho. Além disto, os profissionais realizaram uma rodada interna de capacitação básica em Javascript, habilitando-os a trabalhar com esta tecnologia no desenvolvimento da plataforma.

Após a seleção da linguagem de programação, foi realizada uma escolha de pilha tecnológica que seria utilizada para Javascript. Levou-se em consideração o sistema gerenciador de banco de dados, o framework web, motor de *templates* e tecnologias para a interface.

As opções analisadas de pilha tecnológica e as tecnologias escolhidas (destacadas) são apresentadas na tabela a seguir:

Tabela 3 – Opções de pilha tecnológica, com opções selecionadas (Fonte: elaboração própria)

| Opções | Racional da escolha |
|-------------------------|--|
| Banco de Dados | |
| MariaDB/MySQL | O MongoDB é de fácil uso e possui curva de aprendizagem muito baixa. Além de bom desempenho ele também facilita e muito a evolução dos dados, visto que iremos começar com versões simplificadas dos dados, por exemplo, o cliente inicialmente terá apenas nome, e-mail e senha porém posteriormente serão adicionados CPF, CNPJ, Razão Social, etc. |
| PostgreSQL | |
| MongoDB | |
| WebApp Framework | |
| Express | Express é o framework MVC mais utilizado e base da chamada arquitetura MEAN (Mongo, Express, Angular e NodeJS) (http://nodeframework.com/#mvc e https://www.airpair.com/node.js/posts/nodejs-framework-comparison-express-koa-hapi) |
| Koa | |
| Hapi | |
| Template Engine | |
| Handlebars | Foi selecionado o EJS por facilitar a inserção de Javascript no HTML e facilidade de uso do mesmo. |
| EJS | |
| Jade/Pug | |
| Interface | |
| JQuery + Bootstrap | Embora os outros frameworks considerados possuem vantagens como modularização (Angular), otimização de manipulação de páginas (React e Vue), seus usos são bem mais complexos com curva de aprendizagem muito maior que JQuery e Bootstrap. |
| Angular | |
| React | |
| Vue.js | |

Foi selecionada então a pilha tecnológica composta por MongoDB para o sistema gerenciador de banco de dados, Express para framework web, EJS como motor de *template* e JQuery + Bootstrap para a interface com o usuário.

Por fim, definiram-se os componentes seriam utilizados para a realizar a análise das páginas web, sendo considerados o uso de código objeto, os tipos de integração e a licença, entre outros. A tabela comparativa utilizada é apresentada a seguir:

Tabela 4 – Componentes para análise de páginas web (Fonte: elaboração própria)

| | Nome | Uso de Código-fonte | Uso de Código-objeto | Integração com o sistema | Modificação do Fonte | WebSite | Link Licença | |
|--|--|---------------------|----------------------|--------------------------|----------------------|---|---|---|
| Ferramentas e Tecnologias para os testes | (Usaremos as marcadas em verde e em último caso as marcadas em laranja) | | | | | | | Considerações sobre a licença |
| Ferramentas que geram estatísticas sobre acesso a uma página | Lighthouse | Sim | Sim | Parcial (Uso de rel | Não | https://www.chromelighthouse.com/ | https://www.chromelighthouse.com/licenses/ | OK para uso comercial. Caso, futuramente, haja a necessidade de redistribuir o software, a Faci deverá alertar para o que diz o item 4 da licença contida em https://raw.githubusercontent.com/GoogleChrome/lighthouse/master/LICENSE . Ação: fazer referência ao produto e indicar tanto o |
| | WebPageTest | Sim | Sim | Parcial (Uso de rel | Não | http://www.webpagetest.org/ | https://www.webpagetest.org/licenses/ | |
| | JMeter | Sim | Sim | Parcial (Uso de rel | Não | https://meter.apache.org/ | https://www.apache.org/licenses/ | |
| | PageSpeed API | Não | NA (Chamada de API | Parcial (Uso de rel | Não | https://developers.google.com/speed/api/ | https://developers.google.com/speed/terms/ | |
| | PhantomJS | Sim | Sim | Parcial (Uso de rel | Não | http://www.ariya.at/ | https://github.com/ariya/phantomjs/blob/master/LICENSE | |
| | PhantomJS-UI | Sim | Sim | Parcial (Uso de rel | Não | http://www.ariya.at/ | https://github.com/ariya/phantomjs/blob/master/LICENSE | |
| | Speedline | Sim | Sim | Parcial (Uso de rel | Não | http://www.speedline.com.br/ | http://www.speedline.com.br/pt-br/termos-e-condicoes/ | |
| | Qualiumeter | Sim | Sim | Parcial (Uso de rel | Não | http://www.qualiumeter.com/ | http://www.qualiumeter.com/pt-br/termos-e-condicoes/ | |
| | Perfun.js | Sim | Sim | Parcial (Uso de rel | Não | http://www.perfun.js.com/ | http://www.perfun.js.com/pt-br/termos-e-condicoes/ | |
| | Appmetrics | Sim | Sim | Parcial (Uso de rel | Não | http://www.appmetrics.com/ | http://www.appmetrics.com/pt-br/termos-e-condicoes/ | |
| | Browser-perf | Sim | Sim | Parcial (Uso de rel | Não | http://www.browserperf.com/ | http://www.browserperf.com/pt-br/termos-e-condicoes/ | |
| | Pymetrics | Sim | Sim | Parcial (Uso de rel | Não | http://www.pymetrics.com/ | http://www.pymetrics.com/pt-br/termos-e-condicoes/ | |
| | stepspeed.io | Sim | Sim | Parcial (Uso de rel | Não | http://stepspeed.io/ | http://stepspeed.io/pt-br/termos-e-condicoes/ | OK para uso comercial. Pode-se apenas a manutenção dos avisos de copyright da licença. Ação: fazer referência ao produto e indicar tanto o site quanto a |

De acordo com o levantamento realizado, a ferramenta Lighthouse foi selecionada como componente de análise, com base principalmente na facilidade do uso e termos da licença de uso.

Finalmente, foi construída uma arquitetura básica para o sistema, a utilizar as tecnologias selecionadas e capaz de ser expandida com novos componentes de análise, conforme a necessidade. Esta arquitetura é apresentada na figura a seguir:

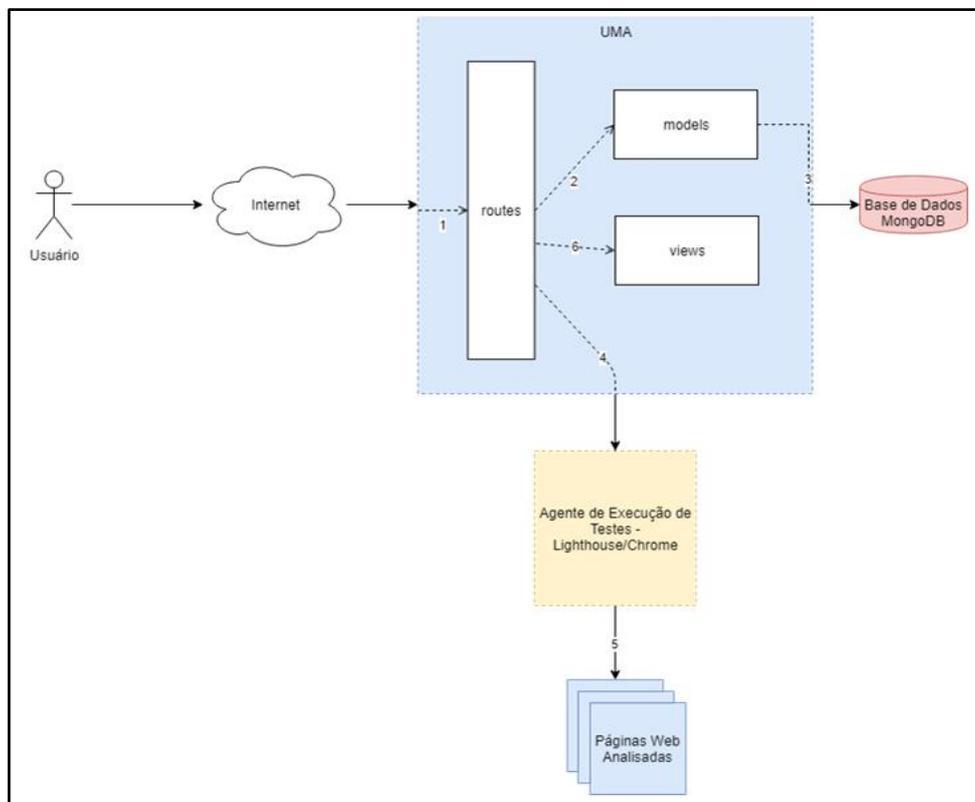


Figura 1 – Arquitetura base definida (Fonte: elaboração própria)

Com base nesta arquitetura, os usuários podem acessar o sistema pela Internet e o sistema se encontrará organizado em um conjunto de rotas, capazes de direcionar as requisições do usuário e tratar das regras de negócio aplicáveis.

Existe, ainda, um conjunto de modelos representando os dados persistidos e um conjunto de visões que representam a interface com o usuário. Este conjunto é responsável por fazer a comunicação com o sistema gerenciador de banco de dados utilizados e acessa uma ou mais instâncias do motor de análise - que pode eventualmente ser distribuído para que as análises possam, por exemplo, ser realizadas independentemente da localização do website ou implementações de outros motores para análises específicas.

Para comunicação entre as partes foi determinado o uso de serviços REST¹, que, por sua vez, utilizariam a notação JSON² para comunicação, facilitando a integração com outros sistemas.

4 RESULTADOS OBTIDOS

Com o fim do desenvolvimento, uma versão completamente funcional do sistema foi obtida, atendendo aos requisitos estabelecidos (isto é, monitoramento e alerta de *downtime* e análise de páginas web).

Foi dado o nome de *Iris* para o sistema, fazendo uma referência ao monitoramento e análise, por meio do olhar. O sistema passou por uma validação por um conjunto de potenciais usuários, que são apresentados nas subseções a seguir.

4.1 O sistema desenvolvido

A tela inicial do sistema desenvolvido é apresentada a seguir:

¹ <https://pt.wikipedia.org/wiki/REST>

² <https://www.json.org/json-pt.html>



Figura 2 – Tela inicial do sistema Iris (Fonte: elaboração própria)

Após o *login*, o usuário pode então ver quantos websites estão sendo monitorados e quantos estão online no momento, conforme apresentado a seguir:

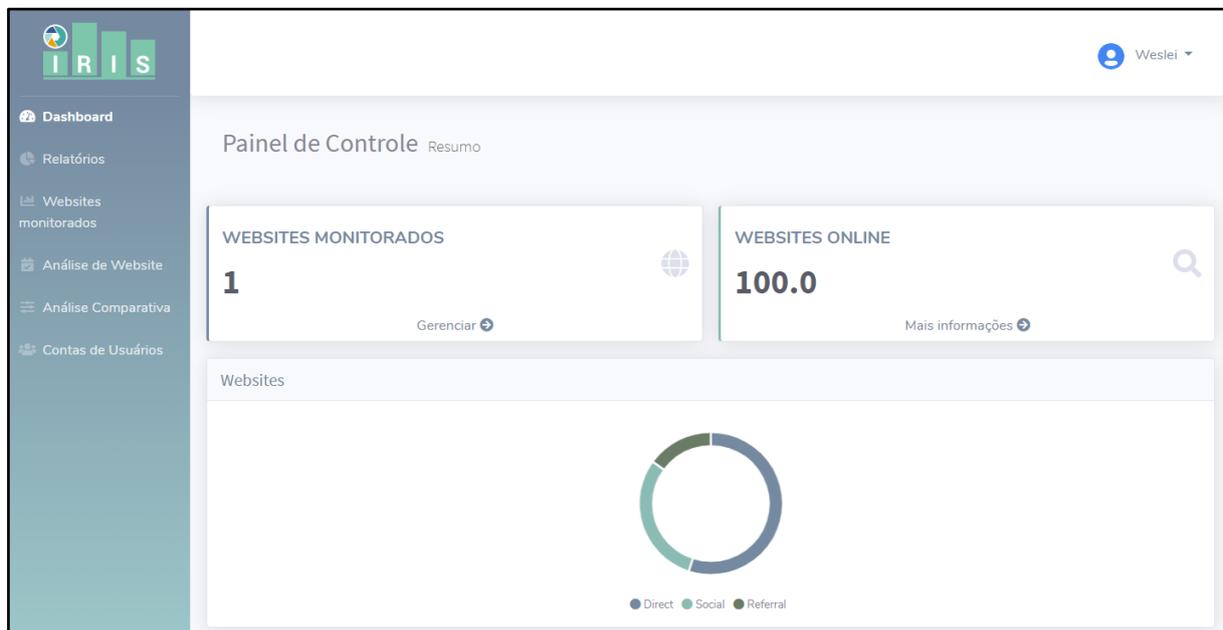


Figura 3 – Tela de entrada do painel administrativo (Fonte: elaboração própria)

Os relatórios gerados pelo sistema apresentam vários indicadores relacionados a desempenho, responsividade (inclusive, uma visão de como a página é apresentada em um navegador por meio de um dispositivo móvel), otimização para motores de busca (Search Engine Optimization - SEO) e segurança (analisando, por exemplo, a existência de bibliotecas Javascript

desatualizadas ou com vulnerabilidades de segurança na página analisada). A página inicial de um relatório de análise é apresentada a seguir:



Figura 4 – Primeira página de relatório de análise (Fonte: elaboração própria)

Também foi desenvolvido um relatório comparativo, por meio do qual os indicadores presentes no relatório de análise podem ser comparados aos resultantes da análise de uma outra página, fornecendo dados relevantes para a tomada de decisão para melhoria da página. As primeiras páginas deste relatório são apresentadas a seguir:



Figura 5 – Primeira página de relatório comparativo (Fonte: elaboração própria)

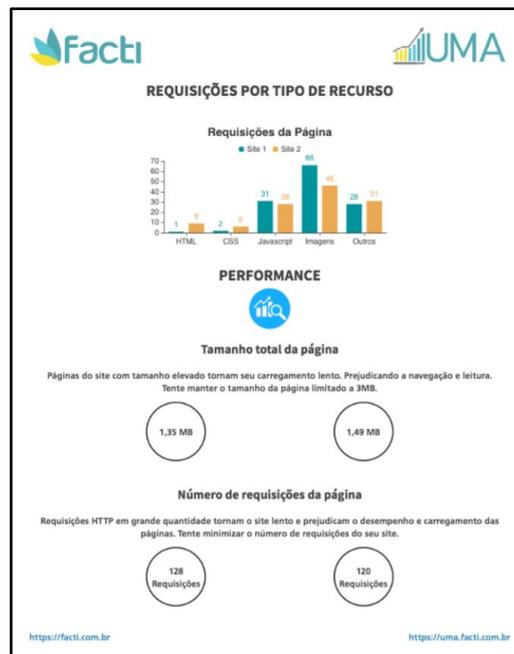


Figura 6 – Segunda página de relatório de análise (Fonte: elaboração própria)

Durante o monitoramento de *downtime*, caso o endereço web monitorado não esteja disponível, um e-mail é enviado ao usuário comunicando o problema, para que as medidas corretivas necessárias sejam por ele tomadas.

4.2 Validação dos resultados

Para validação do sistema, além dos testes com o usuário realizados, foi elaborado um questionário contendo questões como:

- Qual a primeira impressão obtida do sistema?
- Qual a natureza do serviço oferecido pelo sistema?
- Qual o público alvo do serviço?
- Você indicaria este serviço a alguém?

O questionário foi então disponibilizado para 16 participantes em uma avaliação piloto, organizados em dois grupos de oito participantes cada.

Um dos grupos teve uma apresentação prévia do sistema e o outro grupo realizou uma avaliação cega, sem prévio conhecimento sobre do que se tratava o sistema.

Foram obtidas nove respostas. Os resultados de algumas das questões do questionário são apresentados nas imagens a seguir:

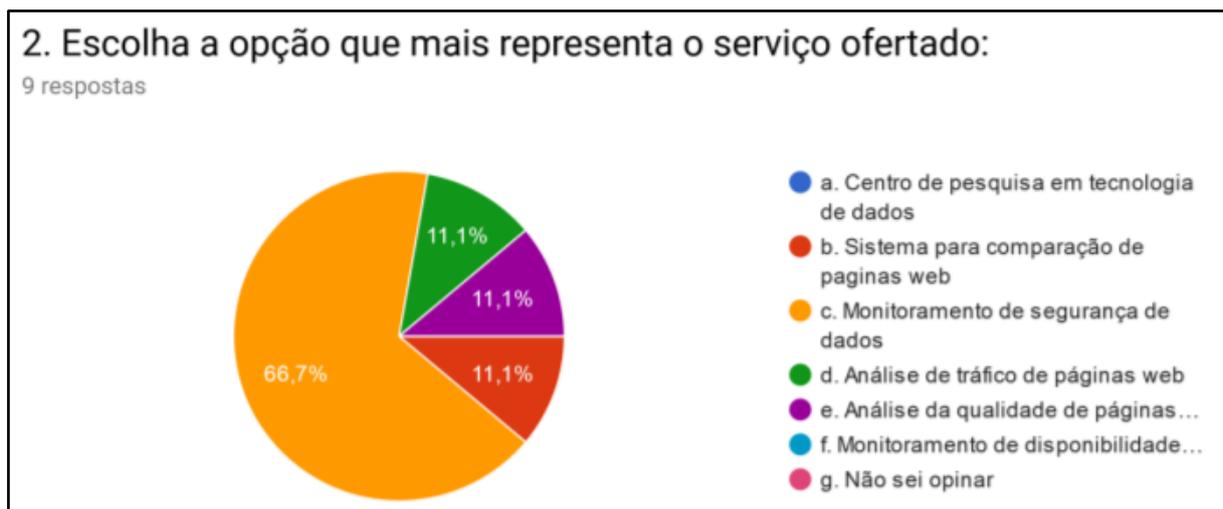


Figura 7 – Resultados da pergunta 2 do questionário (Fonte: Ana Paula Oliveira)

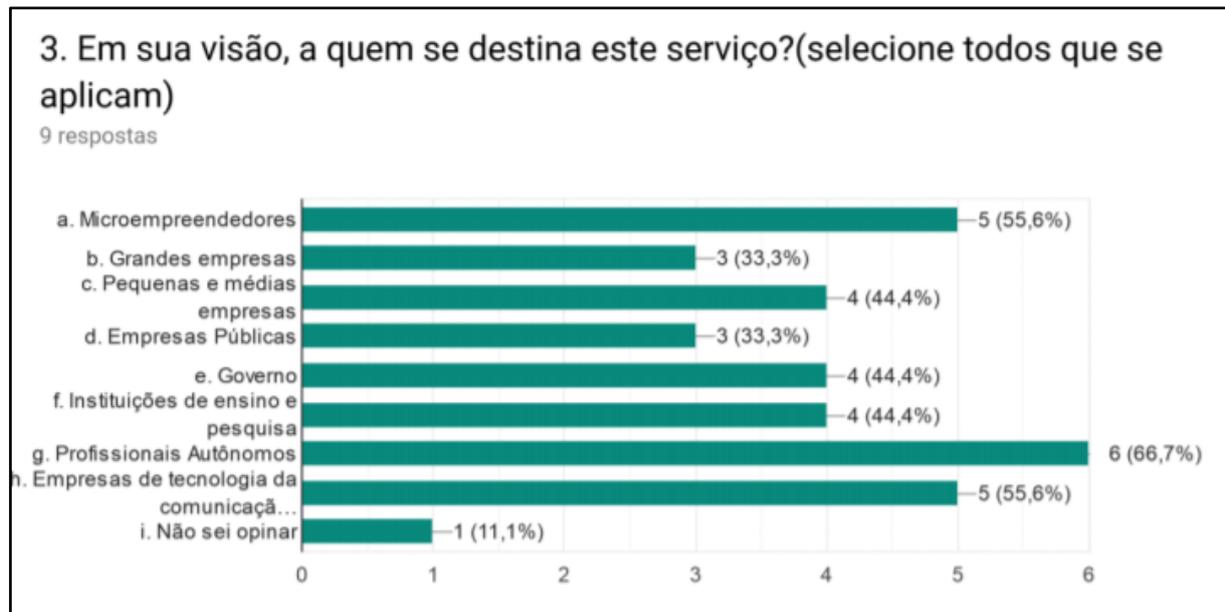


Figura 8 - Resultados da pergunta 3 do questionário (Fonte: Ana Paula Oliveira)

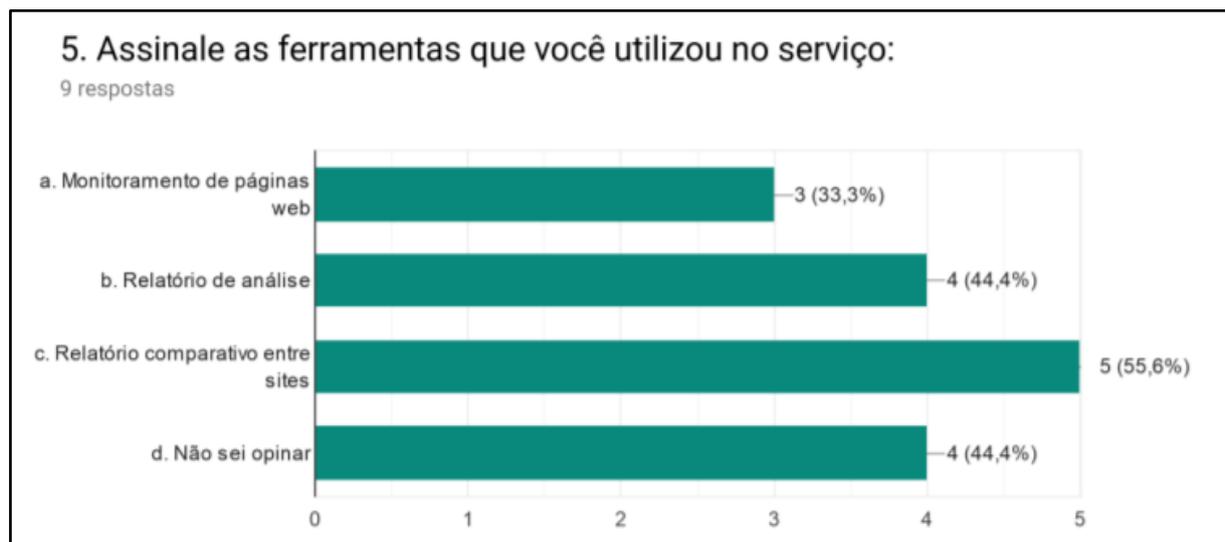


Figura 9 - Resultados da pergunta 5 do questionário (Fonte: Ana Paula Oliveira)

Esta validação permitiu identificar que o sistema atendia aquilo a que se propôs, também possibilitando o desenvolvimento de ações para a melhoria da interface com o usuário do sistema.

Diversos ajustes foram então realizados, com base nesta validação efetuada, para assegurar a melhoria de textos, readequação de layout e identidade visual. Estas ações permitiram a obtenção do resultado final apresentado na subseção anterior.

5 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo apresentar o resultado de PD&I de um sistema para o monitoramento de *downtime* e análise de páginas web.

Foi desenvolvido um sistema robusto que apresenta, além da funcionalidade de monitoramento e alerta sobre o *downtime*, uma análise compreensiva abordando indicadores como o desempenho da página, otimização para motores de busca, responsividade e segurança.

Também foi desenvolvida uma análise comparativa por meio da qual o usuário pode comparar os resultados de avaliação de uma determinada página com os resultados da avaliação de outra página, permitindo a identificação de pontos que precisam ser endereçados.

Os resultados também foram validados junto a potenciais usuários, cujo atendimento à necessidade proposta foi então avaliada por meio da aplicação de um questionário com 9 respondentes.

Os resultados foram satisfatórios, corroborando a utilidade e aderência do sistema ao objetivo proposto e permitindo que melhorias na interface com o usuário fossem realizadas.

Neste momento, o sistema se encontra sendo finalizado para oferta ao uso público, abrangendo pequenos ajustes na implementação de novos indicadores, e sob avaliação para novos contextos.

6 REFERÊNCIAS BIBLIOGRÁFICAS

Gagnaire, Maurice, et al. "Downtime statistics of current cloud solutions." *International Working Group on Cloud Computing Resiliency*, Tech. Rep (2012).

Matteson, Scott. "Report: Software failure caused \$1.7 trillion in financial losses in 2017." *TechRepublic*. [Online]. Disponível em: <https://www.techrepublic.com/article/report-software-failure-caused-1-7-trillion-in-financial-losses-in-2017> (2018).

Popa, Sorin. "WEB Server monitoring." *Annals of University of Craiova-Economic Sciences Series* 2.36 (2008): 710-715.

Ramon, Jerried, and Derek Williams. "Downtime Tracker." *iCascade* 1.1 (2017).

UK, GOV. "Cyber security breaches survey 2018." (2018).

Young, Erik Levon, and Andrew Brown. "Server downtime metering." U.S. Patent Application No. 14/916,295.