

Construção do Fakenewsbook: um portal para exibição de notícias falsas identificadas pelo FakeAnalyzer

Guilherme L. Urcelino, André C. da Silva

Grupo de Pesquisa Mobilidade e Novas Tecnologias de Interação
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP)
Campus Hortolândia – SP – Brasil

lima.u@aluno.ifsp.edu.br, andre.constantino@ifsp.edu.br

Abstract. *The FakeAnalyzer project emerged within the IFSP to create a chatbot that simulates human conversation within WhatsApp with the goal of checking whether a news item is fake or not. To show users that this chatbot is reliable, what information is collected, and enable users to interact with the database, an organized portal focused on the target audience (people from different technology domains) is needed. The goal of this project was to build the portal for the FakeAnalyzer project, the FakeNewsBook. It used the agile methodology, user stories, and the Bulma framework for its development. The differential is a graphic display of the number of occurrences of a fakenews per state.*

Resumo. *O projeto FakeAnalyzer surgiu no âmbito do IFSP para criação de um chatbot que simula conversa humana dentro do WhatsApp com objetivo de verificar se uma notícia é falsa ou não. Para mostrar aos usuários que esse chatbot é confiável, quais informações são coletadas e possibilitar aos usuários interagirem com a base de dados, é necessário um portal organizado, focado no público-alvo (pessoas de diferentes domínios de tecnologia). Assim, o objetivo deste projeto foi a construção do portal para o projeto FakeAnalyzer; o FakeNewsBook. Para o seu desenvolvimento, foram empregadas ritos da metodologia ágil, histórias de usuário e o uso de framework Bulma. Destaca-se que o diferencial é exibir de forma gráfica o número de ocorrências de uma notícia falsa por estado.*

1. Introdução

Fake news é um termo traduzido da língua inglesa que, em tradução livre, significa notícia falsa. Este termo consiste na disseminação de informação falsa através de meios de comunicação como, por exemplo, redes sociais (SILVA, 2021). Com a grande quantidade de pessoas utilizando plataformas de redes sociais como o WhatsApp que, segundo a mantenedora desta rede social são mais de dois bilhões de pessoas em mais de 180 países (WHATSAPP, 2021), a plataforma tornou-se um lugar fácil de se espalhar notícias, muitas delas, falsas. Dentro dessa plataforma, apenas com um pequeno número de toques na tela do celular, é possível selecionar uma mensagem, solicitar o encaminhamento, escolher o(s) destinatário(s) e solicitar o envio. Tentando minimizar o impacto do envio de uma mensagem a diversos usuários (técnica conhecida como realização de SPAM), a plataforma limitou, durante o encaminhamento de uma mensagem, a seleção de um grande número de destinatários. Embora essa restrição auxilia a diminuir o fator “viral” de uma mensagem, as notícias falsas ainda continuam a propagar em larga escala.

Para verificar se as notícias são verdadeiras, a equipe do WhatsApp juntamente com a Aliança Internacional de Checagem de Fatos IFCN (*International Fact-Checking Network*)

criaram um projeto que utiliza robôs, tecnologia denominada *chatbot*, que simulam conversas humanas (Figura 1) para responder perguntas em uma conta do WhatsApp (FABRO, 2020). Porém o projeto é voltado para validar apenas questões sobre COVID-19.

Um *chatbot* é um software que responde as mensagens do usuário, no caso deste *chatbot*, o usuário ao mandar uma mensagem, o *chatbot* responde com uma mensagem de boas-vindas e com um menu de opções (Figura 1a). O usuário ao responder com a mensagem “1”, o *chatbot* solicita ao usuário para enviar uma palavra ou uma frase curta relacionado ao COVID-19 que tenha dúvidas. Por exemplo, o usuário pode enviar a palavra “vitamina C” e o *chatbot* responde com a seguinte informação: “Classificado como Falso: *A vitamina C e a água morna com limão podem prevenir o COVID-19.* Esta informação foi verificada por **Agência Lupa** no dia 03/04/2020. *A vitamina C e a água morna com limão não têm efeito conhecido no novo coronavírus.* <https://piaui.folha.uol.com.br/lupa/2020/03/04/verificamos-vitamina-coronavirus/> (*_Portuguese_*)” (Figura 1b).

Durante os meses de junho de 2020, o professor André Constantino da Silva, orientador deste trabalho, vislumbrou um projeto similar com o intuito de verificar se uma informação é verdadeira ou falsa através de um *chatbot* do WhatsApp. O usuário encaminha uma mensagem contendo uma notícia (ou imagem) e o *chatbot* verifica em uma base de dados se há algum registro da informação dizendo se é verdadeira ou falsa (Figura 2).

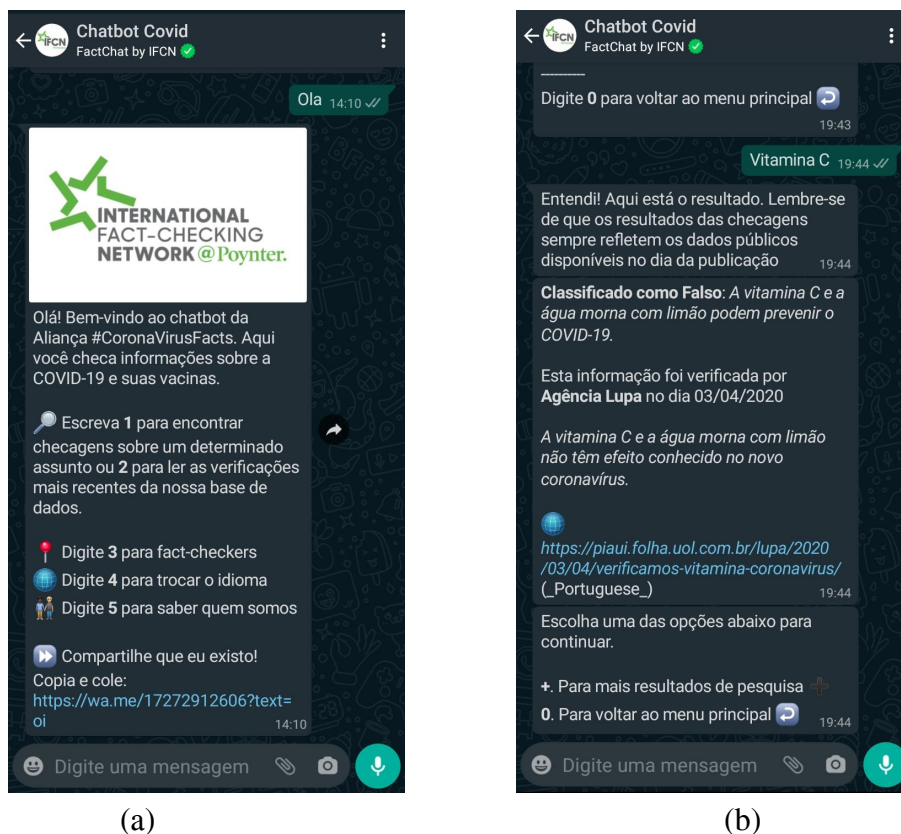


Figura 1. Interação com o *chatbot* da *International Fact-checking Network* (a) ao receber uma mensagem de saudação do usuário e (b) ao usuário escolher a opção 3 e enviar algumas palavras-chaves.

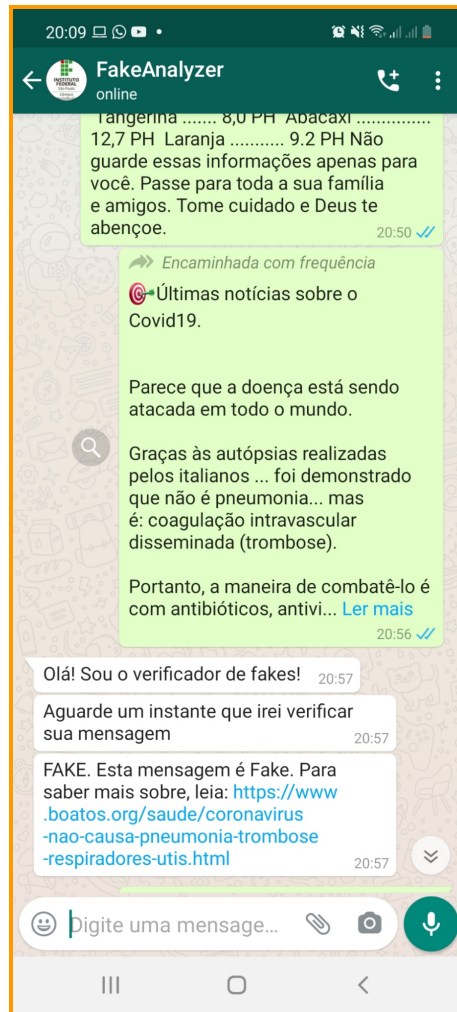


Figura 2. Interação com o chatbot FakeAnalyzer

O *chatbot* coleta qual foi a notícia solicitada de averiguação e armazena, além das notícias, dados não sensíveis do usuário, como o DDD de seu número (o qual é possível identificar qual o estado do país que o usuário se encontra) e a data de solicitação da averiguação. Conversas encerradas, ou seja, após o *bot* responder a análise, são apagadas do histórico de conversa. Com isso é possível: (i) criar uma base de dados aberta com *fake news* e período de circulação de uma notícia e (ii) realizar cruzamento de informações como qual notícia circulou em um determinado período de tempo, em quais regiões e em qual intensidade.

Para registrar se uma notícia é *fake news* ou notícia verdade, pretende-se criar uma comunidade virtual de pessoas que analisam as mensagens. Um dos critérios para se tornar membro é realizar um minicurso sobre checagem de notícias pela Web. Outros critérios ainda serão definidos.

Para que os usuários do *chatbot* possam ter acesso a informações coletadas por ele, fruto de um projeto dentro do IFSP, quais informações são coletadas e possibilitar aos usuários interajam com a base de dados, é necessário um portal organizado, focado no público-alvo (pessoas de diferentes domínios de tecnologia). Assim, o objetivo deste projeto é a construção do portal para o projeto FakeAnalyzer.

Na Seção 2 são apresentados os conceitos de Web, site e desenvolvimento Web, bem como a metodologia ágil. Na Seção 3, é apresentada a metodologia adotada para realização deste trabalho. Na Seção 4, é apresentado o desenvolvimento deste trabalho no qual cada subseção é uma iteração do processo pois foi adotado um desenvolvimento incremental. Encerra-se este artigo, apresentando as conclusões na Seção 5.

2. Fundamentação Teórica

2.1. Desenvolvimento web

A palavra web traduzida do inglês significa rede, e o desenvolvimento web engloba tudo que trabalhe na criação de páginas ou sistemas que trafegam pela rede mundial de computadores, a internet. O desenvolvimento web é o processo de criação, construção e manutenção de sites e aplicações para a internet ou intranet (Portalpos, 2021). Esta tarefa é executada com a codificação de linguagens que são interpretadas por um software chamado navegador que é responsável por entender os comandos realizados pelas linguagens e mostrar na tela para o usuário.

2.2. Site e Aplicações Web

Um site é composto por um link referente a um conjunto de páginas que estão disponíveis em alguma máquina física, essas páginas podem conter qualquer conteúdo que é capaz de ser interpretado pelo navegador, como códigos html, arquivos de mídia e entre outros. (Oxford Languages). Geralmente um site é composto por um código HTML contendo elementos de texto e de interação com o usuário como botões e elementos clicáveis que realizam ações através de linguagens de programação como o JavaScript (Linguagem que o navegador é capaz de codificar). Existe também a opção de utilizar linguagens de programação dentro da máquina que hospeda o site, assim elas não são interpretadas pelo navegador, elas fazem alterações no documento HTML por exemplo e o navegador carrega novamente esse arquivo podendo assim ver a alteração que foi feita.

O primeiro site (Figura 3) disponibilizado na internet história foi o <http://info.cern.ch/> em 06 de agosto de 1991 a CERN (Organização Europeia para a Pesquisa Nuclear) disponibilizou uma página que continha informações sobre a rede mundial de computadores.



Figura 3. Captura de tela do primeiro site disponibilizado na internet que conta sobre a história do W3.

Este site foi construído utilizando uma tecnologia chamada HTML (Linguagem de marcação de hipertexto), vemos que no site na primeira versão do html temos elementos como textos e links clicáveis que nos encaminham para outra página. Atualmente o HTML está na sua versão 5 e conta com elementos como botões, imagens, suporte a áudio, vídeo e entre outros elementos.

A web é composta por sites que estão disponibilizados dentro de servidores. Segundo Paula Filho (2003), Aplicações Web são “produtos de software ou sistemas de informática que utilizam uma arquitetura distribuída, pelo menos parcialmente sob protocolo http. Em consequência, pelo menos parte das interfaces com o usuário é acessível através de um navegador (*browser*)”.

Em termos de requisitos, destaca-se que uma aplicação web difere de uma aplicação *desktop* pelo fato de que a web pode ser vista e acessada por usuários em diversos dispositivos contendo o tamanho da tela diferente. Com isso temos a premissa de construir um site que possa ter seu tamanho ajustado sem perder seu conteúdo, podendo alterar ou diminuir o tamanho dos seus elementos para isso.

Para acessar sites na internet os usuários utilizam um programa de computador chamado navegador. Então na construção de um site deve ser levado em consideração qual navegador será utilizado e construir códigos de programação que possam ser executados pelos principais navegadores atuais.

Lembrando que um site para se manter em execução por um longo período de tempo necessita de manutenção. Pois pode ser que alguma tecnologia utilizada nele fique obsoleta e seja substituída por outra; também há atualizações constantes feitas pela equipe mantenedora do navegador podendo impactar em algo que já foi criado, então devemos utilizar linguagens de programação e ferramentas que estão bem aceitas e estáveis entre a comunidade de programadores.

Visto que os navegadores de internet se atualizam e a cada dia temos mais dispositivos novos o processo de desenvolvimento de um software para a internet necessita de um modelo de engenharia mais ágil do que os outros tipos de software. BEDER (2021) Sendo assim o processo de desenvolvimento web é feito por entregas periódicas chamadas de incremento, com isso entregamos pedaços pronto e usuais do nosso sistema e temos também *feedback* do cliente após cada entrega, não apenas no final do processo de criação do sistema como em modelos de engenharia não ágeis. Caso o cliente necessite de uma mudança urgente em seu software, com este processo é mais fácil a inserção de novos requisitos, pois será apenas um novo incremento a ser criado.

2.3. Coleta de requisitos: Histórias de Usuário

Uma história de usuário é uma descrição curta, informal e em linguagem simples do que um usuário quer fazer dentro de um produto de software (Digité, 2021). Rehkopf (2021) propõe que as histórias sejam elaboradas considerando um formato dividido em 3 partes: "Como [persona], eu [quero], [para que]". Assim o cliente pode informar para o desenvolvedor o que ele deseja que seja criado no sistema de uma forma não técnica. Exemplo de uma história de usuário em um sistema bancário fictício: Como cliente, gostaria de sacar dinheiro em caixas eletrônicos, para evitar filas dentro da agência do banco.

Segundo *Red Hat* (2021), o termo metodologia ágil surgiu em 2001 onde um grupo de desenvolvedores analisou problemas com o modelo mais atual utilizado no momento que era o cascata, assim este grupo analisou os principais pontos que era a falta de feedback do cliente, muita documentação para descrever o projeto e o plano era sempre fixo sem propostas para mudanças. Assim construíram um modelo de trabalho com menos documentação, entregas mais rápidas e menores e passaram a utilizar mais ferramentas do que muita documentação. Uma ferramenta bem utilizada por este modelo de desenvolvimento é o kanban, um quadro que são inseridos cartões contendo a denominação de todas as tarefas que devem ser feitas para realizar a construção do projeto. Este quadro é separado em 3 partes, a fazer, fazendo e feito e as tarefas transitam entre as 3 partes, assim o gerente de projetos tem o controle de tudo que está sendo feito. Outra técnica utilizada são as *daily* Reuniões curtas e diárias em que cada integrante do grupo informa para seu chefe ou gerente de projetos qual a última tarefa feita, qual será a próxima e se houve algum impedimento para a criação das tarefas.

2.4. Framework Google Charts e Bulma

Além do texto, também será necessário o emprego de gráficos. Assim, além de empregar as linguagens inerentes para o *front-end* de um sistema Web, como HTML, JavaScript, CSS, para desenhar a parte gráfica do site utilizaremos uma ferramenta disponível pelo Google chamada *Google Charts* (2021). O *Google Charts* disponibiliza diversos tipos de gráficos diferentes para desenvolvedores utilizarem de forma gratuita em seu site, e os dados para serem exibidos em um tipo de gráfico são fornecidos no formato JSON.

Por exemplo, para mostrar dados em relação a regiões geográficas do Brasil é possível utilizar o gráfico que representa o mapa geográfico, indicando qual região a notícia falsa apareceu (Figura 4).

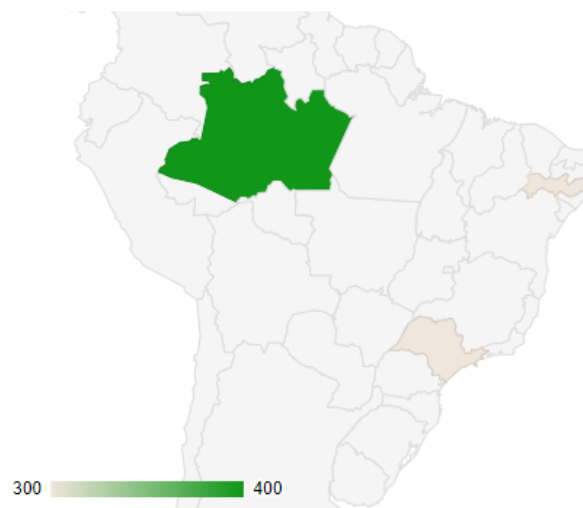


Figura 4. Exemplo de uso do Google Charts destacando os estados brasileiros Amazonas, São Paulo e Pernambuco com valores aleatórios.

Com o avanço da tecnologia hoje em dia temos diversos dispositivos capazes de acessar a internet, por isso temos a necessidade de criar um site responsivo. Sites responsivos são aqueles que adaptam o tamanho das suas páginas (alteração do layout) ao tamanho das telas que estão sendo exibidos, como as telas de celulares e tablets, segundo SEBRAE (2019) e Marcotte (2011).

Para criar a estrutura responsiva do site utilizaremos um framework de CSS chamado Bulma (2020). O Bulma é uma estrutura de código aberto e gratuito que fornece todo suporte para criar design contendo responsividade e traz cores intensas ao seu site como podemos ver na Figura 5.



Figura 5. Exemplo código botões bulma css

Fonte: <https://bulma.io/documentation/elements/button/#states>, 2021

3. Metodologia

Por se tratar de uma colaboração de projetos, tendo em vista que os dois estão fazendo de construção (*FakeAnalyzer e FakenewsBook*), partimos para a elaboração das histórias de usuário visando identificar os requisitos do portal. Em seguida, elaboramos um protótipo que foi refinado em reuniões com o orientador, definindo um design para a interface de usuário. O design proposto foi implementado em três etapas: construção da estrutura geral do portal, a implementação das visualizações gráficas e, ao final do projeto, a inserção de conteúdo sobre o projeto e contato. Para a realização de testes, foram usados dados coletados da base de dados fomentadas pelo *chatbot*.

4. Desenvolvimento

Considerando a metodologia delineada, apresentamos o detalhamento da execução de cada uma das atividades nas subseções a seguir.

4.1. Elaboração de histórias de usuário para o portal

As histórias de usuário foram levantadas em duas reuniões por meio de *brainstorming* (em tradução livre, tempestade de ideias) para o sistema, a primeira contendo as ideias principais levantadas pelo orientador do projeto que se passou como cliente do sistema levantando suas necessidades dentro da plataforma e resultou nas seguintes histórias:

1. Como contribuidor, quero ter acesso a quantidade de notícias que ajudei a identificar que são fake news;
2. Como contribuidor, quero poder informar se uma notícia é fake news;
3. Como contribuidor, quero saber o meu score (pontuação) de contribuidor;
4. Como administrador, quero poder receber pedidos de solicitação de cidadãos para serem contribuidores;
5. Como administrador, quero poder suspender a conta de um contribuidor;
6. Como cidadão, quero fazer uma pesquisa nas notícias da base de dados;

7. Como cidadão, quero ver as 10 fake news mais divulgadas na semana;
8. Como cidadão, quero poder ver notícias separadas por assuntos;
9. Como cidadão, quero saber os dados que o *chatbot* coleta;
10. Como pesquisador, quero ter acesso a base de dados das fake news coletadas;
11. Como cidadão, quero saber como colaborar na identificação de fake news;
12. Como cidadão, quero saber em quais estados uma fake news foi divulgada e a quantidade de transmissões identificadas.

A segunda reunião foi realizada com o intuito de observar novas funcionalidades necessárias dentro do sistema, porém as tais foram levantadas pelo aluno visando adicionar seu próprio pensamento na construção de histórias, pois o cliente não necessita ter todo conhecimento técnico para construção de um sistema, assim ele pode deixar passar algumas funcionalidades essenciais despercebidas. Foram elaboradas as seguintes histórias:

13. Como usuário cadastrado quero interagir com o site podendo avaliar as notícias em botões com emoji para representar diferentes estados emocionais como "gostei/não gostei/tô espantado/furioso/não acredito/uauuu" ou avaliação de "1 a 5";
14. Como contribuidor quero mostrar meu score (pontuação) embaixo das notícias contribuídas;
15. Como pesquisador quero sugerir uma alteração da notícia caso haja uma nova descoberta que invalide a notícia;
16. Como usuário desejo compartilhar que uma notícia é *fake* com um selo e o link para o portal do FakeAnalyzer;
17. Como cidadão quero um filtro para notícias como: notícias mais recentes, notícias mais visualizadas e entre outros filtros possíveis;
18. Como cidadão quero botão para compartilhar notícias em plataformas digitais como whatsapp, e facebook;
19. Como cidadão quero ter acesso ao número de visualizações de cada notícia dentro do site;
20. Como cidadão quero ter acesso a data de cada notícia;
21. Como cidadão quero uma indicação visual chamativa em notícias que são fake;
22. Como cidadão quero ler uma prévia da notícia quando estiver visualizando várias notícias;
23. Como usuário cadastrado quero poder fazer comentários embaixo de notícias;
24. Como cidadão quero um menu que me leve a todas as diferentes áreas do site;
25. Como cidadão, quero saber a quantidade de notícias registradas para serem analisadas por dia, mês e ano;
26. Como cidadão quero encontrar o portal em plataformas de pesquisa como Google e Bing facilmente;
27. Como cidadão, quero saber como contatar o chatbot pelo WhatsApp;
28. Como cidadão, quero saber os criadores do sistema (o 'sobre' do site);
29. Como cidadão, quero poder fazer sugestões para os desenvolvedores.

Ao todo foram elaboradas 29 histórias de usuário, todas sendo explicadas e aceitas pelos dois integrantes nas reuniões. Vale ressaltar que ao longo do desenvolvimento do projeto podem surgir novas histórias de usuários tendo o consentimento dos dois lados, o cliente e a equipe de desenvolvimento.

Para escrevermos as histórias em conjunto e de forma remota utilizamos uma plataforma chamada Trello. Este sistema nos fornece um quadro chamado Kanban e por meio deste levantamos as histórias de usuários conforme a Figura 6.



Figura 6. Sistema Trello contendo algumas das histórias de usuário para o portal

Com o levantamento das histórias de usuários também obtemos o conhecimento dos diferentes tipos de usuários que irão compor o sistema. Foram observados 5 níveis de usuários, estes são: Cidadão, Usuário Cadastrado, Contribuidor, Pesquisador e Administrador. O papel do cidadão seria um usuário comum que tem a opção de visualizar todo conteúdo do site, podendo ler as notícias e observar os dados das *fake news*. O usuário cadastrado além de visualizar e pode interagir com o site, podendo clicar em botões de reação como: gostei e não gostei e também pode realizar comentários em determinadas áreas do sistema. O contribuidor é responsável pelo cadastro das notícias dentro da plataforma. Este usuário deve ter uma atenção especial fora do sistema, necessitando um aprendizado prévio para poder se tornar contribuidor. Com isso surge a funcionalidade do usuário administrador que gerencia as contas de contribuidores podendo aceitar o início de uma nova conta e também de excluir a conta caso necessário. Por fim temos o pesquisador, uma conta mais técnica que tem acesso a todos os dados do sistema e pode então realizar uma alteração em alguma notícia caso haja necessidade.

4.2. Protótipo visual do portal

Para criar os protótipos de design deste projeto utilizamos *wireframes*. Podemos definir *wireframe* como um esqueleto, um protótipo ou uma versão bastante primitiva do visual de um projeto. Segundo Becker (2019), *wireframes* são como desenhos de rascunhos do design do sistema. Para criação dos mesmos foi usada a plataforma Miro. Com esta plataforma podemos desenhar elementos gráficos de sistemas e temos a opção de trabalho simultâneo remoto.

Os *wireframes* da primeira semana de trabalho estão expostos na Figura 7. A parte criativa do design foi elaborada analisando as histórias de usuário e inserindo as suas necessidades dentro das diferentes telas do sistema. com isso cada tela correspondeu as histórias da seguinte forma:

- Frame 1: História 12;
- Frame 2: História 25;
- Frame 3: Histórias 6 e 22;
- Frame 4: Histórias 13, 21, 18, 19, 20;
- Frame 5: História 7;
- Frame 6: História 23;
- Frame 7: História 24;
- Frame 8: História 24;
- Frame 9: História 2;
- Frame 10: História 3;
- Frame 11: História 1.

Histórias restantes que não entraram nos frames: 17, 29, 28, 27, 10, 11, 9, 8, 4, 5, 15, 14, 16 e 26.

Em reunião a primeira versão de protótipo foi avaliada pelo orientador com um respaldo positivo na qualidade técnica da criação porém deixou a desejar no quesito preencher todas as histórias de usuário. Observando a quantidade grande de histórias de usuário foi proposto começar a criação do site apenas para os usuários de visualização que são: Cidadão e Usuário Cadastrado, deixando para trabalhos futuros os usuários: Pesquisador, Administrador e Contribuidor.

Após esta etapa foi iniciado o trabalho de codificação do site passando o protótipo (Figura 7) representado por um desenho para um site real utilizando a linguagem de marcação HTML5 e CSS3 juntamente com o Bulma.css. O HTML é uma linguagem de marcação de texto criada inicialmente para redigir documentos, porém foi utilizada para construção de web sites em navegadores de internet. Em HTML é possível criar toda estrutura de conteúdo do site como títulos, textos e inserção de imagens. O CSS é uma linguagem capaz de formatar os documentos HTML inserindo cores, tamanhos e formatos, podendo também alterar a localização de cada elemento na tela. O Bulma é uma ferramenta de design que traz códigos em CSS3 prontos contendo elementos de design já estilizados como por exemplo botões com cantos arredondados e cores vivas. Utilizando essas 3 ferramentas e seguindo o padrão criado nos *wireframes* foi criada a primeira versão do protótipo fora do papel, conforme ilustrado na Figura 8. A utilização dos *wireframes* foi muito útil pois no momento da codificação foi necessário apenas escrever os códigos HTML e ajustar o design com CSS obedecendo os tamanhos e cores propostos, assim não teve necessidade de pensar como seria o design durante o momento da codificação, economizando esforço e tempo.

Esta primeira versão do site foi criada em um servidor Apache em um computador local funcionando somente com acesso dentro da mesma rede de conexão com a internet. O próximo passo foi armazenar o nosso código em um local seguro, evitando perda de trabalho caso o computador local sofra algum dano irreparável em seu armazenamento de dados, para isso a comunidade de desenvolvedores de software utilizam um sistema de versionamento de código chamado GIT com ele é possível fazer upload dos seus arquivos em um local armazenado dentro da internet, e é possível ter o controle de todas as versões do código pois ele oferece a possibilidade de salvar todas as versões criadas. Para utilização do sistema Git utilizamos o site GitHub, uma plataforma que fornece uma interface web para realizarmos o upload dos códigos. Assim pode-se acessar o código fonte deste projeto pelo link: <https://github.com/guilhermeUrcelino123/Portal-FakeAnalyzer>.

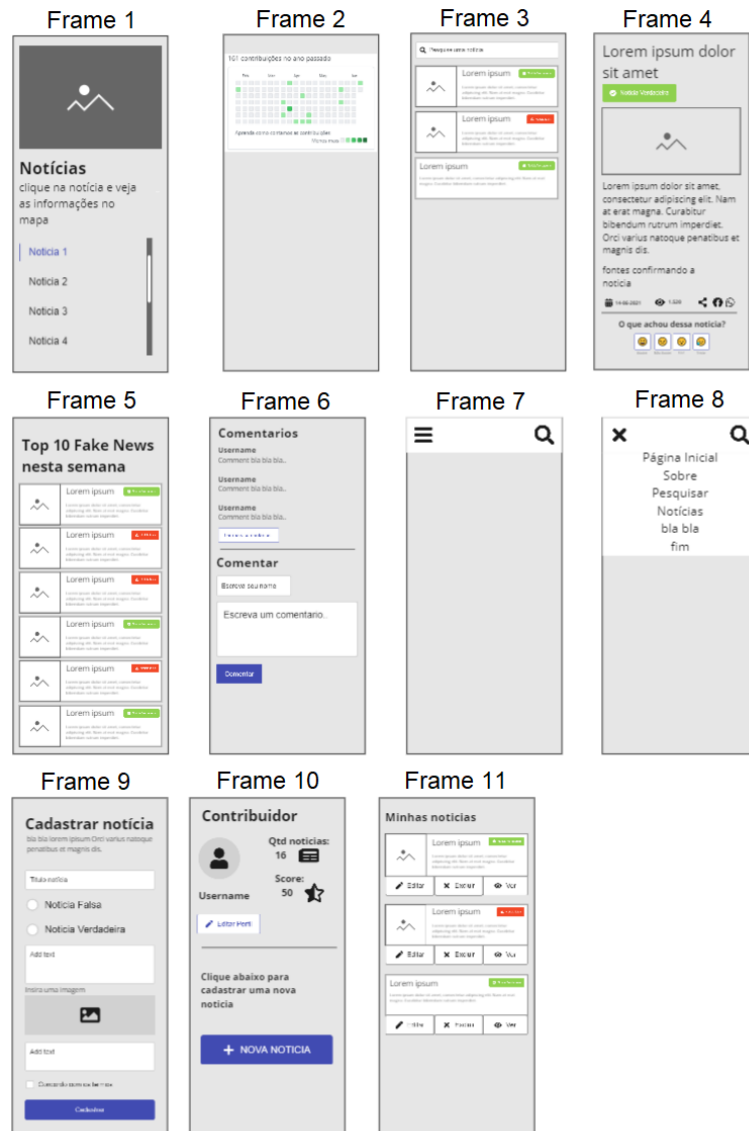
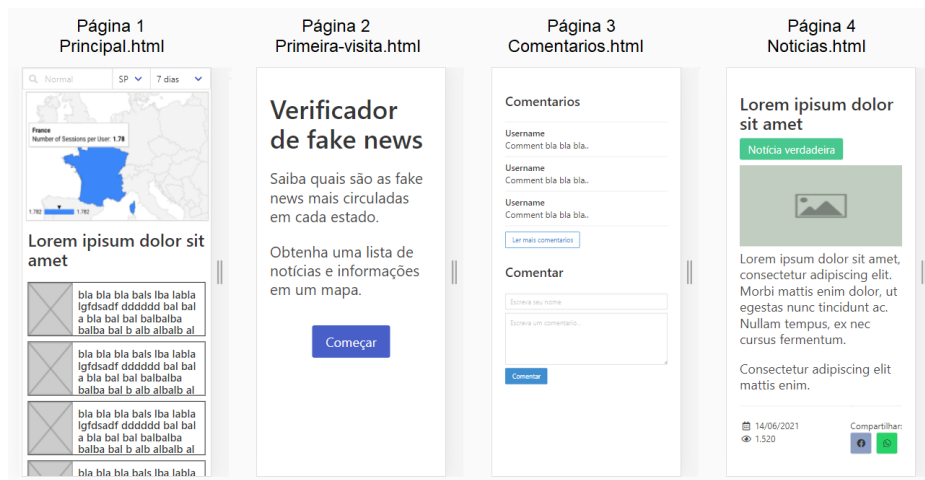


Figura 7. Wireframes contendo a primeira versão do design do sistema



(a) (b) (c) (d)

Figura 8. Design final das telas após discussões : (a) tela de apresentação de notícias, (b) tela de informação sobre o sistema, (c) tela de visualização de comentários e (d) tela de visualização de notícia.

4.3. Iteração 1: codificando a visualização de notícias e visualização de comentários

Na primeira iteração de codificação do protótipo foi decidido criar os frames relacionados às notícias da plataforma com isso utilizando apenas HTML, CSS e o Bulma.css foi criado os frames 1, 4 e 6 resultando nas páginas 1, 4 e 3 conforme ilustra a Figura 8. Também foi criado uma página inicial para que um usuário entre pela primeira vez e conheça um pouco do sistema, descrito na Figura 8b. Com isso atendemos as histórias 12, 13, 21, 18, 19, 20 e 23.

4.4. Iteração 2: Inserção de dados dinâmicos através de base de dados

Com o design das páginas de notícias prontos, nesta etapa foi adicionado uma base de dados para trabalhar com dados dinâmicos dentro das páginas, assim podemos utilizar *loops* repetindo elementos da nossa página contendo conteúdos salvos na base de dados como por exemplo uma lista com títulos e imagens de notícias. Com a inserção de dados reais o design sofreu leves alterações para conter os dados como, por exemplo, tamanhos de fonte e reposicionamento de elementos na tela.

As páginas também passaram a ter códigos de programação, para isso utilizamos a linguagem PHP. Foi criada uma lista de notícias contendo título e foto que, ao receber um clique, redireciona para página de notícia. Para isso utilizamos o método GET do protocolo HTTP, que consiste em passar um valor dentro da URL da página. Na Figura 9, há um exemplo da utilização do método GET do protocolo HTTP ao realizar uma pesquisa no buscador Google. Dentro da elipse vermelha na imagem vemos a URL: [google.com/search](https://google.com/search?q=teste) seguida pela instrução: “?q=teste” isso significa que a página de pesquisa do Google recebeu o valor “teste” via GET e atribuiu a “q”, com isso utilizou o valor de “q” para pesquisar em sua base de dados e retornou 164.000.000 resultados.

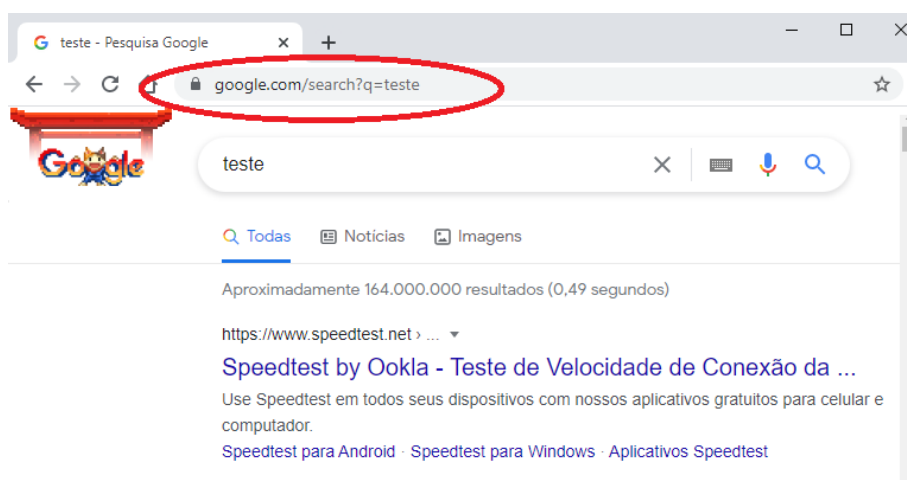


Figura 9. Printscreen da tela de resposta por uma busca no buscador Google demonstrando uso do método GET do protocolo HTTP.

Utilizando esta técnica passamos o número de identificação de uma notícia de uma página para a outra e assim realizamos uma pesquisa na base de dados retornando os dados da notícia relativa ao id recebido. Na base de dados foram adicionadas duas notícias aleatórias obtidas na aba de notícias apenas para testarmos nosso sistema, cujo resultado é mostrado na Figura 10.

As alterações foram feitas nas páginas 1 e 4 e a página 3 que é responsável pelos comentários foi adicionada na página 4 após o final da notícia, a página 2 não sofreu

alteração. O Quadro 1 apresenta parte de um código que exemplifica a aplicação do Bulma.css, mais especificamente, para exibição de uma notícia. Devido ao tamanho do arquivo, alguns trechos foram substituídos por reticências, assim focando o leitor para compreender o uso do Bulma.css .

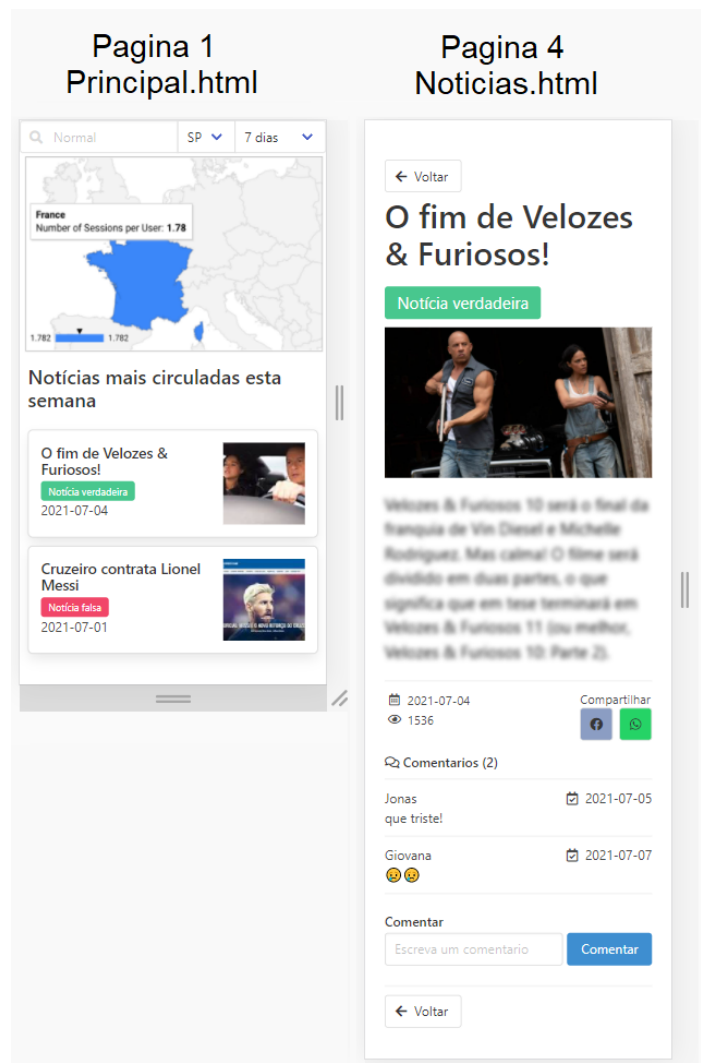


Figura 10. Páginas 1 e 4 atualizadas na segunda iteração, contendo dados em base de dados (texto borrado na imagem para não atrapalhar leitura do artigo).

A linha 7 do código no Quadro 1 indica ao navegador para utilizar o framework Bulma.css, carregando os arquivos.

4.5. Iteração 3: Inserção de dados dinâmicos através de base de dados.

Na iteração anterior foi cuidado apenas da parte visual do sistema, não contendo interações do usuário como, por exemplo, tratamento de cliques em botões como o botão comentar. O tratamento foi realizado na terceira iteração do desenvolvimento. Nesta iteração também foram identificados alguns requisitos não funcionais dentro do sistema. Segundo (SOMMERVILLE, 2011), requisitos funcionais são aqueles que o sistema, o sistema de software ou o componente de sistema deve ser capaz de executar

Quadro 1. Trecho de código criado para a visualização de notícias utilizando HTML5, CSS e o Bulma.

```
1 <?php //abre espaço de codificação em php
2 ...
3 ?>
4 <html>
5 <head>
6 ...
7 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.3/css/bulma.min.css">
8 <!--código do bulma css-->
9 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@5.15.3/css/all.min.css"/>
10 <!--código online contendo framework font awesome, framework usado para obter ícones-->
11 <style>
12 .box-new { /*classe para adicionar edições sobre o box bulma*/
13 display: flex; /* Altera o modo de organização dos elementos dentro do box */
14 justify-content: space-between; /*distribui os elementos de maneira uniforme colocando o primeiro no início e o segundo no fim*/
15 border: 1px solid #dbdbdb; /*código em hexadecimal de uma variação de cinza claro */
16 margin-bottom: 10px; /*espaço de 10 pixels na parte debaixo do box*/
17 padding: 10px; /*espaço de 10 pixels dentro do box */
18 }
19 .box-new h5{
20 margin-bottom: 0px; /*altera o tamanho da margem para 0px;*/
21 }
22 </style>
23 </head>
24 <body>
25 ...
26 <a href="noticia.php?id=?php echo $Id; ?>">
27 <!-- link para noticia.php informando o id da noticia via HTTP GET -->
28 <div class="box content box-new">
29 <!--utiliza classes box e content do bulma e a classe box-new criada acima na tag style -->
30 <article>
31 <!--tag simples html que informando início da escrita de algo-->
32 <h5><?php echo $Titulo; ?> </h5>
33 <!--utiliza tag de título 5 e lê o conteúdo da variável título em php-->
34 <?php
35 if ($Status == 0){ //verifica se variável status é 0 ou 1 como se fosse verdadeiro ou falso
36 echo "<span class='tag is-success '>Notícia verdadeira</span>";/*utiliza php para escrever tag html,
37 classe tag, class is success deixa o elemento com a cor verde*/
38 }else{
39 echo "<span class='tag is-danger '>Notícia falsa</span>";/*classe is danger deixa o elemento com a cor vermelha */
40 }
41 ?>
42 <br/><!--pula linha em html-->
43 <span>
44 <?php echo $Data; ?> <!--Escreve na tela a data contida da variável data-->
45 </span>
46 </article>
47 <!--Converte a imagem em base 64
48 para o formato necessário pelo html e a mostra na tela-->
49 </div>
50 </a>
51 </body>
52 </html>
```

Na iteração anterior foi criado um sistema de comentários embaixo das notícias, como podemos ver na Figura 11. Porém só foi criada a parte de leitura de dados, podendo apenas visualizar os comentários. Nesta etapa criamos a parte de interação do usuário para realizar o comentário. Para isso foi usado a linguagem PHP com o método POST. Semelhante ao método GET mencionado no tópico acima o 4.4 o POST também serve para fazer envio de dados, ele geralmente é utilizado para fazer envio de dados preenchidos por formulários na web, ele faz o envio de dados dentro do corpo da mensagem HTTP e os dados não ficam visíveis a nível de usuário como no GET e só podem ser captados por meio de programação.

Após o envio de dados pelo POST, obtemos os dados através de um script em linguagem PHP e inserimos dentro de uma base de dados MYSQL. Assim ao carregarmos a página teremos o comentário enviado pelo usuário, pois a página de comentário carrega todos os comentários realizados sobre a notícia visualizada na página.

Os requisitos funcionais criados foram manipulações de elementos HTML em tempo real para que o sistema de comentários se tornasse um pouco mais dinâmico e interativo com o usuário.

Um exemplo é o botão de ver todos os comentários. Inicialmente são mostrados apenas 3 comentários para não atrapalhar a leitura do texto com muitos comentários e também não tornar a página muito grande, porém caso o usuário queira ler todos os comentários sobre a notícia basta clicar no botão que os comentários irão aparecer instantaneamente na tela.

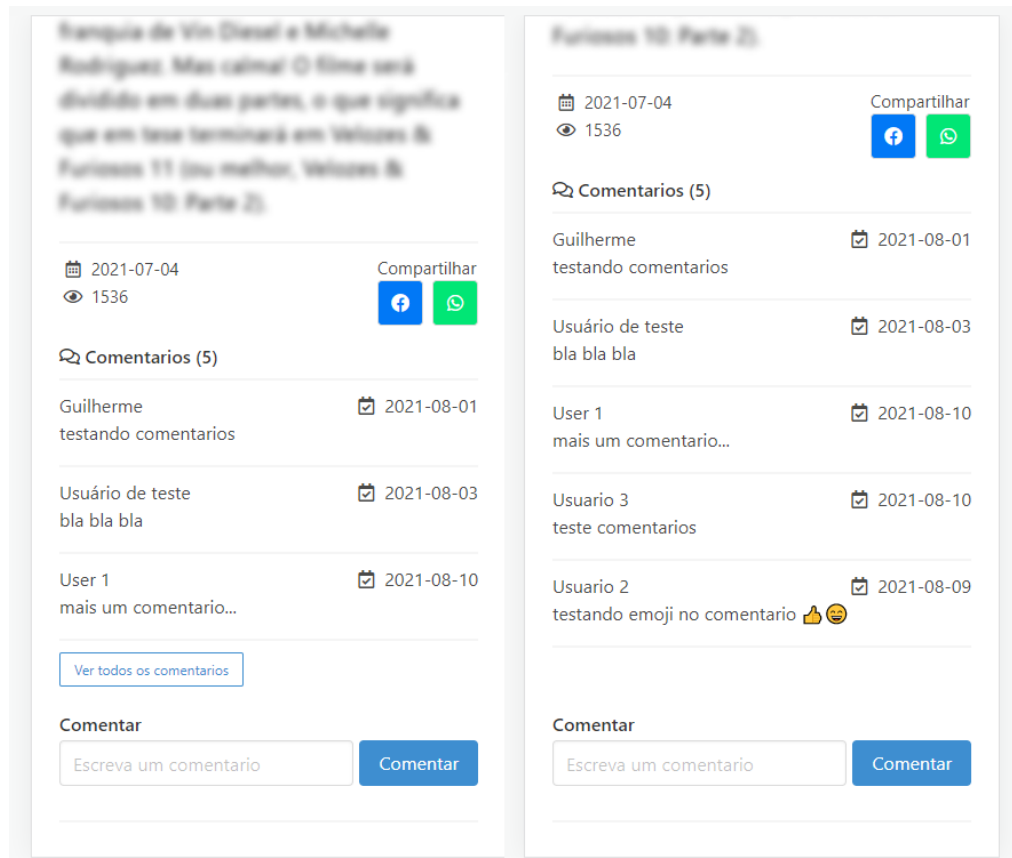


Figura 11. Exemplificação da função de ver todos os comentários.

Estes tipo de manipulação de HTML pode ser feito com o uso da linguagem JavaScript juntamente com CSS. O JavaScript contém funções capazes de obter qualquer elemento contido dentro do HTML, com isso podemos obter os elementos de comentários e modificarmos ele como, por exemplo, alterando o CSS de cada elemento. O comando utilizado foi o:

```
document.getElementsByClassName(".comentario");
```

O CSS é responsável por modificar o design dos elementos HTML, nesta linguagem temos um comando capaz de remover um elemento da tela o tornando invisível este comando é escrito desta forma:

```
display: none;
```

Sendo assim nosso HTML foi construído da seguinte forma conforme mostra imagem abaixo,

```
><div class="comentario" style="display:block; ">...</div>  
><div class="comentario" style="display:block; ">...</div>  
><div class="comentario" style="display:block; ">...</div>  
><div class="comentario" style="display:none; ">...</div>  
><div class="comentario" style="display:none; ">...</div>
```

Até o terceiro elemento adicionamos o `display: block`, este comando é o contrário do `display:none`, ele não promove nenhuma alteração no elemento HTML. E com auxílio da linguagem PHP, após o terceiro elemento adicionamos o `display: none`, assim estes elementos não aparecerão na tela.

Ao clicarmos no botão “ver todos os comentários” acionamos o código em JavaScript do Quadro 2 que obtém todos os elementos que contém a classe `comentario`, e utilizamos um laço de repetição para adicionarmos o `display:block` em todos os elementos capturados. Assim todos os comentários são mostrados instantaneamente na tela para o usuário.

Quadro 2. Função criada para exibir todos os comentários.

```
function mostrar_comentarios(){  
  const comentarios = document.querySelectorAll('.comentario');  
  comentarios.forEach(element => {  
    element.style.display = 'block';  
  });  
};
```

Também foi adicionada outra manipulação de elemento HTML como essa, a manipulação de inserir um novo comentário. Ao inserir um novo comentário o mesmo é adicionado na base de dados, e para aparecer na tela o usuário teria que carregar a página novamente para que a página possa ler todos os comentários (inclusive o comentário recém inserido). Para evitar recarregar a página, adicionamos este contato dinamicamente assim que o usuário clicar no botão comentar, assim o sistema de comentários se torna mais dinâmico.

Outro requisito não funcional inserido no projeto foi o de responsividade. Foi adicionado um container em volta da página toda. Assim a página pode ser vista em diversos dispositivos com um tamanho razoável para o usuário navegar no site. Para fazer isso foi utilizado o CSS, adicionamos um elemento HTML se referindo ao código CSS exposta no Quadro 3, em volta de toda página.

Quadro 3. Código CSS para inclusão de container em volta da página.

```
.container{  
  max-width:800px;  
  margin:0 auto;  
}
```

Assim definimos que o tamanho máximo da página será de 800 pixels e estará centralizada no meio da tela independente de seu tamanho, caso a tela for menor de 800 pixels o site irá preencher a tela toda, com isso o site funciona em dispositivos de todos os tamanhos.

4.6. Iteração 4: inserção do mapa mostrando quantidade de *fake news* em cada estado

Na iteração 4 foi decidido adicionar o mapa contendo a quantidade de *fake news* em cada estado perante a busca da página principal do sistema. Essa é uma das funcionalidades principais do sistema, então foi necessário uma atenção especial nesta etapa.

Para não construirmos códigos muito complexos do zero a comunidade de programação composta por empresas e profissionais da área disponibilizam códigos prontos para serem utilizados dentro de seu sistema. Esta ideia é mencionada como não reinventamos a roda.

A empresa Google disponibilizou um serviço chamado Google Charts. Este serviço disponibiliza diversos gráficos para serem utilizados dentro de páginas Web. Ao todo, até a data atual, são disponibilizados 29 gráficos diferentes para seu uso como: gráfico de barras, gráfico de pizza e entre outros.

Um dos gráficos disponibilizados é o *GeoCharts*, um gráfico que contém um mapa mundial que pode ser manipulado alterando cores e adicionando informações em diversas regiões. Este mapa tem o contexto exato de um requisito proposto por uma história de usuário deste sistema: a história de usuário 12 (Figura 12).

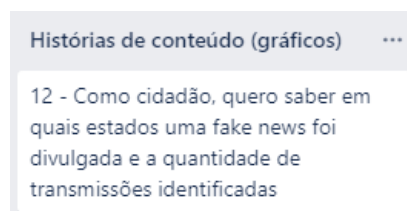


Figura 12. História de usuário 12 para o portal FakeAnalyzer

Assim decidimos utilizar este mapa para construir esta funcionalidade. Para utilizarmos o mapa necessitamos então obter o estado onde cada notícia foi divulgada. Com isso foi adicionado no banco de dados a tabela estado, contendo o nome de todos estados, e na tabela de notícia uma referência para a tabela estado para cada notícia poder obter a referência do estado onde ela foi divulgada.

Falando sobre a parte técnica em programação para utilizar o mapa foi liberado pela google no link: <https://developers.google.com/chart/interactive/docs/gallery/geochart#fullhtml> um código de exemplo do GeoCharts com fácil acesso a modificação para ser utilizado em seu website. Este código cria o seguinte mapa (mostrado na Figura 13), ao passar o ponteiro do mouse ou o dedo em dispositivo de toque na tela em cima de cada país o mapa informa um dado que foi escrito no código informado no link acima.

E o código referente aos dados é uma estrutura de dados chamada Array que é capaz de armazenar diversos dados que podem ser acessados depois por um número de índice. O código foi disponibilizado na linguagem JavaScript e tem sua escrita semelhante ao exposto no Quadro 4.

Com isso os dados foram disponibilizados dentro do mapa. Por ser uma estrutura de dados simples e de baixa complexidade de entendimento, ela pode ser modificada com facilidade por programadores, adicionando qualquer dado que deseja e modificando o mapa conforme sua necessidade.

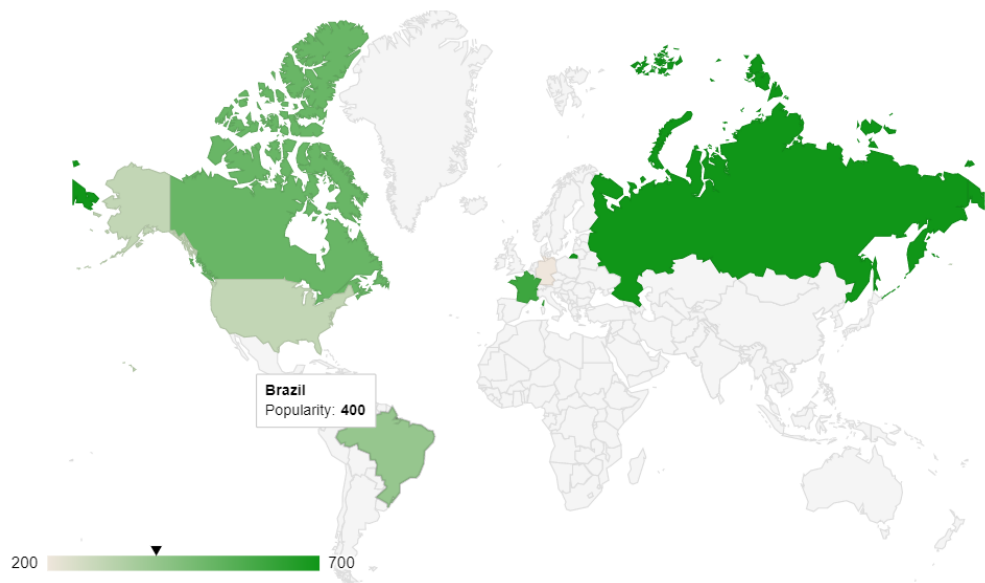


Figura 13. Exibição de dados considerando a distribuição geográfica por país

Quadro 4. Exemplo de código em formato JSON apresentando dados numéricos associados a países.

```
var data = google.visualization.arrayToDataTable([
  ['Country', 'Popularity'],
  ['Germany', 200],
  ['United States', 300],
  ['Brazil', 400],
  ['Canada', 500],
  ['France', 600],
  ['RU', 700]
]);
```

Assim, adicionamos um mapa à página principal do nosso sistema contendo as informações de quantas notícias estão circulando em cada estado e obtivemos o resultado exposto na Figura 14.



Notícias mais circuladas esta semana

Figura 14. Exibição de mapa com dados sobre fake news nas regiões do Brasil

4.7. Iteração 5: Inclusão da tela de login e do processo autenticação

O processo de login e autenticação foi pensado para usuários poderem interagir dentro do sistema, porém nem todos os usuários têm a intenção de realizar alguma interação então o sistema terá seus dois modos, modo de visualização em que o usuário não necessita logar dentro do sistema, apenas entrar no site e ver as notícias. e temos o modo de interação que necessita da criação de um usuário contendo nome e senha, assim este usuário pode salvar comentários e realizar interações dentro do sistema como clicar em botões interativos. Assim criamos uma página inicial onde temos a opção de logar ou entrar livremente dentro do sistema conforme relatamos na Figura 15.

O método de segurança utilizado no login foi um método simples mas eficaz para este projeto, será realizado a pesquisa dentro da base de dados verificando se a senha e o usuário estão corretas. A senha será salva utilizando um modelo de criptografia chamada MD5 onde temos uma função de entrada e saída em que inserimos um dado em formato de texto e ele retorna este dado criptografado. Assim, um desenvolvedor ao olhar para a base de dados não terá acesso a senha, apenas ao dado criptografado.

4.8. Responsividade da aplicação Web

A responsividade do projeto foi criada da seguinte forma, criamos um container centralizado na página, inserimos nossos elementos dentro deste container. Para deixarmos a página maleável em diversos tamanho de tela foi aplicada uma regra ao container, em dispositivos de até 300 pixels de resolução na vertical o container ocupava 100% do espaço sem sofrer alteração no seu tamanho. Acima de 300 pixels o container continuava em 100% da tela, porém os elementos da esquerda e da direita iam para suas extremidades deixando um espaço vazio no centro. Após a tela atingir mais de 800 pixels de resolução o container ficava com os 800 pixels centralizado no meio da tela independente de seu tamanho final. Assim a tela se torna legível dentro dos aparelhos mais usados no momento para entrar na internet que são tablets celulares notebooks e computadores de mesa.



Figura 15. Tela de login para o Fakenewsbook

5. Conclusões

Durante os meses de junho a agosto de 2020, o professor André Constantino da Silva vislumbrou um projeto similar com o intuito de verificar se uma informação é verdadeira ou falsa através de um *chatbot* do WhatsApp. Para mostrar aos usuários que esse *chatbot* é confiável, fruto de um projeto dentro do IFSP, quais informações são coletadas e possibilitar aos usuários interagirem com a base de dados, é necessário um portal organizado, focado no público-alvo. Assim, o objetivo deste projeto foi a construção da primeira versão do portal do projeto FakeAnalyzer. Para desenvolvimento desta versão, iniciou-se o processo com a identificação das funcionalidades por sessões de *brainstorming* seguidas de prototipação das telas da interface de usuário. Após a compreensão dos requisitos do portal, adotou-se uma abordagem incremental com 5 iterações para a construção do código. A cada incremento, o orientador do trabalho analisava e sugeria modificações, consideradas no incremento seguinte, até a finalização dos 5 incrementos que compõem a primeira versão do portal, atingindo o objetivo do projeto. Contribuímos de modo que as pessoas possam ter acesso a uma fonte de notícias e verificar se são verdadeiras ou falsas.

Este projeto possibilitou realizarmos uma colaboração entre dois projetos dentro do campus do IFSP (este trabalho e o projeto FakeAnalyzer), possibilitando expor os dados coletados pelo *chatbot*, bem como informando sobre o projeto, informações importantes para o público e, desta forma, contribuindo para o projeto FakeAnalyzer. Como trabalhos futuros, sugerimos disponibilizar o site em um servidor na internet, assim oferecendo ao usuário o uso real da aplicação. Sugere-se também a construção de uma nova versão inserindo as funcionalidades do primeiro escopo que foram removidas no segundo. E destacando a identificação de novas funcionalidades que envolvem apoiar a comunidade na análise dos dados coletados pelo *chatbot*.

Por questão de tempo decidimos diminuir o escopo (escopo inicial Figura 6 e escopo final Figura 7) e entregar o objetivo principal do projeto, que é visualizar as notícias de forma simples e poder interagir com o sistema inserindo comentários e visualizar dados como quantidade de visualizações e data de cada notícia. O projeto fica em aberto podendo receber interações como responder enquetes ou pesquisas e interações que são comuns entre redes sociais como por exemplo reagir a uma notícia com emojis e está em aberto para utilizar a criatividade.

Em questão técnica foi possível aprender a desenvolver um projeto de sistema para a internet dando foco no seu design e nas entregas incrementais que foram realizadas. Aprendemos princípios novos de modelagem de software como história de usuário e utilizamos o método ágil de forma simples e funcional utilizando o aluno como time de desenvolvimento e o orientador como cliente para avaliar as entregas. Assim, foram articulados conhecimentos assimilados nas disciplinas de Lógica de Programação, Arquitetura de Software, Desenvolvimento Web, Metodologias Ágeis, Interação Humano-Computador, Banco de Dados I, Metodologia de Pesquisa Científica, Inglês Técnico, Inglês Técnico Avançado e Comunicação e Expressão. Também adquiri novos conhecimentos como emprego do framework de apresentação Bulma, *GeoCharts* e responsividade.

Referências

BEDER, D. M. Uma abordagem sistemática: uma Abordagem Sistemática Para o Desenvolvimento de Aplicações web. EdUFSCar; 1ª edição (3 dezembro 2021).

- BECKER, L. “Wireframes, o que são e por que os utilizamos?”, Disponível em:<<https://www.organicadigital.com/blog/o-que-sao-wireframes-e-por-que-os-utilizamos/>>. Acesso em: 20 de julho . 2021.
- BULMA, “Bulma : o framework CSS moderno que simplesmente funciona”, 2020, Disponível em:<<https://bulma.io>>. Acesso em: 27 Maio. 2021.
- DIGITÉ, “histórias de Usuários: O Que São e Por Que e Como Utilizá-las“, 2021, Disponível em: <<https://www.digite.com/pt-br/agile/histórias-de-usuarios/>>. Acesso em: 16 de Junho. 2021.
- FABRO, C. O que é bot para WhatsApp e como funciona? TechTudo. Jun. 2020a. Disponível em:<<https://www.techtudo.com.br/listas/2020/06/bot-para-whatsapp-7-perguntas-erespostas-sobre-robos-no-app.ghml>>. Acesso em: 20 de julho . 2021.
- HEDHAT, “O que é metodologia Ágil?” 1, Disponível em: <<https://www.redhat.com/pt-br/devops/what-is-agile-methodology>> Acesso em: 31 de outubro. 2021.
- PAULA FILHO, W. de P. Engenharia de software: fundamentos, métodos e padrões. Rio de Janeiro: LTC, 2003.
- PORTALPOS, Por que desenvolvimento web é uma carreira promissora? Disponível em:<<https://blog.portalpos.com.br/desenvolvimento-web/>>. Acesso em: 20 de julho . 2021.
- SILVA, “Histórias de Usuários: O Que São e Por Que e Como Utilizá-las“, 2021, Disponível em: <<https://aisel.aisnet.org/isla2021/17/>>. Acesso em: 05 de outubro. 2021.
- SILVA JUNIOR, J. L. & GRAEML, A. R., "Fake News e seus Impactos nas Organizações" (2021). *ISLA 2021 Proceedings*. 17. Disponível em:< <https://aisel.aisnet.org/isla2021/17/>> . Acesso em: 05 de outubro. 2021
- SOMMERVILLE, I. Engenharia de Software. 10. ed. São Paulo: Pearson Education do Brasil, 2018.
- WHATSAPP, “Sobre o whatsapp” 2021, Disponível em: <<https://www.whatsapp.com/about?/>>. Acesso em: 05 de Junho. 2021.

Documento Digitalizado Público

TCC - Anexo I

Assunto: TCC - Anexo I
Assinado por: Andre Constantino
Tipo do Documento: Relatório Externo
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Documento Digital

Documento assinado eletronicamente por:

- **Andre Constantino da Silva, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 25/03/2023 19:16:22.

Este documento foi armazenado no SUAP em 25/03/2023. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1278195

Código de Autenticação: 7b028d0615

