Proposta e Implementação de uma Aplicação de Reconhecimento e Descrição de Imagens para Deficientes Visuais

Lucas Izidorio Almeida[1], **Rodrigo Gonçalves Ribeiro**[1]

[1]PUC Minas em Contagem

{izidorio.lucas7,roxdrigo1883}@gmail.com

Resumo. *Em um mundo cada vez mais tecnológico, a inclusão de pessoas deficientes visuais se tornou um tópico importante a ser discutido. A experiência do usuário deve ser pensada para que todos possam utilizar os sistemas de informação, principalmente os sistemas em que os usuários são criadores de conteúdo, como em redes sociais. No contexto de imagens geradas e publicadas pelos usuários do sistema, a geração de descrição do conteúdo das imagens se torna ainda mais desafiadora. Este trabalho tem como objetivo propor e implementar o Web-Assistente de Reconhecimento Digital de Imagens (WARD), uma extensão para browser capaz de identificar imagens na página Web e criar descrições acessíveis, para que os usuários deficientes visuais tenham acesso à informação sobre o conteúdo que é exibido pela imagem. A aplicação integra as plataformas de software como serviço Google Cloud Vision*[1] *para reconhecimento de imagens e Google Cloud Translate*[2] *para tradução de textos. Como resultados, apresenta-se a especificação, projeto e implementação da aplicação. Além disso, avaliações da aplicação mostram que ela é capaz aumentar a acessibilidade de redes sociais ao identificar imagens, obter descrições dessas imagens e traduzir as descrições com baixo impacto para o usuário em termos de resposta e de custo financeiro associado às plataformas de software como serviço.*

1. Introdução

A deficiência visual é uma característica que acomete milhões de brasileiros. Segundo as informações da última Pesquisa Nacional de Saúde (PNS) realizada pelo Instituto Brasileiro de Geografia e Estatística (IBGE), o Brasil possuía, em 2019, cerca de 3,4% da população com 2 anos ou mais de idade com muita dificuldade para enxergar ou ausência total da visão, o que representava quase 7 milhões de brasileiros [IBGE 2021]. Tais pessoas, entretanto, acabam enfrentando dificuldades ao utilizar os sistemas que estão disponíveis na Web, principalmente quando se trata de redes sociais. Apesar de existirem funcionalidades de acessibilidade em algumas redes sociais, a maior parte do conteúdo visual publicado é feito pelos usuários e fica sem uma descrição acessível. Tal fato acaba criando barreiras que dificultam a participação de pessoas com deficiência no uso das redes sociais [Gleason et al. 2020a].

Existem softwares de acessibilidade para sistemas operacionais que auxiliam no uso de computadores pelo público que possui alguma deficiência visual

[1]Vision AI - Disponível em: https://cloud.google.com/vision. Acesso em 20 abr. 2021.

[2]Cloud Translation - Disponível em: https://cloud.google.com/translate. Acesso em 20 abr. 2021.

[Mynatt and Weber 1994]. Dentre tais softwares, podemos citar ferramentas que fazem a leitura da tela para o usuário, como o *Non Visual Desktop Access* (NVDA)[3] [Mealin and Murphy-Hill 2012]. Entretanto, os leitores de tela não permitem que o usuário compreenda o contexto completo quando o conteúdo apresenta imagens. Esse problema pode ser contornado por meio de descrições das imagens, que podem ser providas pelo desenvolvedor do *site* durante sua criação.

Grande parte do conteúdo das redes sociais mais utilizadas atualmente atrela fortemente a descrição (ou o texto da postagem) com alguma imagem associada (ou até mesmo mais de uma imagem). Para tornar a experiência de utilização das aplicações com mídia visual acessível é necessário prover descrições alternativas para as imagens. Apesar da existência do atributo *"alt"* (texto alternativo) em páginas Web, muitos *sites* não o utilizam nas imagens e, quando falamos de redes sociais, o assunto fica ainda mais complexo. A maior parte dos usuários não sabem como criar uma descrição acessível ou simplesmente não pensam no público com deficiência na hora de fazer uma postagem.

Ao pensar na experiência do usuário durante a utilização dos sistemas de informação, é necessário considerar quais são os requisitos de acessibilidade para aumentar o público da aplicação. Mas e quando os desenvolvedores não são os únicos responsáveis pela acessibilidade? **Como melhorar a experiência dos usuários com deficiência visual em relação ao conteúdo de imagem publicado por outros usuários?** Este trabalho busca explorar tais questões ao propor e implementar um sistema, denominado Web-Assistente de Reconhecimento Digital de Imagens (WARD), que realiza a descrição automática de imagens que não possuem o atributo *"alt"* definido. O sistema deve funcionar em conjunto com o *browser* do usuário, na forma de extensão, para identificar e criar textos alternativos para as imagens disponíveis na tela do usuário e torná-las acessíveis para o público com deficiência visual.

Para atingir o objetivo principal do trabalho é necessário cumprir os objetivos específicos:

- Analisar as ferramentas atuais de acessibilidade para pessoas com deficiência visual e levantar as principais funcionalidades em comum entre elas.
- Criar uma extensão para *browser* capaz de ler e modificar o arquivo de Linguagem de Marcação de Hipertexto (HTML, do inglês *Hypertext Markup Language*) de uma página Web.
- Criar uma Interface de Programação de Aplicação (API, do inglês *Application Programming Interface*), para ser consumida pela extensão, que seja capaz de gerar uma descrição para uma determinada imagem a partir de um *Uniform Resource Locator* (URL).

A aplicação proposta e implementada é um sistema gratuito e de código-aberto composto por duas camadas. A primeira é uma extensão em HTML, JavaScript e *Cascading Style Sheets* (CSS) exclusiva para o Google Chrome[4]. A segunda é uma API em Node.js responsável por gerar as descrições para as imagens articulando o uso das APIs Google Cloud Translate e Google Cloud Vision de acordo com o necessário. A aplicação

[3]NV Access - Disponível em: https://www.nvaccess.org/. Acesso em: 12 de setembro de 2021.

[4]Google Chrome - Disponível em: https://www.google.com/chrome/. Acesso em: 26 de setembro de 2021.

é especificada e projetada por meio de diagramas da Linguagem de Modelagem Unificada (UML, do inglês *Unified Modeling Language*). O sistema é implementado para entender quais imagens devem ter uma descrição, quais descrições já existentes devem ser traduzidas e identificar e descrever novas imagens à medida em que elas aparecem. Os resultados obtidos mostram que o sistema é capaz de gerar e disponibilizar textos alternativos para imagens da Web que antes não eram acessíveis a deficientes visuais. Também mostram que, devido ao baixo custo de uso das APIs do Google Cloud, o sistema é financeiramente viável.

Este trabalho está estruturado em 7 seções. A Seção 2 deste documento contém os fundamentos teóricos utilizados no trabalho. Na Seção 3, são apresentados os trabalhos relacionados. Em seguida, apresenta-se a metodologia utilizada no projeto, na Seção 4. A Seção 5 contempla a especificação, projeto e implementação do sistema. As discussões sobre o trabalho estão na Seção 6. Por fim, as conclusões são apresentadas na Seção 7.

2. Referencial Teórico

Nesta seção são descritos os principais conceitos que estão relacionados a sistemas de acessibilidade e técnicas que são utilizadas no trabalho.

2.1. Acessibilidade para Deficientes Visuais

Quando se pensa em acessibilidade na Web, não se trata de um assunto completamente novo. Na verdade, é uma questão que tem sido estudada há muitos anos, e uma das principais diretrizes a ser considerada nesse assunto é a *Web Content Accessibility Guidelines* (WCAG), um conjunto de recomendações a respeito de acessibilidade na Web que foi publicada em maio de 1999 pela *World Wide Web Consortium* (W3C), principal organização de padronização da internet [Adams et al. 2021].

Dentre as recomendações da WCAG, uma das principais é referente aos textos alternativos para conteúdo não-textual. De acordo com a W3C, é necessário que os textos alternativos estejam ligados por meio do código-fonte ao conteúdo não-textual de forma que as tecnologias assistivas sejam capazes de lê-los e usá-los conforme o necessário [Adams et al. 2021][5]. Textos alternativos podem ser inseridos em imagens na página HTML por meio do atributo *"alt"*, e geralmente é esse o atributo analisado pelos leitores de tela para descrever a imagem para o usuário.

2.2. Tecnologias Assistivas

Tecnologia assistiva (AT, do inglês *Assistive Technology*) é um termo usado para definir produtos e serviços usados por pessoas com deficiência para permitir e melhorar a inclusão das mesmas em todos os âmbitos [de Witte et al. 2018][6]. Leitores de tela são exemplos de tecnologia assistiva para deficientes visuais, pois transformam todo o conteúdo originalmente textual em áudio por meio de um sintetizador de voz. O sistema proposto usa o funcionamento de leitores de tela como base para prover a descrição das imagens para o usuário ao prover textos alternativos que o leitor possa analisar.

[5]No original: *In order for people with disabilities to be able to use this text - the text must be "programmatically determinable" - This means that the text must be able to be read and used by the assistive technologies (and the accessibility features in browsers) that people with disabilities use.*

[6]No original: *Assistive technology (AT) is an umbrella term for products and related services used by persons with disability to enable and enhance their inclusion in all domains of participation.*

2.3. Descrição de Imagens

Imagens podem ser descritas por diversos métodos, seja por pessoas ou por robôs. Exemplos de processos de descrição são descrição de cores, objetos importantes na imagem, informações sobre o fundo e outros conteúdos que o usuário talvez não seja capaz de perceber somente por meio do texto original [Gleason et al. 2019]. Existem Inteligências Artificiais (IA) que fazem a identificação do conteúdo de uma imagem. Em geral, existem dois principais tipos de algoritmos para a identificação do conteúdo: reconhecimento de rótulo único (SLR, do inglês *single-label recognition*) e reconhecimento de rótulo múltiplo (MLR, do inglês *multi-label recognition*). O SLR parte do pressuposto que uma imagem possui apenas um rótulo, enquanto o MLR assume que há mais de um rótulo presente na imagem [Gao et al. 2017]. Diversos fatores estão relacionados com a identificação de um elemento, como o tamanho do elemento na imagem, a localização espacial e a visibilidade [Gao and Zhou 2021]. O algoritmo usado pelo Google Cloud Vision, que é usado na aplicação proposta, é do tipo MLR.

3. Trabalhos Relacionados

O Twitter[7] e o Facebook[8] disponibilizam uma funcionalidade para que os usuários adicionem uma descrição de texto alternativo às imagens anexadas na postagem. Além disso, também existe um movimento conhecido como "#PraCegoVer"[9] (ou "#PraTodosVerem") nas redes sociais que utilizam *hashtags* que consiste em trazer uma descrição textual para postagens que têm imagens anexadas, provida pelo próprio autor da publicação. As publicações feitas utilizando alguma das *hashtags* são agrupadas dentro da rede social.

Apesar das funcionalidades existentes nessas plataformas, muitos usuários não as usam por não conseguir encontrá-las ou por não saber o que escrever [Sacramento et al. 2020]. Além disso, essas funcionalidades não estão presentes em outros *sites* e redes sociais, como o WhatsApp e o YouTube. Em uma amostra de 9,22 milhões de tuítes, 1,09 milhões possuíam fotos anexadas e apenas 0,1% destes possuíam textos alternativos criados com a funcionalidade provida pela rede social, o que reforça essa ideia [Gleason et al. 2019].

Gleason et al. (2020) propuseram uma extensão para *browser* capaz de ler e editar o HTML do Twitter, denominada Twitter A11Y. Sempre que uma nova imagem é carregada, a extensão extrai a URL da imagem e faz uma requisição no servidor *backend* da aplicação, onde a imagem é processada para gerar o texto alternativo adequado [Gleason et al. 2020b]. A resposta da requisição com o texto alternativo é enviada de volta para a extensão, que edita o HTML do Twitter novamente para inserir o texto no atributo *"alt"* da imagem processada. A extensão Twitter A11Y, no entanto, funciona apenas no Twitter, diferente da proposta deste trabalho.

Guinness et al. (2018) desenvolveram o Caption Crawler, um *plugin* para *browser* que funciona de maneira semelhante ao Twitter A11Y, identificando as imagens na página HTML para inserir textos alternativos na propriedade *"alt"* [Guinness et al. 2018]. Toda-

[7]Twitter - Disponível em: https://twitter.com/. Acesso em: 12 de setembro de 2021.

[8]Facebook - Disponível em: https://facebook.com/. Acesso em: 12 de setembro de 2021.

[9]#PraCegoVer - Página do Facebook - Disponível em: `https://www.facebook.com/PraCegoVer/`. Acesso em: 20 de abril de 2021.

via, o Caption Crawler utiliza o mecanismo de pesquisa reversa do Google[10] para encontrar correspondências da imagem e reutilizar descrições que foram usadas anteriormente. A abordagem proposta no trabalho não é capaz de inserir descrições em qualquer imagem, pois é necessário que ela tenha sido descrita previamente por outra pessoa ou mecanismo, diferente da proposta deste trabalho.

Bodi et al. (2021) apresentam uma ferramenta para auxiliar deficientes visuais a assistirem vídeos com dois robôs diferentes: *NarrationBot*, um robô que descreve cenas nos vídeos a cada nova cena, e *InfoBot*, um robô que responde perguntas específicas sobre a cena quando solicitado [Bodi et al. 2021]. Para fazer a descrição de cenas, a aplicação desenvolvida pelos autores escolhe o melhor *frame* do vídeo entre os *frames* repetidos onde a cena muda e analisa os elementos mais importantes da imagem, processo semelhante ao que é executado pela aplicação desenvolvida neste trabalho.

4. Metodologia

Esta seção tem como objetivo especificar todas as etapas do trabalho, detalhando o que foi utilizado para o desenvolvimento do sistema e a metodologia especificada para realizar os testes na aplicação. Na seção também são especificadas as tecnologias utilizadas em cada parte do trabalho e suas versões.

4.1. Etapas do Trabalho

Este trabalho apresenta a proposta e implementação de um *software* capaz de identificar e criar descrições para as imagens disponíveis em uma página Web. As seguintes etapas são realizadas para o desenvolvimento do trabalho:

I. Especificação
II. Projeto do Sistema
III. Implementação
IV. Validação

Na etapa de **especificação**, as principais funcionalidades e regras de negócio são identificadas e representadas. Os principais meios de obtenção dos requisitos são a análise de sistemas relacionados e pesquisas sobre o funcionamento de tecnologias assistivas para deficientes visuais. Por meio do processo, são definidos os principais requisitos para desenvolver o Mínimo Produto Viável (MVP, do inglês *Minimum Viable Product*). Ainda na etapa, o diagrama de caso de uso do sistema e o diagrama de atividades do seu requisito principal são desenvolvidos com auxílio da ferramenta Lucidchart[11].

Na etapa de **projeto do sistema**, são criados os diagramas de pacotes e de implantação (ambos da UML) com o objetivo de definir os elementos arquiteturais do sistema. O diagrama de pacotes detalha como o sistema é divididos e suas respectivas interações. Já o diagrama de implantação mostra a maneira de execução do sistema, incluindo o meio pelo qual os componentes se comunicam.

Na etapa de **implementação**, todos os algoritmos necessários para entregar as funcionalidades previstas nos diagramas são implementados e testados no nível de unidade. A implementação foi dividida em dois *softwares*: uma extensão para o *browser*

[10]Google - Disponível em: https://www.google.com.br/. Acesso em: 26 de setembro de 2021.
[11]Lucidchart - Disponível em: https://www.lucidchart.com/. Acesso em: 26 de setembro de 2021.

Google Chrome e uma API capaz de integrar com os sistemas externos necessários para a descrição e tradução. As linguagens, ferramentas de desenvolvimento e *frameworks* utilizados na implementação são especificados na Seção 4.2.

Na etapa de **validação**, são feitos testes de sistema e de cenário para avaliar a integração entre a ferramenta assistiva de leitura de tela NVDA (na versão 2021.2) e o sistema WARD. Para a avaliação, define-se que nas redes sociais Twitter, Facebook e Instagram[12], que são amplamente usadas atualmente, os seguintes cenários sejam avaliados em termos de funcionalidade e de tempo de resposta:

1. Leitor de tela ativo e extensão inativa.
2. Leitor de tela ativo e extensão ativos, tradução inativa.
3. Leitor de tela, extensão e tradução ativos.

As APIs pertencentes ao Google Cloud utilizadas no sistema são plataformas de Software como Serviço (SaaS, do inglês *Software as a Service*). Para utilizá-las, é necessário ter uma chave de API que pode ser criada por meio do Google Cloud Console[13]. O sistema WARD necessita de uma chave de API gerada pelo usuário para funcionar. As chaves de API utilizadas nos testes não são disponibilizadas ao público.

4.2. Ambiente de Implementação

O *framework* escolhido para o desenvolvimento da API é o Node.js, um *framework* de alta escalabilidade que utiliza JavaScript como linguagem padrão em seu ambiente de execução. A escolha do Node.js se deve a diversas vantagens de uso da aplicação, sendo a principal o fato da linguagem de programação ser JavaScript, que conta com diversos materiais de apoio distribuídos pela Web. Além dessa vantagem, há também o fator da flexibilidade de utilizar o maior repositório de *softwares* do mundo para gerenciar pacotes: o *npm*[14]. Para a implementação da extensão, foram utilizadas as linguagens JavaScript, HTML e CSS devido ao comportamento de extensões se assemelhar ao de páginas Web e tais linguagens serem amplamente usadas para o desenvolvimento das mesmas. A versão do Node.js utilizada é a v14.15.0. A extensão utiliza a versão ES2015 do JavaScript, a linguagem de marcação HTML5 e a linguagem CSS3.

O Visual Studio Code[15] é o editor de código utilizado para o desenvolvimento da extensão e da API. Se trata de um editor de código multiplataforma da Microsoft[16] voltado para o desenvolvimento de aplicações Web, dando grande suporte para as linguagens utilizadas no trabalho. Apesar do objetivo principal ser o desenvolvimento Web, o editor também dá suporte para diversas outras linguagens. O ambiente de desenvolvimento e testes é o sistema operacional Windows 10 na versão 10.0.19042 *Build* 19042.

Para controle de versão de código, é utilizado o sistema Git[17], junto a plataforma GitHub[18], na qual todas as mudanças e atualizações de códigos são armazenadas. O Kan-

[12]Instagram - Disponível em: https://www.instagram.com/. Acesso em: 16 de outubro de 2021.

[13]Google Cloud Console - Disponível em: https://console.cloud.google.com/. Acesso em: 05 de outubro de 2021

[14]npm - Disponível em: https://www.npmjs.com/. Acesso em: 26 de setembro de 2021.

[15]Visual Studio Code - Disponível em: https://code.visualstudio.com/. Acesso em: 26 de setembro de 2021.

[16]Microsoft - Disponível em: https://www.microsoft.com/pt-br/. Acesso em: 16 de outubro de 2021.

[17]Git SCM - Disponível em: https://git-scm.com/. Acesso em: 26 de setembro de 2021.

[18]GitHub - Disponível em: https://github.com/. Acesso em: 26 de setembro de 2021.

ban é o *framework* para gerenciamento do projeto escolhido para o processo de desenvolvimento do sistema WARD, devido a sua simplicidade de uso e suporte na ferramenta Trello[19]. As tarefas são criadas com base nas *issues* do repositório do projeto no GitHub, que são criadas ao longo do desenvolvimento do projeto e divididas em três principais entregas (*milestones*): v0.1, v0.5 e v1.0, sendo a última a entrega da primeira versão funcional do sistema por completo. Para gerenciar as versões do projeto, é utilizado um repositório no GitHub com duas pastas de controle: uma para a extensão e outra para a API. O fluxo de trabalho escolhido é o *trunk-based development*[20], um método onde os desenvolvedores trabalham exclusivamente no *trunk* principal (*main*), devido a facilidade de uso para projetos em início e times pequenos de desenvolvimento.

5. Resultados

Esta seção tem como objetivo informar os resultados de cada etapa da Seção 4, trazendo a descrição e limitações do sistema por meio da especificação. Ela apresenta os diagramas de pacotes, de implantação, de atividade e de casos de uso no projeto do sistema. Aborda sobre as duas camadas do sistema, sendo elas a extensão e a API, detalhadamente na implementação e os resultados e testes aplicados nas páginas Web na etapa de validação.

5.1. Especificação

A proposta do sistema WARD é possibilitar que a experiência de acesso a páginas Web seja acessível a usuários deficientes visuais. Para isso, define-se que a extensão deve funcionar em conjunto com o *software* leitor de tela de preferência do usuário. Também é necessário que o sistema funcione em segundo plano e sem a necessidade de ações por parte do usuário para ser ativado. A ativação do sistema consiste na identificação de imagens na página Web atual do usuário, que resulta na geração de uma descrição para cada imagem identificada.

O sistema gera descrições a partir da integração com a API Google Cloud Vision, uma API capaz de descrever imagens com bons resultados e gratuita para uso limitado. As descrições são inicialmente em inglês, e portanto é necessário traduzir o conteúdo para o português. Para atingir tal objetivo, o sistema integra com a API Google Cloud Translate, também de uso gratuito limitado. Quando a descrição é criada e traduzida, o sistema deve editar a página Web para inseri-la no texto alternativo da imagem descrita.

A Figura 1 apresenta o diagrama de casos de uso (*use case*) do sistema WARD. Por meio dele é possível identificar os principais atores, que são o próprio usuário, o leitor de tela e as APIs Google Cloud Vision e Google Cloud Translate. Além disso, o diagrama também apresenta as interações entre os atores.

5.2. Projeto do Sistema

O **diagrama de pacotes** (Figura 2) mostra a divisão de pacotes e organiza os modelos utilizados no sistema. O sistema WARD é dividido em dois principais componentes, sendo o primeiro a extensão WARD. A extensão utiliza o segundo principal componente, que é a API WARD. A API é responsável por realizar a comunicação do sistema com

[19]Trello - Disponível em: https://trello.com/. Acesso em: 26 de setembro de 2021.

[20]Trunk Based Development - Disponível em: https://trunkbaseddevelopment.com/. Acesso em: 26 de setembro de 2021.

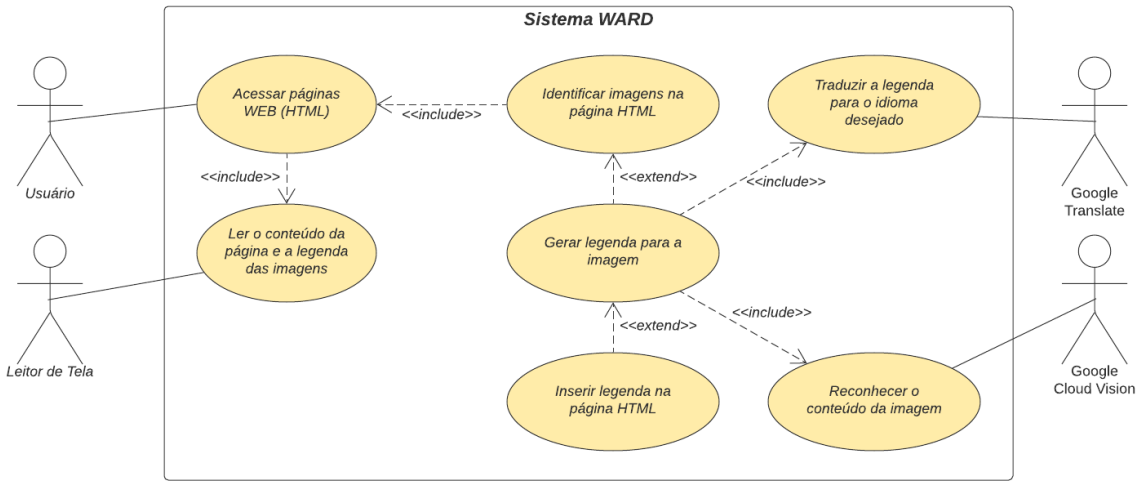

Figura 1. Diagrama de de Caso de Uso

os demais sistemas externos do Google Cloud, que são as APIs Google Cloud Vision e Google Cloud Translate.

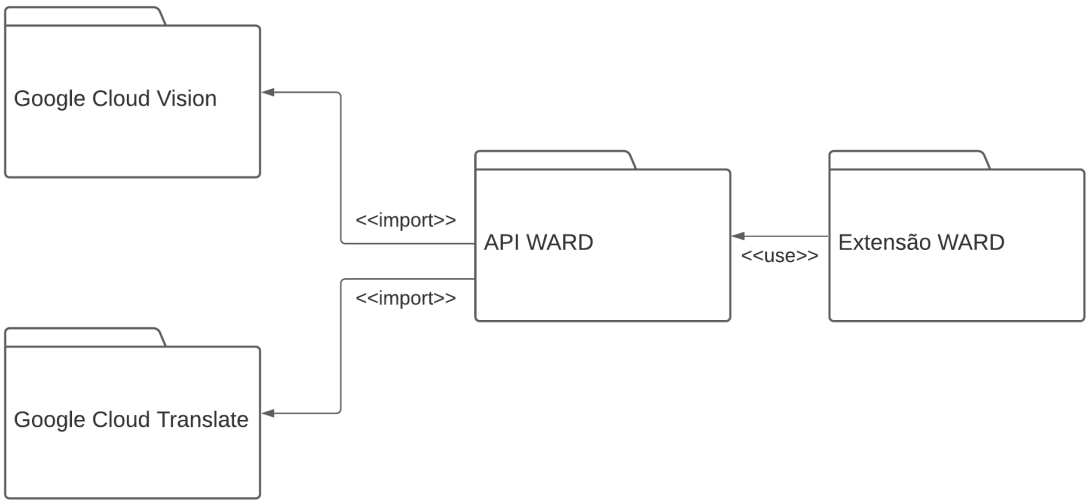

Figura 2. Diagrama de Pacotes

O **diagrama de implantação** da UML (Figura 3) é desenvolvido para demonstrar como a arquitetura do sistema WARD é executada. A extensão WARD é desenvolvida para o navegador Google Chrome e comunica com a API desenvolvida em Node.js para o envio de URLs de imagens por meio de conexões HTTP. A API WARD comunica também por meio de conexões HTTP com duas APIs da Google, sendo elas Google Cloud Vision e Google Cloud Translate. A comunicação é feita por meio de requisições HTTP.

O **diagrama de atividades** da UML (Figura 4) é desenvolvido para demonstrar o fluxo de controle das atividades do sistema. O fluxo começa na primeira raia, onde o usuário, ao navegar por páginas Web com um leitor de tela (segunda raia), ativa a extensão WARD, que lê e absorve o conteúdo HTML da página (terceira raia), filtrando todas as

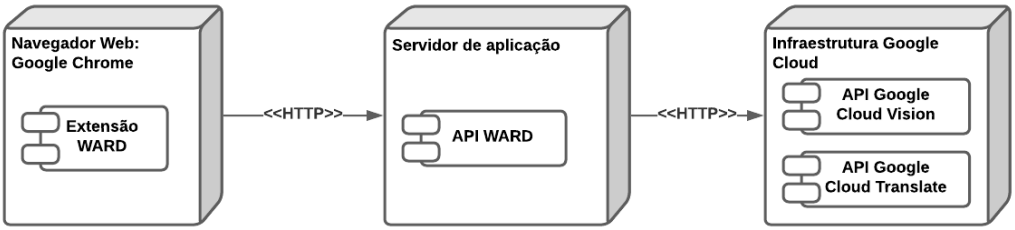

Figura 3. Diagrama de Implantação

tags HTML *img* com atributos *"alt"* vazios e enviando a URL da imagem para a API do sistema WARD (quarta raia). A API então envia a URL para a API Google Cloud Vision, que interpreta a imagem e gera elementos identificados na imagem e os manda de volta para a API WARD. Ao receber os elementos identificados da imagem, são enviados para a API Google Cloud Translate (quinta e última raia) que traduz os elementos e manda de volta como resposta. Com os elementos traduzidos, a API WARD gera uma descrição coerente e envia de volta para a extensão, que por sua vez altera o atributo *"alt"* das imagens da página Web.

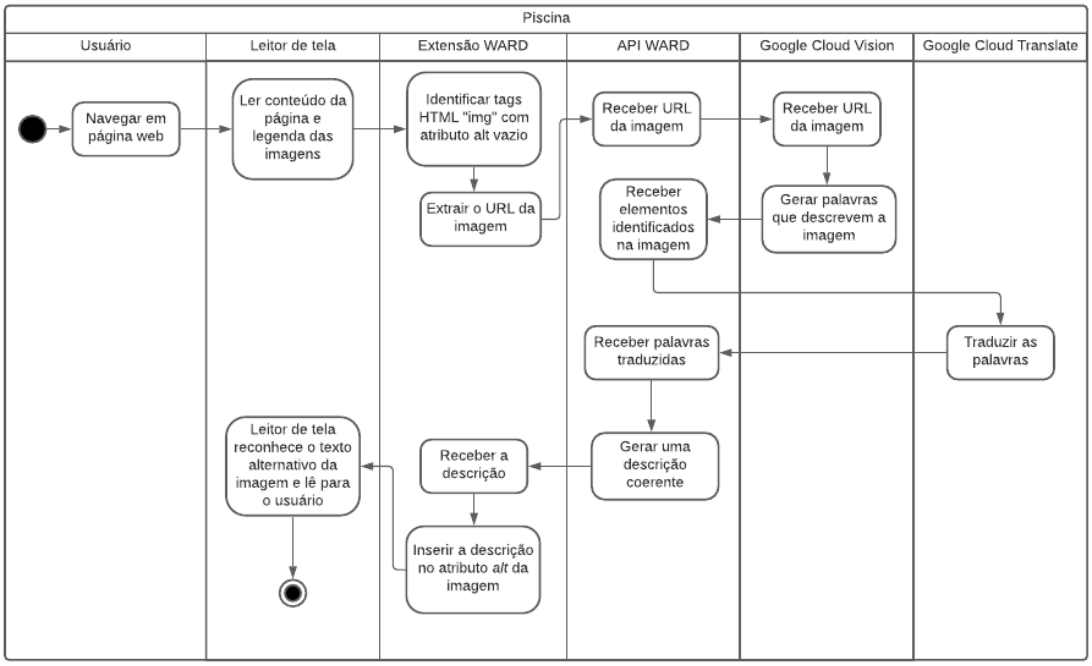

Figura 4. Diagrama de Atividades

5.3. Implementação

O sistema WARD é composto por duas camadas que operam em conjunto. A primeira camada é uma API que fornece um método para descrever uma imagem a partir de uma URL. A segunda camada é uma extensão para *browser* feita para o navegador Google Chrome, e é por meio dela que as imagens são identificadas. Além disso, também é

responsabilidade da extensão modificar o HTML da página Web acessada para inserir a descrição gerada pela API.

A API WARD é desenvolvida para funcionar como ponte entre a aplicação disponível para o usuário final (a extensão WARD) e as APIs externas do Google Cloud. Para desenvolvê-la, é utilizado o *framework* Node.js. O projeto possui três estruturas principais. A estrutura de inicialização do servidor Node é definida no arquivo *server.js*. A estrutura de rotas da API, que possibilita que as funções internas sejam chamadas por meio de requisições, é distribuída entre os arquivos *app.js*, *routes/index.js* e *routes/imageRoute.js*. Por fim, a estrutura principal é o *controller* definido por meio do arquivo *imageController.js*, que é responsável por definir os métodos de reconhecimento de imagem por meio do consumo da API Google Cloud Vision, de tradução de textos por meio da API Google Cloud Translate, tratamento dos dados gerados e criação da descrição de uma determinada imagem.

A função disponibilizada pela API recebe por parâmetro três campos principais no corpo de uma requisição do tipo POST no formato *JavaScript Object Notation* (JSON), sendo eles a URL da imagem a ser descrita, a descrição atual da imagem (se houver) e a linguagem de tradução da descrição. Também é necessário enviar a chave de API do Google Cloud, e é necessário que chave esteja habilitada tanto para a API Vision quanto para a API Translate. A chave é enviada por meio do campo *"key"*. Por padrão, a API WARD considera o português como parâmetro para a linguagem. A resposta da requisição contém um campo contendo o resultado (*true* ou *false*), um campo com uma mensagem a respeito do resultado e o campo *"caption"*, que contém a descrição gerada pela API. Caso não seja possível gerar a descrição devido a falta de elementos identificados pela API Google Cloud Vision, a descrição será preenchida com uma mensagem de aviso. Caso ocorra qualquer outra exceção, a requisição não retornará o campo *"caption"* e o erro será descrito no campo *"message"*.

A extensão WARD é executada em segundo plano após a instalação sem a necessidade de qualquer atalho pressionado pelo usuário ou qualquer interação com interfaces. Para habilitá-la é necessário registrar a chave de API em um *input* do HTML da extensão para que seja enviada em cada requisição. A Figura 5 apresenta a interface da extensão e sua localização no navegador, incluindo o campo de configuração da chave. Para desativar a extensão, é necessário acessar as configurações do navegador Google Chrome e desabilitá-la na aba extensões. A extensão lê todo o Modelo de Objeto de Documento (DOM, do inglês *Document Object Model*) da página Web assim que ela termina de carregar e sempre que novos elementos surgem. Em casos de *sites* que renderizam novos *posts* a medida que o usuário navega, como redes sociais, o novo conteúdo adicionado é obtido pela extensão por meio do método *scroll*.

Todas as manipulações do HTML da página são realizadas por meio de funções em JavaScript. Ao ler todo o DOM da página por meio de uma função, a extensão filtra todas as *tags "img"* do HTML. Com as *tags* filtradas, o atributo *"src"* (abreviação de *source*, que é o atributo responsável por apontar o endereço de determinado elemento) de cada imagem é extraído para enviá-lo por meio de uma requisição para a API WARD. Após o fim do processamento da imagem pela API, a extensão recebe como resposta a descrição da imagem. A extensão cria o atributo *"alt"* na imagem caso ele ainda não exista ou o altera caso ele já esteja presente.

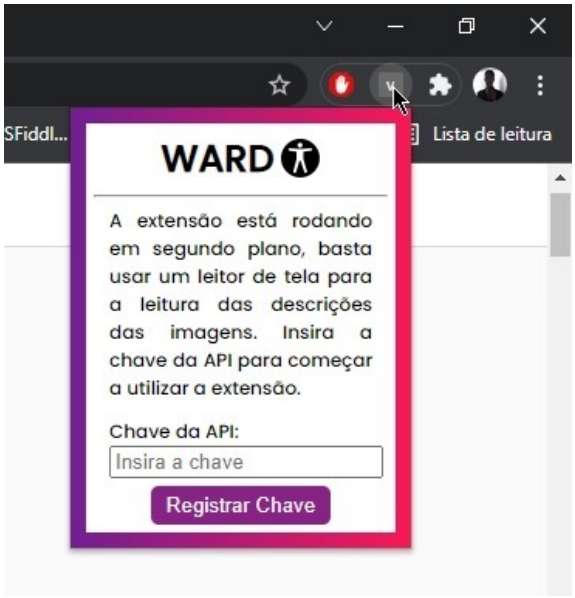

Figura 5. Interface da extensão no navegador Google Chrome para configuração da chave de API do Google Cloud.

Para otimizar o sistema, a extensão identifica textos alternativos que não contém informações providas pelo usuário nas redes sociais Instagram, Twitter e Facebook. Portanto, atributos *"alt"* que tenham sido preenchidos sem informações sobre as imagens ou apenas de forma automática nessas páginas serão complementados, no final, com a descrição gerada pelo sistema. Entretanto, caso o texto alternativo já seja útil, a extensão apenas traduz a descrição para minimizar as requisições na API Google Cloud Vision. Nos casos onde não há atributo *"alt"*, a extensão cria o atributo e a descrição é inserida por completo. À medida que novos elementos são renderizados na tela, as funções que capturam as *tags "img"* vão sendo reativadas e a extensão refaz o processo de chamada de requisição para as novas imagens.

O código-fonte do projeto é disponibilizado no GitHub por meio do repositório ward (www.github.com/lucas-izidorio/ward). Um vídeo de apresentação do sistema em execução na perspectiva do usuário final também é disponibilizado por meio do YouTube (www.youtube.com/watch?v=L5zbLzb04SU), que explica como instalar a extensão, ativá-la e utilizá-la. Além disso, o vídeo demonstra os três cenários de uso que são descritos na Seção 5.4.

5.4. Validação

Na validação do sistema, são avaliados o tempo de resposta de cada imagem que foi enviada para a API com ou sem tradução e o tempo de carregamento da página nos cenários previamente listados. Quanto ao carregamento da página, não é notada nenhuma diferença nas três redes sociais, Twitter, Facebook e Instagram, já que a extensão roda em segundo plano e a manipulação do DOM é feita após a página carregar e também só manipula um elemento em específico de uma *tag* HTML. Quanto ao tempo de resposta para gerar a descrição nos cenários na qual a extensão está ativa, sendo eles, com ou sem tradução, é notada pequena diferença de tempo do resultado final, com exceção do primeiro carregamento de uma página que é o maior tempo notado devido a falta de *cache* do navegador.

Os testes foram realizados nas redes sociais Facebook, Instagram e Twitter, analisando o tempo de resposta para o carregamento total de uma mesma imagem trinta vezes em cada rede. A Tabela 1 apresenta a média, desvio padrão, maior tempo e menor tempo de carregamento para cada rede social em cada um dos cenários: com a extensão inativa, somente com a descrição ativa (sem tradutor) e com a descrição e tradutor ativos. É possível concluir que a diferença é muito pequena, já que a tradução de palavras é muito mais rápida do que a detecção de elementos visuais de uma imagem. Vale notar que o tempo de resposta também varia bastante, às vezes podendo ser muito mais rápido do que a média, como por exemplo no primeiro carregamento da página em que os tempos são maiores, enquanto nos últimos o tempo reduziu consideravelmente.

Tabela 1. Tempo até o carregamento total da imagem em uma página Web (em milissegundos).

Rede social	Métrica	Sistema inativo	Sem tradutor	Com tradutor
Facebook	Maior tempo	2.985	6.105	6.265
	Menor tempo	2.502	3.474	3.504
	Desvio padrão	138	762	960
	Tempo médio	2.705	4.374	4.501
Instagram	Maior tempo	1.621	4.263	5.150
	Menor tempo	1.325	2.630	2.733
	Desvio padrão	89	433	688
	Tempo médio	1.502	3.211	3.251
Twitter	Maior tempo	3.051	4.373	6.445
	Menor tempo	2.852	3.710	4.216
	Desvio padrão	58	248	672
	Tempo médio	2.932	4.107	4.546

Com os resultados apresentados na tabela é notável que o tempo de espera que um usuário deve lidar é muito pequeno e provavelmente imperceptível. O maior tempo de resposta entre as três redes sociais com tradução foi de 3,6 segundos e o menor de 799 milissegundos. Como o usuário provavelmente terá que ouvir por meio do leitor de tela os outros elementos da tela antes de chegar de fato na imagem, esse tempo tende a ter pouco efeito na dinâmica de uso.

As APIs do Google Cloud são de uso gratuito limitado. Para utilizar as APIs, é necessário possuir uma chave de API. A chave pode ser gerada por meio do Google Cloud Console[21], plataforma do Google para gestão de uso das ferramentas. O sistema WARD permite que o usuário insira suas próprias chaves por meio de um campo na extensão. As chaves de API funcionam sob demanda, ou seja, o preço varia de acordo com a quantidade de requisições feitas. Além disso, a quantidade de requisições é calculada de forma mensal, o que faz com que o total de requisições feitas seja reiniciado ao fim de cada mês.

A API Google Cloud Translate é cobrada de acordo com o total de caracteres enviados nas requisições durante o mês. Até o limite de 1 milhão de caracteres, o uso da API é gratuito. Caso o uso ultrapasse o limite durante o mês, o custo da API é de R$ 56,39 por milhão de caracteres. Já a API Google Cloud Vision é cobrada pela quantidade de requisições feitas. O limite de gratuidade é de mil requisições por mês, e o custo da

[21]Google Cloud Console - Disponível em: https://cloud.google.com/cloud-console. Acesso em: 27 de outubro de 2021.

API ao ultrapassar o limite gratuito é de R$ 8,45 por cada mil requisições.

Para estimar os custos para manter a API são considerados os dados relativos ao comportamento de um usuário que consome, em média, uma hora e meia de conteúdo em páginas Web por dia. Durante o tempo de uso, são levados em consideração três possíveis cenários para um conjunto de 100 imagens identificadas. O cenário pessimista leva em consideração que apenas 10% das imagens possuem texto alternativo. O segundo cenário é o cenário moderado, onde 50% das imagens possuem texto alternativo. Por fim, no cenário otimista, 90% das imagens já possuem texto alternativo. Para todas as imagens, considera-se que o serviço de tradução deve ser executado. Os valores calculados para o cenário levam em conta o uso médio durante 30 dias, que é o ciclo de cobrança e renovação do Google Cloud.

Em todos os cenários, são feitas 100 requisições para a API de tradução. O resultado ao longo dos 30 dias é de 3.000 requisições de tradução, com 540.000 caracteres enviados levando em consideração a média de 180 caracteres por requisição. Sendo assim, apenas a diferença de uso da API de descrição de imagens é relevante para o custo final.

No **cenário pessimista**, são feitas 90 requisições para a API de descrição de imagens por dia de uso, resultando em 2.700 requisições ao fim do período de cobrança, com o valor final de R$ 14,38. Já no **cenário moderado**, são feitas 50 requisições para a API de descrição de imagens, sendo 1.500 requisições no total e R$ 4,23 de custo final. O **cenário otimista** conta com apenas 10 requisições para a API de descrição de imagens por dia, o que resulta em apenas 300 requisições por mês e se mantém dentro do limite gratuito. Por fim, é importante destacar que mesmo em usos acima do cenário otimista é possível permanecer dentro do limite gratuito (33% das imagens com texto alternativo).

6. Discussões

O usuário não precisa se preocupar com o desenvolvimento da lógica para consumir os serviços em nuvem do Google, pois o processo é abstraído para que seja necessário somente possuir uma chave válida. O resultado é um *software* otimizado, capaz de orquestrar o consumo das APIs do Google Cloud de acordo com o necessário para otimizar o desempenho e os custos. A API de descrição de imagens, que possui o maior custo devido ao limite gratuito ser baixo, só é chamada caso o sistema não identifique textos alternativos providos pela página Web acessada.

A escolha das APIs do Google Cloud, mesmo que elas não sejam totalmente gratuitas, se deve ao fato de se tratar de uma infraestrutura em nuvem com grande potencial de evolução. As IAs de tradução e descrição de imagens são constantemente treinadas e atualizadas, sendo necessário apenas dar manutenção na parte de integração do sistema WARD com as mesmas. Caso o sistema fosse baseado em uma estrutura interna para descrição de imagens e tradução de textos, os modelos seriam estáticos e não estariam em aprendizado constante como os do Google.

O valor médio de consumo das APIs é financeiramente viável. O sistema WARD é totalmente gratuito e de código-aberto, enquanto as APIs do Google Cloud possuem baixo custo mensal para o usuário, podendo ser até mesmo 100% gratuita caso as chaves de API sejam configuradas para bloquear gastos acima do limite gratuito ou caso o uso não

ultrapasse as 1.000 imagens (para descrição) e 1.000.000 caracteres (para tradução). Até mesmo em um cenário de uso médio em que nenhuma imagem possui texto alternativo, o custo é de R$ 16,92, que ainda é viável. Além disso, também é possível utilizar uma conta de nível gratuito do Google Cloud para receber um crédito de R$ 1.605,25 durante um período de 90 dias.

O sistema também possui viabilidade funcional, pois não é necessário que o usuário espere por muito tempo até que uma imagem seja descrita. O leitor de tela percorre todos os elementos da página Web, então é comum que dados textuais sejam lidos antes de chegar à primeira imagem da página. Dessa forma, quando o leitor encontrar uma imagem, normalmente o texto alternativo já terá sido preenchido de acordo com o necessário pelo sistema. Além disso, o processo ocorre em segundo plano e não impacta no carregamento da página Web. A navegação do usuário também não é impactada pelas mudanças que o sistema faz na página ao longo dos novos carregamentos.

7. Conclusão

Este trabalho apresentou a proposta e o desenvolvimento de uma aplicação capaz de identificar e descrever imagens em qualquer página Web. Os resultados obtidos no processo de validação mostraram que a ferramenta é viável tanto em funcionalidade quanto em custos. O sistema é capaz de inserir as descrições sem que a experiência de navegação do usuário seja impactada, pois é capaz de rodar em segundo plano e o tempo de resposta das requisições é baixo, o que faz com que sempre haja uma descrição disponível para uma imagem descrita pelo leitor de tela. Além disso, a forma como as APIs do Google Cloud são articuladas por meio do sistema desenvolvido faz com que o uso das mesmas seja otimizado ao máximo, fazendo requisições apenas quando identificada a necessidade.

O trabalho tem como principal contribuição o fator de aumento da inclusão de deficientes visuais. Por meio do sistema, é possível ter acesso a serviços atuais no quesito de IA para tradução e descrição de imagens sem a necessidade de ter conhecimentos sobre programação. O sistema, uma vez configurado, está pronto para ser executado em qualquer página Web. As descrições geradas pelo sistema podem ajudar pessoas deficientes visuais a entender o contexto de imagens e aumentar o acesso ao conteúdo disponível em páginas Web, principalmente em redes sociais, onde textos alternativos são mais escassos. Por fim, o sistema funciona em harmonia com leitores de tela, a principal AT voltada para deficientes visuais.

Como trabalhos trabalhos futuros, uma possibilidade de melhoria para o sistema seria a formulação de uma frase consistente com as *labels* geradas pelo Google Cloud Vision, assim como foi feito no trabalho de Hsu et al. (2021), no qual gráficos são examinados por um sistema que gera uma legenda com base em um *dataset* com milhões de figuras que auxiliam na descrição [Hsu et al. 2021]. Outra possibilidade seria generalizar a detecção de textos alternativos existentes para qualquer página Web, e não somente no Twitter, Instagram e Facebook. Por fim, seria possível adicionar Reconhecimento Ótico de Caracteres (OCR, do inglês *Optical Character Recognition*) para complementar a descrição em casos de imagens que possuam conteúdo textual.

Referências

Adams, C., Campbell, A., Montgomery, R., Cooper, M., and Kirkpatrick, A. (2021). Web content accessibility guidelines (WCAG) 2.2. Disponível em: https://www.w3.org/TR/WCAG22/. Acesso em: 16 de outubro de 2021.

Bodi, A., Fazli, P., Ihorn, S., Siu, Y.-T., Scott, A. T., Narins, L., Kant, Y., Das, A., and Yoon, I. (2021). Automated video description for blind and low vision users. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA. Association for Computing Machinery.

de Witte, L., Steel, E., Gupta, S., Ramos, V. D., and Roentgen, U. (2018). Assistive technology provision: towards an international framework for assuring availability and accessibility of affordable high-quality assistive technology. *Disability and Rehabilitation: Assistive Technology*, 13(5):467–472. PMID: 29741965.

Gao, B.-B., Xing, C., Xie, C.-W., Wu, J., and Geng, X. (2017). Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, 26(6):2825–2838.

Gao, B.-B. and Zhou, H.-Y. (2021). Learning to Discover Multi-Class Attentional Regions for Multi-Label Image Recognition. *IEEE Transactions on Image Processing*, 30:5920–5932.

Gleason, C., Carrington, P., Cassidy, C., Morris, M. R., Kitani, K. M., and Bigham, J. P. (2019). "it's almost like they're trying to hide it": How user-provided image descriptions have failed to make twitter accessible. In *The World Wide Web Conference*, WWW '19, page 549–559, New York, NY, USA. Association for Computing Machinery.

Gleason, C., Carrington, P., Chilton, L. B., Gorman, B., Kacorri, H., Monroy-Hernández, A., Morris, M. R., Tigwell, G., and Wu, S. (2020a). Future research directions for accessible social media. *SIGACCESS Access. Comput.*, (127).

Gleason, C., Pavel, A., McCamey, E., Low, C., Carrington, P., Kitani, K. M., and Bigham, J. P. (2020b). Twitter a11y: A browser extension to make twitter images accessible. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA. Association for Computing Machinery.

Guinness, D., Cutrell, E., and Morris, M. (2018). Caption crawler: Enabling reusable alternative text descriptions using reverse image search. pages 1–11.

Hsu, T.-Y., Giles, C. L., and Huang, T.-H. K. (2021). Scicap: Generating captions for scientific figures.

IBGE (2021). *Pesquisa nacional de saúde : 2019 : ciclos de vida : Brasil / IBGE, Coordenação de Trabalho e Rendimento, [Ministério da Saúde]*. Instituto Brasileiro de Pesquisa de Estatística, Rio de Janeiro.

Mealin, S. and Murphy-Hill, E. (2012). An exploratory study of blind software developers. pages 71–74.

Mynatt, E. and Weber, G. (1994). Nonvisual presentation of graphical user interfaces: Contrasting two approaches.

Sacramento, C., Nardi, L., Ferreira, S. B. L., and Marques, J. a. M. d. S. (2020). #pracegover: Investigating the description of visual content in brazilian online social media.

In *Proceedings of the 19th Brazilian Symposium on Human Factors in Computing Systems*, IHC '20, New York, NY, USA. Association for Computing Machinery.