



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

ESCOLA POLITÉCNICA

PROGRAMA DE PÓS-GRADUAÇÃO

EM TECNOLOGIA EM SAÚDE

AMOS RODRIGUES MARTINS JUNIOR

**TECLADO ASSISTIVO DINÂMICO COM CONTROLES ALTERNATIVOS EM
AMBIENTE WEB**

**CURITIBA
2024**

AMOS RODRIGUES MARTINS JUNIOR

**TECLADO ASSISTIVO DINÂMICO COM CONTROLES ALTERNATIVOS EM
AMBIENTE WEB**

Dissertação apresentada ao Programa de Pós-Graduação em Tecnologia em Saúde da
Escola Politécnica da Pontifícia Universidade Católica do Paraná.

Orientador: Prof. Dr. Percy Nohama.

**CURITIBA
2024**

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Luci Eduarda Wielganczuk – CRB 9/1118

M386t
2024 Martins Junior, Amos Rodrigues
Teclado assistivo dinâmico com controles alternativos em ambientes web / Amos Rodrigues Martins Junior ; orientador: Percy Nohama. – 2024.
72 f. ; il. : 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2024
Bibliografia: f. 69-72

1. Tecnologia médica. 2. Distúrbios psicomotores. 3. Tecnologia assistiva.
4. Pessoas com deficiências – Meios de comunicação. I. Nohama, Percy.
I. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Tecnologia em Saúde. III. Título.

CDD. 20. ed. – 610.28



Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Tecnologia em Saúde

TERMO DE APROVAÇÃO DE DISSERTAÇÃO Nº 328

A Dissertação de Mestrado intitulada **“TECLADO ASSISTIVO DINÂMICO COM CONTROLES ALTERNATIVOS EM AMBIENTE WEB”**, defendida em sessão pública pelo(a) candidato(a) **Amos Rodrigues Martins Junior**, no dia **29 de agosto de 2024**, foi julgada para a obtenção do grau de Mestre em Tecnologia em Saúde, e aprovada em sua forma final, pelo Programa de Pós-Graduação em Tecnologia em Saúde.

BANCA EXAMINADORA:

Prof. Dr. Percy Nohama – Presidente – PUCPR

Prof. Dr. Maurício Kugler – ESIGELEC/France

Profa. Dra. Simone Infingardi Kruger – UERJ

A via original deste documento encontra-se arquivada na Secretaria do Programa, contendo a assinatura da Coordenação após a entrega da versão corrigida do trabalho.

Curitiba, 11 de novembro de 2024.

Prof. Dr. Percy Nohama
Coordenador do Programa de
Pós-Graduação em Tecnologia em Saúde
PUCPR

RESUMO

A inserção social por meio da internet utilizando computadores ocorre predominantemente através do uso de dispositivos de entrada como mouses e teclados. No entanto, pessoas com deficiências motoras enfrentam desafios significativos ao manusear esses dispositivos, levando-as a buscar controles alternativos para facilitar a utilização. Este projeto concentra-se no desenvolvimento de um teclado virtual dinâmico destinado a esse grupo. O objetivo principal é criar um *layout* diferenciado em uma plataforma web, incorporando funcionalidades de previsão de palavras e controle por meios alternativos, considerando as adversidades das condições motoras de pessoas com deficiências físicas. Este projeto foi desenvolvido utilizando a linguagem de programação web javascript, utilizando pacotes para programação de funções para controle de dispositivos alternativos para controle do teclado virtual, um método de predição de palavra estático baseado em entradas prévias do usuário foi adotado. O teclado virtual foi submetido a uma série de testes automatizados para validar sua viabilidade para distribuição aos usuários finais. Esses testes incluíram métricas como contagem de palavras por minuto e ganhos relativos à previsão de palavras. Os resultados foram promissores, com médias de até 35 palavras por minuto nos melhores cenários, representando um aumento de 5 palavras por minuto em comparação com a velocidade média de digitação em um teclado convencional. Em contraste, a abordagem assistiva utilizada como referência atinge médias de 23 palavras por minuto com usuários treinados. Com base nos resultados dos testes automatizados, conclui-se que o teclado virtual está pronto para ser distribuído para testes com usuários reais. Esses testes permitirão a coleta de indicadores mais precisos, considerando fatores como o tipo de controlador utilizado, a especificidade da deficiência motora e os fatores cognitivos que podem influenciar o uso da ferramenta, os quais não podem ser adequadamente simulados em testes automatizados.

Palavras-chave: Deficiências Motoras, Teclado Virtual, Predição de Palavras, Tecnologia Assistiva, Comunicação Aumentativa e Alternativa.

ABSTRACT

Social inclusion through the internet using computers is predominantly facilitated by input devices such as mice and keyboards. However, individuals with motor disabilities face significant challenges in handling these devices, which instigates them to seek alternative controls to ease usage. This project focuses on developing a dynamic virtual keyboard for this group. The main objective is to create a novel layout on a web platform, incorporating word prediction functionalities and control via alternative means, considering the adversities of motor conditions. This project was developed using the web programming language JavaScript, incorporating packages for programming functions to control alternative devices for virtual keyboard control. A static word prediction method based on the user's previous inputs was adopted. The virtual keyboard underwent a series of automated tests to validate its viability for distribution to end-users. These tests included metrics such as words per minute count and gain related to word prediction. The results were promising, with averages of up to 35 words per minute in the best scenarios, representing an increase of 5 words per minute compared to the average typing speed on a conventional keyboard. In contrast, the assistive approach used as inspiration achieves averages of 23 words per minute with trained users. Based on the results of the automated tests, it is concluded that the virtual keyboard can be distributed for testing with real users. These tests will allow for the collection of more precise indicators, considering factors such as the type of controller used, the specificity of the motor disability, and cognitive factors that may influence the use of the tool, which cannot be adequately simulated in automated tests.

Keywords: Motor Impairment, Virtual Keyboard, Assistive Technology, Augmentative and Alternative Communication, Word Prediction.

LISTA DE FIGURAS

Figura 1 - Uso de internet pelas pessoas em contexto mundial em 2022.....	15
Figura 2 - DARCI TOO é um computador com conexões para dispositivos alternativos.	18
Figura 3 – Diferença em tempos relativos à utilização do teclado virtual com previsões.	21
Figura 4 – Layout teclado virtual DASHER.	23
Figura 5 - Pontos levantados que um teclado virtual deve conter para prover acessibilidade.....	25
Figura 6 - Layout de um teclado virtual ideal após uma revisão de literatura.	26
Figura 7 - Software web ETM.....	27
Figura 8 - Software web ETM edição de layout de teclado.....	27
Figura 9 - Interface gráfica teclado virtual multilingual.	28
Figura 10 - Rastreamento de símbolos por rastreamento ocular contínuo.	29
Figura 11 - Interface gráfica teclado virtual controlado por rastreamento ocular e piscar dos olhos.....	30
Figura 12 - Eletrodos aliados na face do usuário para captura dos sinais elétricos.	31
Figura 13 - headset composto por eletrodos para captura de sinais elétricos.	32
Figura 14 - Interface teclado virtual por comandos de prompt.....	33
Figura 15 – Controle teclado virtual a partir da língua através de indutância variável.....	34
Figura 16 - Fluxograma até a digitação de um caractere por entrada realizada por controle bucal.....	35
Figura 17 - Interface acionadora do teclado virtual.....	36
Figura 19 - Fluxograma geral do software.	41
Figura 20 - Fluxo da conexão e utilização de um dispositivo incomum via HID. ..	43
Figura 21 - Exemplificação algoritmo preditivo.....	51
Figura 23 – Primeiro layout proposto.....	52
Figura 24 – Layout padrão duas linhas.	54

Figura 25 – Layout padrão quatro linhas.	54
Figura 26 - Tela configurações controles finalizadas.	55
Figura 27 - Captura da ação por contagem de tempo.....	56
Figura 28 - Captura da ação de forma imediata.....	56
Figura 29 - Reorganização do teclado.....	57
Figura 30 - Exibição dos números e símbolos.	58
Figura 31 - Tela feedback finalizada.	61
Figura 32 - Customização de Layout.	62

LISTA DE TABELAS

Tabela 1 - Contabilização de com quantos caracteres faltantes uma palavra foi predita.	40
Tabela 2 - Cenários de Testes Mapeados.	42
Tabela 3 - Resultados Primários Testes Automatizados	48
Tabela 4 - Resultados dos Testes Automatizados de Layout Vertical dos Controles Joystick	50
Tabela 5 - Resultados dos Testes Automatizados de Layout Vertical dos Controles Sensoriais	50
Tabela 6 - Resultados dos Testes Automatizados de Layout Horizontal dos Controles Joystick	52
Tabela 7 - Resultados dos Testes Automatizados de Layout Horizontal dos Controles Sensoriais	52

LISTA DE SIGLAS

WPM - Contagem de Palavras por Minuto

PPM - Palavras Preditas por Minuto

CPM - Contagem Caracteres por Minuto

EPM - Taxa de Erros por Minuto

HID - Dispositivo de Interface Humana

ITR - Taxa de Transferência de Informação

VP - Velocidade do Ponteiro

HCI - Interface Homem-Computador

IHM - Interface Homem-Máquina

BCI - Interface Cérebro-Computador

AI - Inteligência Artificial

HTML - Hypertext Markup Language

API - Application Programming Interface

USB - Universal Serial Bus

SUMÁRIO

1. INTRODUÇÃO	10
1.1 PROBLEMA	11
1.2 JUSTIFICATIVA	12
1.3 OBJETIVOS	13
1.3.1 Objetivo Geral	13
1.3.2 Objetivos Específicos	13
2. FUNDAMENTAÇÃO TEÓRICA	15
2.1 Tecnologia Assistiva	15
2.2 Uso da Internet por Pessoas com Deficiências Motoras	16
2.2.1 Oportunidades de Trabalho	16
2.2.2 Comunicação e Inclusão Social	16
2.2.3 Lazer e Entretenimento	16
2.3 Interface Homem Máquina	17
2.3.1 Dispositivos Tradicionais	17
2.3.2 Dispositivos Assistivos Alternativos	17
2.3.3 Computadores Especiais	18
2.3.4 Desafios e Avanços	19
2.4 Teclados Virtuais Assistivos	19
2.5 Princípios de Acessibilidade na Web	37
2.5.1 Perceptividade	37
2.5.2 Operatividade	37
2.5.3 Compreensibilidade	38
2.5.4 Robustez	38
2.5.4 Velocidade	38
3. METODOLOGIA	39
3.1 Tipo de Pesquisa	39
3.2 Estrutura do Projeto	39
3.3 Desenvolvimento de Software Web (Site)	40
3.4 Conexão de Controladores	42
3.5 Sintetizador de Fala	45
3.6 Indicadores de Desempenho	45
3.7 Armazenamento de Feedbacks e Desempenho	47
3.8 Testes Automatizados	48
3.9 Predição de Palavras	49
4. RESULTADOS	51
5. DISCUSSÃO	62
5.1 Metodologia	62
5.2 Resultados	63
5.3 Comparativo com abordagens existentes.	65

5.4 Limitações	65
5.5 Trabalhos Futuros	66
6. CONCLUSÕES	66
7. CONTRIBUIÇÕES DO TRABALHO	67
7.1 Contribuições Tecnológicas	68
7.2 Contribuições Comunicacionais	68
7.3 Contribuições Assistivas	68
REFERÊNCIAS	69

1. INTRODUÇÃO

A interação com computadores, atualmente, é predominantemente mediada por dispositivos como o *mouse* e o teclado. Essas ferramentas são essenciais para a utilização de aplicações web em diversos contextos, incluindo relações sociais, trabalho, estudos e lazer. Segundo o relatório anual de 2022 do DATAREPORTAL (<https://datareportal.com/reports/digital-2022-global-overview-report>), 4,5 bilhões de pessoas estão conectadas à internet, representando 62,5% da população mundial (KEMP, 2022). No entanto, pessoas com deficiências motoras enfrentam barreiras significativas ao utilizar *desktops* e *notebooks*, principalmente devido ao *design* padronizado desses dispositivos.

Embora existam recursos de acessibilidade integrados nos sistemas operacionais, eles são frequentemente genéricos e exigem que os usuários se adaptem às suas limitações. A necessidade de tecnologia assistiva personalizada é evidente, pois indivíduos com deficiências motoras têm necessidades, desejos e preferências distintas (WESSELS et al., 2003). Muitos usuários acabam recorrendo a dispositivos alternativos, como *joysticks*, acionadores, teclados matriciais e controladores de *videogame*, que, apesar de mais adaptáveis, necessitam de *softwares* de mapeamento funcional, complicando ainda mais sua utilização eficaz (LYNDS, 2002).

Especificamente no caso dos teclados, uma abordagem comum é a simulação de um teclado físico de forma virtual. No entanto, essa prática enfrenta desafios significativos, como a baixa velocidade de digitação, que é drasticamente menor em comparação com teclados físicos, alcançando apenas uma média de 7 wpm contra 30 wpm nos convencionais (WARD & MACKAY, 2002; HOSTE & SIGNER, 2013). Nos últimos anos, ferramentas de predição textual têm sido incorporadas aos teclados virtuais para melhorar a eficiência da digitação, resultando em um aumento significativo na taxa de wpm (GARAY-VITORIA & ABASCAL, 2006).

Uma das soluções existentes é o teclado virtual Dasher, que oferece uma abordagem alternativa para a digitação por pessoas com deficiências motoras (WARD & MACKAY, 2002). No entanto, apesar de suas inovações, o Dasher apresenta limitações em termos de velocidade e precisão de digitação, bem como uma curva de aprendizado acentuada que pode desmotivar novos usuários, além disso, a interface do Dasher pode não ser intuitiva para todos, exigindo adaptação significativa dos usuários (WARD et al., 2000).

Neste trabalho, propõe-se revisitar este modelo, extrair as características positivas e suprir as negativas com novas ferramentas que se tem à disposição, utilizando uma abordagem preditiva e proposta de layout diferente em um ambiente web. Com isso, este estudo visa aumentar a eficiência da digitação para pessoas com deficiências motoras, facilitando sua interação com tecnologias computacionais.

1.1 PROBLEMA

Os teclados e os *mouses* são os dois principais dispositivos de entrada para a utilização de computadores *desktops* e *notebooks*. Desde suas primeiras versões, tiveram poucas mudanças em relação aos seus *layouts* e propósitos, tornando-se paradigmas consistentes no que se refere à interface homem-máquina (YASUOKA & YASUOKA, 2011). Este tipo de dispositivo apresenta médias aceitáveis com usuários treinados, atingindo até 30 wpm (WARD & MACKAY, 2002; Hoste & Signer, 2013), com variações de *layout*, sendo o QWERTY o mais conhecido, e mudanças em sua composição material.

Contudo, para pessoas com deficiências motoras, estes dispositivos apresentam um problema significativo, pois exigem movimentos complexos, localização precisa, teclas pequenas e força de pressão nos dedos (SZETO, 2015). Estas dificuldades resultam em performances muito inferiores às dos usuários sem essas deficiências, tornando-se um grave problema em diversos âmbitos, especialmente o social. A conexão a computadores pode ser crucial para este grupo, que pode ter condições mais agravantes que impossibilitem sua locomoção, onde a utilização desses dispositivos pode fornecer formas essenciais de interação social, lazer, estudos e trabalho.

Uma das soluções existentes é o teclado virtual Dasher, que oferece uma abordagem alternativa para a digitação por pessoas com deficiências motoras (WARD & MACKAY, 2002). No entanto, apesar de suas inovações, o Dasher apresenta limitações em termos de velocidade e precisão de digitação, bem como uma curva de aprendizado acentuada que pode desmotivar novos usuários, além disso, a interface do Dasher pode não ser intuitiva para todos, exigindo adaptação significativa dos usuários (WARD et al., 2000).

Diante desse contexto, a questão que norteia este estudo é: um teclado virtual dinâmico em ambiente web, com predição de texto, voltado para pessoas com deficiências motoras, pode aumentar a velocidade de digitação em comparação às outras

abordagens apresentadas na literatura, incluindo o Dasher? Este trabalho busca preencher essa lacuna ao desenvolver uma solução que seja mais intuitiva, rápida e precisa, facilitando a interação digital para este público.

1.2 JUSTIFICATIVA

Este problema vem sendo abordado ao longo dos anos; contudo, as soluções existentes geralmente não conseguem atingir boas performances ou, quando o fazem, apresentam diferentes níveis de aceitação pelo público. Por exemplo, o DASHER permite a digitação de até 23 wpm, adotando um conceito dinâmico, ao contrário dos teclados que utilizam uma abordagem estática (WARD & MACKAY, 2002). No entanto, esse conceito dinâmico pode tornar a interface complexa e menos intuitiva, especialmente para usuários com deficiências motoras. A complexidade em interfaces dinâmicas pode dificultar a usabilidade e a aceitação, especialmente quando essas interfaces não evoluem para atender aos padrões modernos de design e acessibilidade (LOWGREN & STOLTERMAN, 2004).

Portanto, é essencial desenvolver uma interface atualizada, de fácil acesso, em um ambiente web que não necessite de instalação prévia, como os demais *softwares*, além de buscar facilitar a digitação e a comunicação de pessoas com graves comprometimentos motores. Esse tipo de ambiente oferece a vantagem de receber atualizações automáticas, sem necessidade de qualquer ação do usuário. Além disso, a interface deve permitir ao usuário utilizar o dispositivo que melhor se adeque à sua deficiência motora para controlar o teclado virtual, e deve incluir o principal método para potencializar a digitação de textos: a predição de palavras.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

O objetivo geral desta dissertação é desenvolver um teclado virtual dinâmico em plataforma web com recursos que possibilitem uma taxa de digitação próxima à de um teclado convencional, permitindo ao usuário com deficiência motora conectar dispositivos via Bluetooth ou USB para sua utilização, possibilitando o usuário digitar mais rápido para facilitar seu desempenho em suas atividades de trabalho, lazer e de interação comunicativa.

1.3.2 Objetivos Específicos

1. Desenvolver meios de controle alternativos para o teclado virtual;
2. Implementar a funcionalidade de predição de palavras no teclado;
3. Integrar um recurso de síntese de fala para as palavras digitadas, para auxílio em comunicação;
4. Coletar feedbacks dos usuários sobre o software;
5. Estabelecer indicadores de desempenho do software;
6. Desenvolver testes automatizados para simulação do uso do software;
7. Validar a viabilidade do software com base nos testes automatizados para posterior distribuição e teste com usuários reais.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Tecnologia Assistiva

Atualmente, a forma mais simples de uma pessoa se inserir no meio social é por meio da utilização da internet e seus recursos. De fato, tornou-se o principal meio para propósitos variados como: redes sociais, trabalho, estudos e lazer conforme o apontado por discussões que ocorreram e ocorrem em portais e notícias pelo mundo como apresentado em discussão aberta do projeto da Secretaria de Assuntos Legislativos do Ministério da Justiça e Cidadania (PENSANDO O DIREITO, 2015). Conforme listado pelo relatório anual do DATAREPORTAL (<https://datareportal.com/reports/digital-2022-global-overview-report>) de 2022, 4,95 bilhões de pessoas estão conectadas à internet, o que representa 62,5% da população mundial conforme ilustrado na Figura 1 (KEMP, 2022).

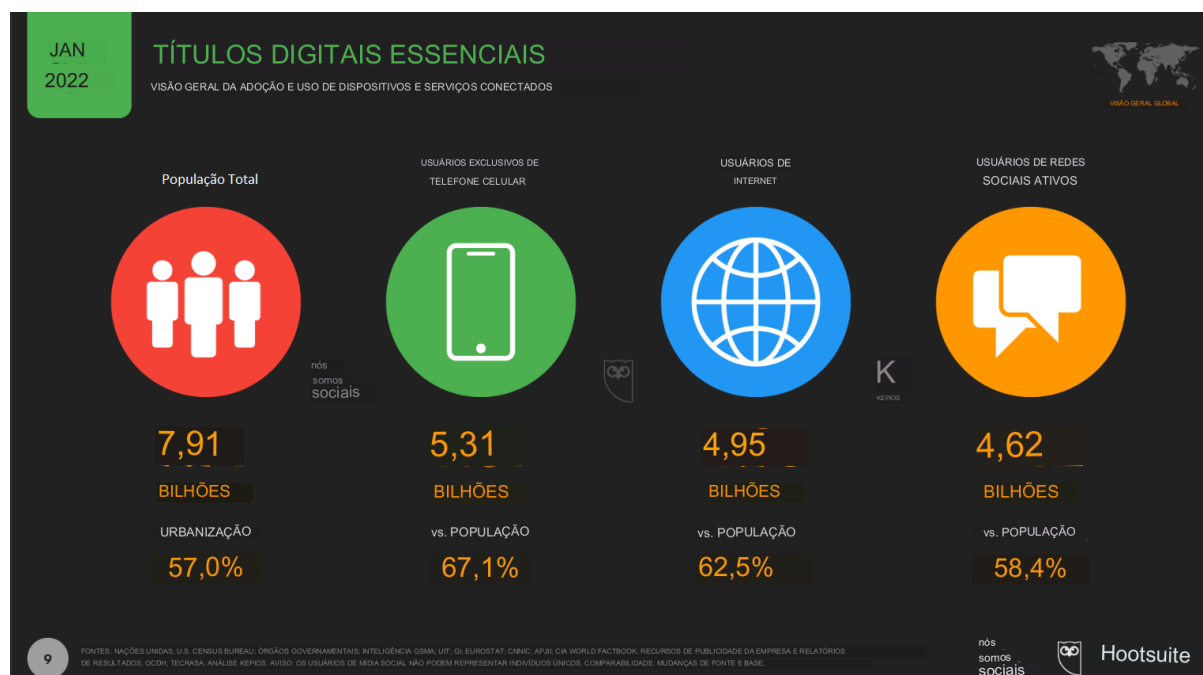


Figura 1 - Uso de internet pelas pessoas em contexto mundial em 2022.

Fonte: DATAREPORTAL (KEMP, 2022).

Pessoas com deficiências motoras têm dificuldade para utilizar dispositivos como computadores. Normalmente, buscam por tecnologias alternativas que melhorem sua adaptação a sua condição física específica. A tecnologia assistiva é aquela que se refere a qualquer tipo de dispositivo ou *software* que seja projetado especificamente para pessoas com algum tipo de deficiência ou limitação funcional, promovendo estratégias, facilidades e práticas para reduzi-las (COOK & HUSSEY, 2020). Para que essa tecnologia seja realmente útil às pessoas, ela deve buscar formas de ser individualizada, isto é, mesmo que um grupo de pessoas discorde de um mesmo

problema elas ainda têm sentimentos, desejos e preferências específicas. A literatura relata que grande parte do que é desenvolvido acaba sendo genérico e força o usuário a se adaptar à tecnologia quando o objetivo deveria ser justamente o contrário, este sendo um dos maiores paradigmas de tecnologia assistiva (WESSELS *et al.*, 2003).

2.2 Uso da Internet por Pessoas com Deficiências Motoras

A internet desempenha um papel essencial na inclusão digital de pessoas com deficiências motoras, permitindo o acesso a diversas oportunidades de trabalho, comunicação e lazer. Com o avanço das tecnologias assistivas, usuários com comprometimentos físicos significativos podem navegar e interagir com a web de maneira mais eficiente, promovendo sua integração em diversos contextos sociais e profissionais (KAYE *et al.*, 2011).

2.2.1 Oportunidades de Trabalho

A utilização da internet por pessoas com deficiências motoras tem se mostrado uma ferramenta valiosa para facilitar o ingresso e a permanência no mercado de trabalho. O trabalho remoto, por exemplo, tem se consolidado como uma alternativa inclusiva, oferecendo a flexibilidade necessária para que essas pessoas possam exercer suas atividades profissionais, superando barreiras físicas e de deslocamento (SCHUR *et al.*, 2013). Tecnologias de assistência, como teclados virtuais, sintetizadores de voz e softwares de controle por movimento ocular, tornam possível o uso de plataformas digitais por pessoas com limitações motoras, ampliando suas oportunidades de trabalho (LAZARRO & KLEINER, 2019).

2.2.2 Comunicação e Inclusão Social

A internet também proporciona meios de comunicação eficientes para pessoas com deficiências motoras, permitindo que elas mantenham conexões sociais por meio de redes sociais, videoconferências e fóruns de discussão. Tecnologias de comunicação aumentativa e alternativa (CAA), como o uso de sintetizadores de fala e teclados adaptados, facilitam a interação online, possibilitando que essas pessoas participem de atividades cotidianas de comunicação (COOK & POLGAR, 2015). Esse ambiente digital se torna um espaço crucial para a inclusão social, uma vez que muitas limitações motoras físicas impedem o contato social presencial (BARKER *et al.*, 2017).

2.2.3 Lazer e Entretenimento

Além das oportunidades de trabalho e comunicação, a internet proporciona inúmeras opções de lazer para pessoas com deficiências motoras. Jogos eletrônicos com

interfaces adaptadas, acessibilidade em plataformas de streaming e ferramentas de navegação por voz permitem que essas pessoas participem de atividades de entretenimento online (VANDERHEIDEN, 2000). O acesso a atividades recreativas, como assistir a vídeos, participar de comunidades virtuais e jogar videogames, contribui para uma melhor qualidade de vida e bem-estar psicológico (GARAY-VITORIA & ABASCAL, 2006).

2.3 Interface Homem Máquina

A interface homem-máquina (IHM) refere-se à conexão entre o usuário e o hardware ou software, utilizando dispositivos de entrada ou saída que permitem a interação com sistemas computacionais (PREECE et al., 2007). Tradicionalmente, essa interação ocorre por meio de teclados e mouses, mas para pessoas com deficiências motoras, o uso desses dispositivos apresenta desafios significativos, como a necessidade de movimentos finos e precisos, além de pressão constante nas teclas (SZETO, 2015).

2.3.1 Dispositivos Tradicionais

Mouses e teclados físicos ainda são os dispositivos de entrada mais comuns para a utilização de computadores. O modelo de teclado QWERTY, por exemplo, é amplamente difundido e utilizado globalmente. No entanto, para pessoas com comprometimentos motores, o uso desses dispositivos pode ser limitado ou impossível. A dificuldade de localização precisa do cursor no mouse ou a complexidade de pressionar teclas pequenas em um teclado convencional resultam em uma experiência frustrante para muitos usuários (HOSTE & SIGNER, 2013).

2.3.2 Dispositivos Assistivos Alternativos

A evolução das tecnologias assistivas tem proporcionado uma variedade de alternativas para substituir os dispositivos tradicionais de entrada, permitindo que pessoas com deficiências motoras acessem computadores de maneira mais eficiente. Entre os dispositivos mais utilizados estão:

Joysticks: Dispositivos adaptados de controle, muitas vezes derivados de controles de videogame, que permitem movimentos amplos e menos precisos, sendo uma solução eficaz para aqueles que têm dificuldade em usar o mouse. Estes controles têm se tornado populares devido à sua ergonomia e à possibilidade de adaptação para o controle de sistemas computacionais (LAZZARO & KLEINER, 2019).

Teclados Virtuais com Bluetooth: Esses teclados permitem que o usuário interaja com o computador sem a necessidade de teclas físicas, utilizando conexões sem fio e predição de texto para facilitar a digitação. A tecnologia Bluetooth tem se tornado cada vez mais presente nesse tipo de solução, facilitando a conectividade com uma ampla gama de dispositivos de entrada, como joysticks e controles adaptados (BIGHAM et al., 2017). Além disso, a predição de palavras contribui para aumentar a velocidade de digitação, reduzindo o esforço necessário para escrever (GARAY-VITORIA & ABASCAL, 2006).

2.3.3 Computadores Especiais

Soluções como o DARCI TOO exemplificam sistemas de computação projetados especificamente para pessoas com deficiência motora. O DARCI TOO é equipado com uma série de entradas alternativas, como botões grandes, switches e conexões para joysticks, permitindo a adaptação às necessidades específicas de cada usuário (LYNDS, 2002). Esses computadores permitem que pessoas com deficiências motoras graves interajam com sistemas computacionais de maneira eficaz, utilizando mapeamento de funções e controladores customizados.



Figura 2 - DARCI TOO é um computador com conexões para dispositivos alternativos.
Fonte: LYNDS, 2002.

2.3.4 Desafios e Avanços

Embora as tecnologias assistivas tenham evoluído significativamente, ainda existem desafios na interface homem-máquina para pessoas com deficiências motoras. A principal limitação é a adaptação dos sistemas a diferentes graus de mobilidade, que podem variar muito entre os usuários. A tendência crescente do uso de dispositivos sem fio, como os controladores Bluetooth, oferece maior flexibilidade e independência, permitindo que os usuários personalizem sua experiência de acordo com suas necessidades específicas (KAYE et al., 2011).

2.4 Teclados Virtuais Assistivos

De maneira geral, os dispositivos que mais se adequam às pessoas com deficiências motoras são os incomuns, que necessitam de mapeamento para substituírem o teclado ou o *mouse*. Especificamente para teclados, há o tipo virtual que, seguindo a definição de MOLINA et al (2009), é um tipo de teclado virtualizado por *software*, isto é, por meio de uma tela de computador apresenta-se um *layout* parecido com um teclado físico de modelo QWERTY, com o qual o usuário pode interagir e a máquina entende como entrada efetiva de um teclado físico. Este tipo de teclado torna-se interessante para pessoas que conseguem manipular dispositivos incomuns como *joysticks*, por exemplo, que não contemplam todas as letras do alfabeto, mas pelo seu ponteiro analógico podem interagir com o teclado virtual.

Contudo, apenas é resolvido o problema da utilização, mas ainda há as diferenças de desempenho entre os modelos real e virtual. WARD & MACKAY (2002) e HOST & SIGNER (2013) verificaram que um teclado comum pode produzir em média mais de 30 wpm, em comparação aos teclados virtuais assistivos que podem produzir em média 4-7 wpm.

Esses teclados podem ter sua performance melhorada quando aplicados algoritmos de predição de palavras. Conforme descrito por SAEED et al. (2022), existem métodos estáticos e por aprendizagem de rede neural profunda. No método estático, normalmente, usa-se um dicionário no qual, ao entrar com nova letra, este é consultado para predizer qual é a próxima letra da sequência, ou a próxima palavra, conforme o exemplificado por Matthew (1996). Já o método de aprendizado profundo parte do princípio de que o algoritmo se comporta como o cérebro humano, processando os dados e predizendo a próxima palavra de forma a simular o comportamento do mesmo, normalmente chamada de rede neural profunda

(DARGAN *et al.*, 2019). No trabalho de GARAY-VITORIA & ABASCAL (2006) foi medido o desempenho de pesquisa de um dos maiores *sites* de pesquisa, o Google. Basicamente, esse sistema de pesquisa visa abrir uma caixa de sugestão de N palavras, conforme o que é digitado pelo usuário. Observou-se que a wpm recebeu um acréscimo entre 5,3~5,6 wpm. Isto significa que adotando um padrão de 30 wpm, chegar-se-ia a 35,6 wpm. Alguns problemas relacionados à predição são listados na literatura como fatores que envolvem previsões de longo prazo, pois é necessário um contexto do determinado texto, podendo necessitar de mais recursos de processamento, conforme apontado no trabalho de SAEED *et al.* (2022). Efetivamente, para a previsão se manter rápida e coesa.

Em trabalhos relatados do projeto AmpliSoft, voltados ao auxílio à comunicação de pessoas portadoras de necessidades especiais, especificamente no trabalho de JORDAN *et al.* (2020), é relatado até 61% de eficiência na digitação de textos em testes realizados por crianças com diferentes perfis cognitivos. Baseado no Modelo Oculto de Markov que é um modelo estatístico em que o sistema modelado é assumido como um processo de Markov com parâmetros desconhecidos, e o desafio é determinar os parâmetros ocultos a partir dos parâmetros observáveis. Os parâmetros extraídos do modelo podem então ser usados para realizar novas análises, por exemplo para aplicações de reconhecimento de padrões.

Em um modelo regular de Markov, o estado é diretamente visível ao observador, e portanto os únicos parâmetros usados são as probabilidades de transição de estado. Cada estado possui uma distribuição de probabilidade sobre os possíveis resultados. Esse tipo de modelo é conhecido por sua aplicação na área de reconhecimento de padrões temporais como a fala, a escrita, os gestos e a bioinformática. Neste projeto, foi coletado pelos autores 40329 palavras selecionadas com base em textos infantis, ou seja, uma coleção de dados estática, essas palavras foram devidamente marcadas com *tags* que as classificavam. Por meio destas palavras e *tags*, um algoritmo de predição foi aplicado para os seguintes propósitos: “previsão das palavras de abertura em uma frase”, “prevendo a próxima palavra de uma palavra conhecida”, “procurando por uma palavra enquanto digita”, “pesquisando a palavra digitada pelo usuário”. Para os testes, os autores buscaram a partir da utilização de crianças contabilizar necessidade de cliques e tempo de execução a partir de quatro grupos, sendo um deste de controle que ficou responsável pela utilização sem predição, posteriormente os resultados foram comparados entre os grupos e determinar o percentual de acurácia do modelo preditivo. A Figura 3 apresentada no trabalho dos autores mostra

para os usuários dos grupos a diferença entre os tempos e quantidade de cliques entre os modelos com e sem predição.

Table 1

Performance of software prediction

Volunteer	No prediction		Prediction	
	click	time(s)	click	time(s)
V1	258	732	189	812
V2	233	743	182	850
V3	96	-	77	-
V4	256	939	210	907
V5	318	1747	-	-
V6	283	883	205	919
V7	295	1733	-	-
V8	264	1019	185	781
V9	259	733	188	757
V10	282	1189	238	1079
V11	273	677	156	438
V12	294	1360	-	-
V13	315	1104	250	834
V14	251	730	232	544
V15	326	950	273	850
V16	270	992	222	730
V17	220	623	199	595
V18	300	585	175	470
V19	305	850	173	840
V20	279	590	220	685
V21	237	1040	175	881
V22	242	610	227	710
V23	291	751	199	751
V24	304	1020	235	794
V25	248	840	230	963
V26	292	1175	224	1373
V27	287	951	273	830
V28	261	677	205	731
V29	272	950	256	1020
V30	293	669	186	735
V31	265	1500	-	-
V32	324	853	190	865
V33	253	840	163	761
V34	254	1050	192	997
V35	314	5648	169	3959
V36	304	5216	215	3929
V37	343	4275	167	2394
V38	297	1864	245	2744
V39	388	3155	260	2374
V40	298	4733	195	1420
V41	372	5620	258	3781

Figura 3 – Diferença em tempos relativos à utilização do teclado virtual com predições.

Fonte: JORDAN *et al.*, 2020.

Os autores concluem sobre o principal ganho do estudo, que efetivamente está no algoritmo de predição textual tal qual indicaram redução no número de cliques em até 51,3% (média de 25,5%) que significa até 61% de eficiência na digitação de textos, tal qual poderia ser aplicado a pessoas com deficiência motora onde cada clique pode ter um *delay* maior quando comparado a pessoas sem deficiência motora.

Em 2002, foi proposto um teclado virtual que se difere dos teclados modelo QWERTY, denominado DASHER, ilustrado na Figura 4 (WARD & MACKAY, 2002). Este teclado é baseado no conceito de predição probabilística, projetado para digitação rápida e eficiente por meio de um único botão. Diferentemente dos outros teclados virtuais, ele

não tenta simular graficamente um teclado físico, sendo dinâmico, onde o mesmo se movimenta para maximizar a eficiência na digitação de palavras. Os principais meios de funcionamento deste teclado envolvem as seguintes funcionalidades:

1. Visualização: interface de texto em que uma única área de exibição de texto é mostrada conforme Figura 4;
2. Entrada por deslizamento: o principal método de entrada do Dasher é o deslizamento do dedo ou controle ocular, cada letra fica dentro de um retângulo colorido. O usuário "retira" as letras da direita e arrasta-as para a esquerda, onde as palavras vão sendo montadas, à medida que o usuário escolhe uma letra, um sub-alfabeto já se forma do lado direito. Esse sub-alfabeto é formado pelas letras mais prováveis a serem utilizadas a seguir;
3. Predição inteligente: à medida que o usuário desliza o dedo sobre a área de exibição, o Dasher usa um algoritmo preditivo para tentar antecipar qual caractere o usuário está tentando digitar;
4. Sugestões de palavras: à medida que o usuário continua a deslizar o dedo, o Dasher exibe uma lista de palavras possíveis que o usuário pode estar tentando digitar;
5. Espaços e pontuação: para inserir espaços entre as palavras ou inserir pontuação, o usuário pode usar gestos específicos;

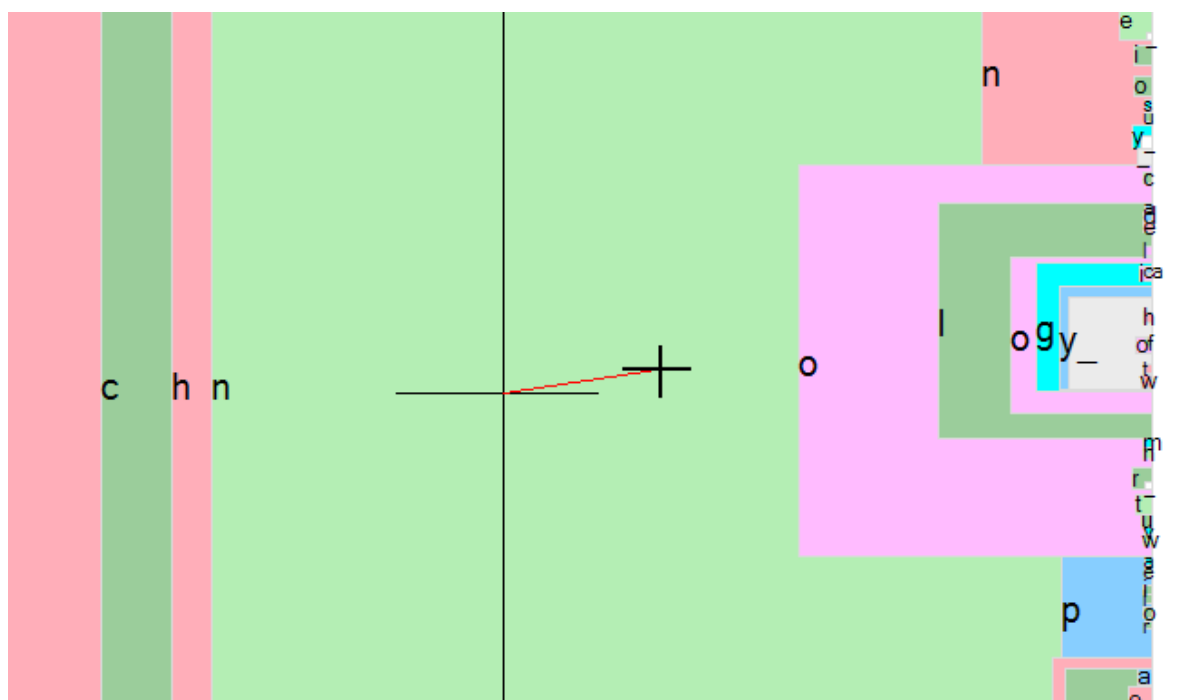


Figura 4 – Layout teclado virtual DASHER.

Fonte: WARD; MACKAY, 2002.

É relatado que utilizando dispositivos como *mouses* ou alternativos é possível digitar até 25 wpm para usuários treinados; já para dispositivos como rastreador de olhos é

possível digitar até 15 wpm (WARD & MACKAY, 2002). Este teclado apresenta uma nova abordagem e resultados interessantes quando comparados com os teclados físicos, porém a complexidade e a habilidade requeridas para utilizar eficazmente esse teclado podem impactar a velocidade de digitação, destacando a necessidade de um desenvolvimento cognitivo robusto para alcançar níveis de eficiência semelhantes a um teclado convencional.

WELTON et al. (2016) conduziram um estudo inovador que explorou o uso do Dasher combinado com uma interface cérebro-computador (BCI), permitindo que pessoas com deficiências motoras severas digitassem utilizando apenas sinais cerebrais. O estudo demonstrou que, apesar das limitações em termos de velocidade de digitação (média de 5,7 wpm), a interface conseguiu proporcionar uma experiência de uso funcional para pessoas com deficiência motora grave, permitindo que interajam com o sistema sem a necessidade de dispositivos físicos de entrada. A pesquisa destacou que o uso da BCI no Dasher poderia ser ampliado para melhorar a eficiência por meio de melhorias nos algoritmos de predição (WELTON et al., 2016).

Outro estudo relevante foi realizado por POLACEK et al. (2017), que compararam diferentes métodos de entrada de texto, incluindo o Dasher, com o foco em usuários com deficiências motoras. O estudo observou que, apesar da curva de aprendizado mais acentuada no uso do Dasher, usuários com deficiências motoras leves a moderadas conseguiram alcançar velocidades de até 15 wpm após períodos de prática. Além disso, os pesquisadores destacaram a importância da personalização do sistema, sugerindo que ajustes no *layout* e na predição poderiam melhorar significativamente a experiência do usuário (POLACEK et al., 2017). O estudo concluiu que o Dasher, embora inovador, ainda enfrenta desafios em termos de acessibilidade universal, particularmente em usuários com deficiências mais graves.

Um dos principais fatores que colaboram para a performance dos teclados na literatura seriam como as letras são organizadas nos teclados virtuais ou físicos “a organização das teclas e símbolos deve ser feita de forma a maximizar o desempenho e minimizar o esforço de digitação” (GHOSH, 2011). A lei de Fitts é um dos modelos que ajudam a medir o tempo de movimento com base na distância entre as teclas, podendo ser usado para otimizar o tempo de digitação (FITTS, 1954). No que se refere à medição de digitação de teclas são listadas seis possíveis formas: “palavras por minuto (WPM), caracteres por minuto (CPM), gestos por caractere (GPC),

ativação de tecla por caractere (KSPC), taxa total de erros (TER), distância da sequência de erros (MSD).” conforme GOMIDE *et al.* (2016). A medida mais utilizada normalmente é o número de palavras por minuto conforme apontado por WOBBROCK (2007). O cálculo pode ser feito de algumas formas diferentes, mas basicamente a variável chave é saber com quantas letras considerar uma palavra formada, visto que dependendo desta o valor por minuto pode se alterar; normalmente, é considerada a média de 5 letras (NICOLAU *et al.*, 2013). Ainda no trabalho de GOMIDE *et al.* (2016), após estudo de mais de 250 publicações relacionadas a teclados virtuais, eles propõem uma nova abordagem de teclado virtual conforme ilustrado na Figura 5, a partir dos principais pontos encontrados:

1. presença de caracteres especiais;
2. *feedbacks* visuais e audíveis;
3. navegação linear;
4. seleção dinâmica;
5. método de escaneamento automático;
6. posição das letras circular e matricial;
7. sequência de letras de forma lógica;
8. presença de ambiguidade;

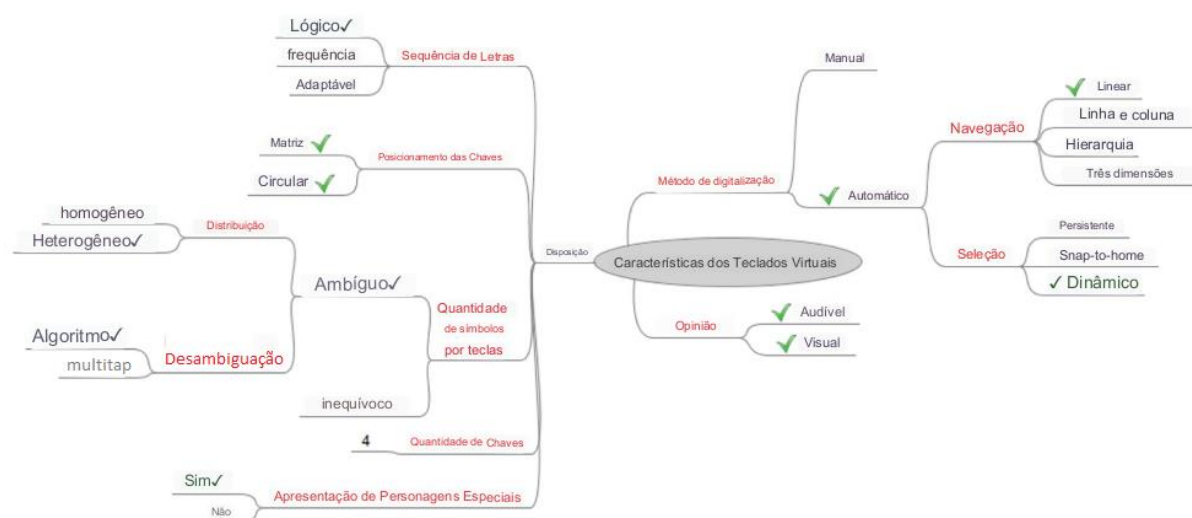


Figura 5 - Pontos levantados que um teclado virtual deve conter para prover acessibilidade.

Fonte: GOMIDE *et al.*, 2016.

Basicamente, em um teclado de varredura linear como ilustrado na Figura 6, cada letra é pré-selecionada e o usuário confirma a seleção por meio de um acionador. Os inventores propõem botões de ações específicos para determinadas necessidades como: correção textual, nova palavra, espaço, teclas especiais. Eles assinalam que as teclas especiais tem uma tela específica em varredura para sua seleção, assim

não atrapalhando o *layout* das letras principais. Algo interessante pontuado neste trabalho também é relativo à língua portuguesa e como seria a sua melhor distribuição: “Identificamos que a distribuição das letras nas palavras do português segue um padrão. Oitenta e cinco por cento das palavras do dicionário neste idioma são compostas por uma sequência de vogais e consoantes intercaladas. Portanto, decidimos agrupar as vogais em uma única tonalidade e distribuir alfabeticamente as consoantes nas tonalidades restantes.” (GOMIDE *et al.*, 2016). O trabalho deles busca da literatura um grande agrupamento relativo aos teclados virtuais, além de propor um modelo ideal; contudo, os resultados do modelo proposto por eles ainda não foram apresentados. Também não foi levado em conta o teclado DASHER de WARD & MACKAY (2002), que mostra resultados pertinentes relativos ao indicador de wpm que uma pessoa com deficiência motora consegue digitar com diferentes tipos de acionadores.

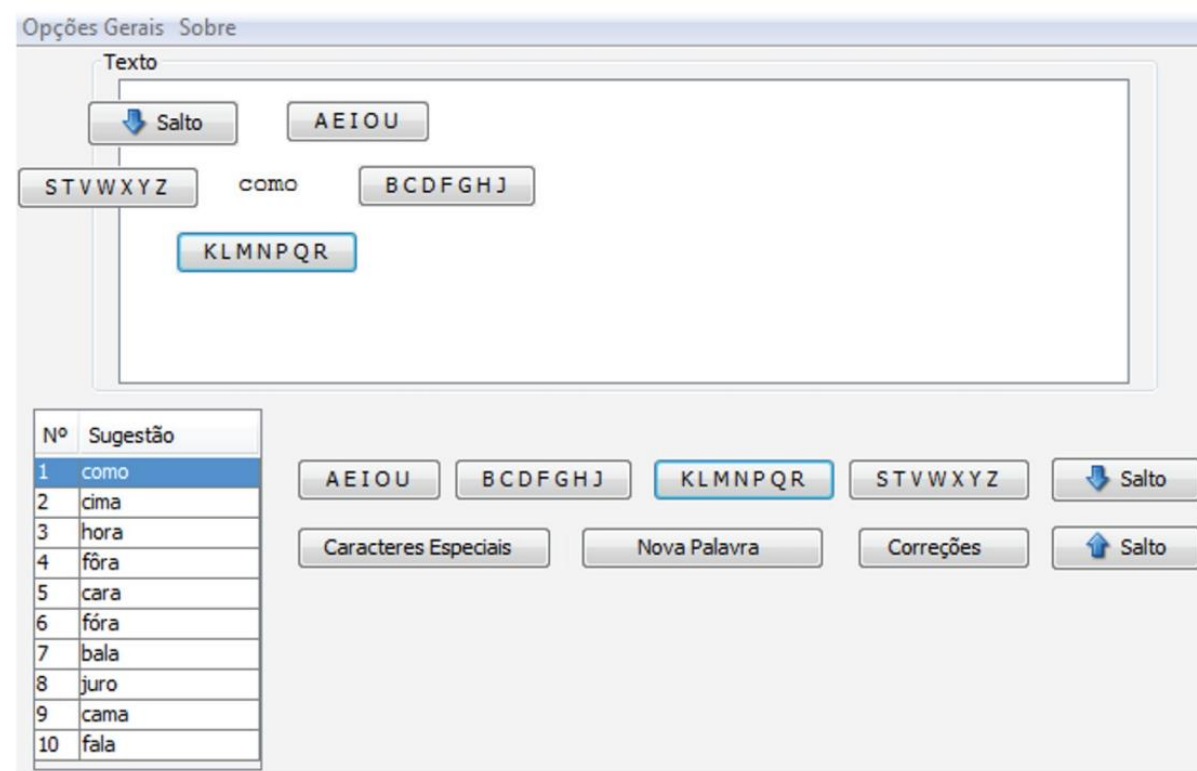


Figura 6 - Layout de um teclado virtual ideal após uma revisão de literatura.

Fonte: GOMIDE *et al.*, 2016.

Em 2016, foi proposto um teclado virtual denominado “ETM Web” voltado para usuários com paralisia cerebral, originalmente instalável no sistema operacional; porém, com a versão atualizada em ambiente web, como ilustrado na Figura 7. Este teclado pode operar em modo varredura, assim como de outras formas alternativas de controlar, sendo a principal delas o *joystick* (HENZEN & NOHAMA, 2016). Basicamente, a forma de varredura funciona com as teclas em destaque por determinado intervalo de tempo, onde o usuário, por meio de um acionador único,

dispara a ação de seleção. Este tipo de teclado apresenta determinadas limitações quando se refere a velocidade de seleção (wpm), sendo necessário passar por todas as teclas dependendo de qual seja o alvo do usuário. Também é possível que o usuário perca o tempo de seleção, duplicando o tempo necessário para digitação da letra.

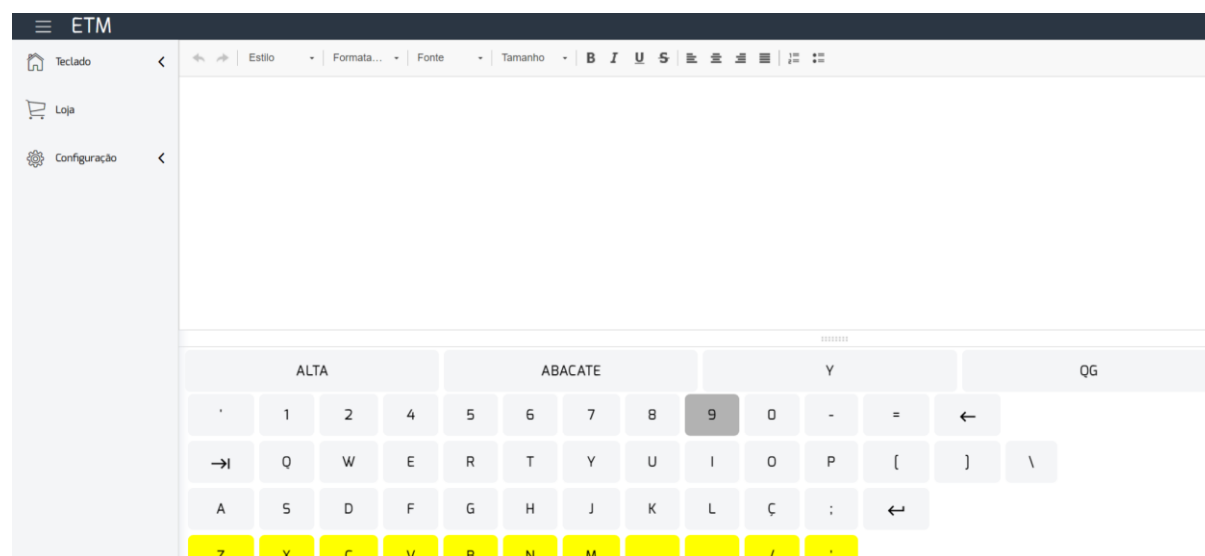


Figura 7 - Software web ETM.

Fonte: Autor.

Como explicado pelos autores, este é focado primeiramente para pessoas com paralisia cerebral tais quais apresentam mais dificuldades relacionadas ao controle motor sendo ideal o *software* por ativação única. É possível que os usuários customizem da forma que preferirem os *layouts* dos teclados, sendo possível priorizar letras mais digitadas conforme ilustrado na Figura 8. Também é possível que o *layout* seja disponibilizado para outros usuários, sendo uma forma interessante de união entre os usuários do *software*, e perspectiva da ferramenta que poderia ser explorada em âmbito social.

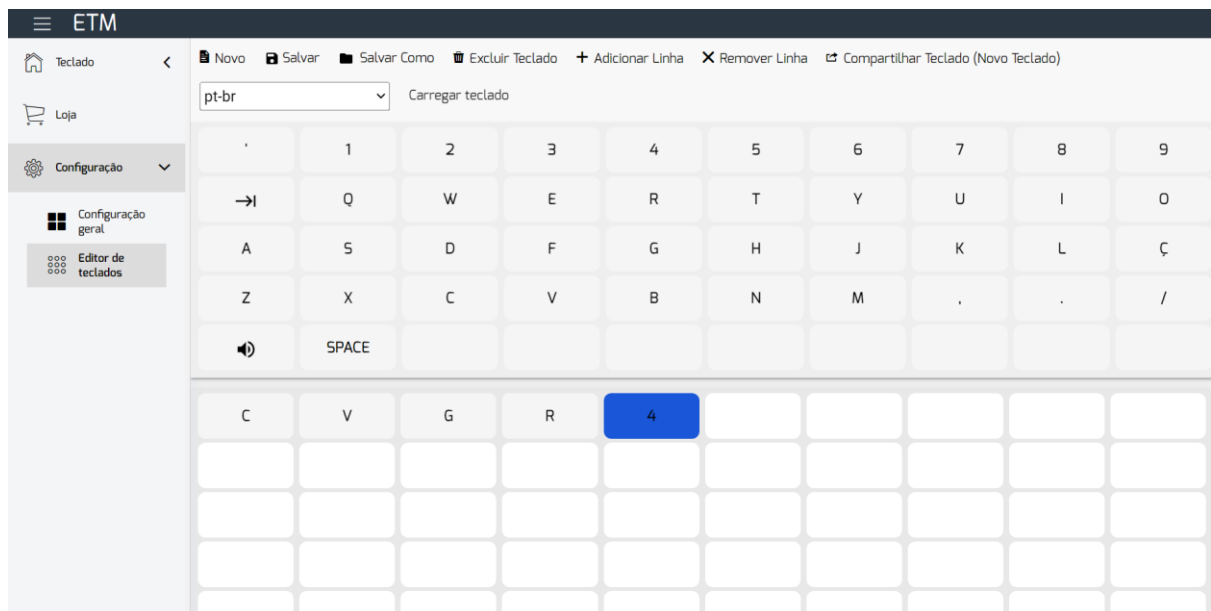


Figura 8 - Software web ETM edição de *layout* de teclado.

Fonte: Autor.

Um dos métodos alternativos ao controle via *joystick* é o controle ocular, o qual se aplica, principalmente às pessoas que apresentam graus avançados de deficiência motora, em trabalhos recentes é proposto um teclado virtual que contempla prioritariamente este tipo de controle, além de um *layout* adequado. Este *layout*, apresentado no trabalho de CECOTTI *et al.* (2019), tem como foco principal organizar os símbolos possíveis de digitação em *layouts* menores, conforme ilustrado na Figura 9. O aplicativo suporta três línguas distintas: Latim, Bangla e Devanagari. Por meio de foco do olhar, o usuário pode se localizar em uma das seções possíveis onde o símbolo desejado está contido, até chegar nos menores níveis e selecionar apenas uma letra ou palavra.

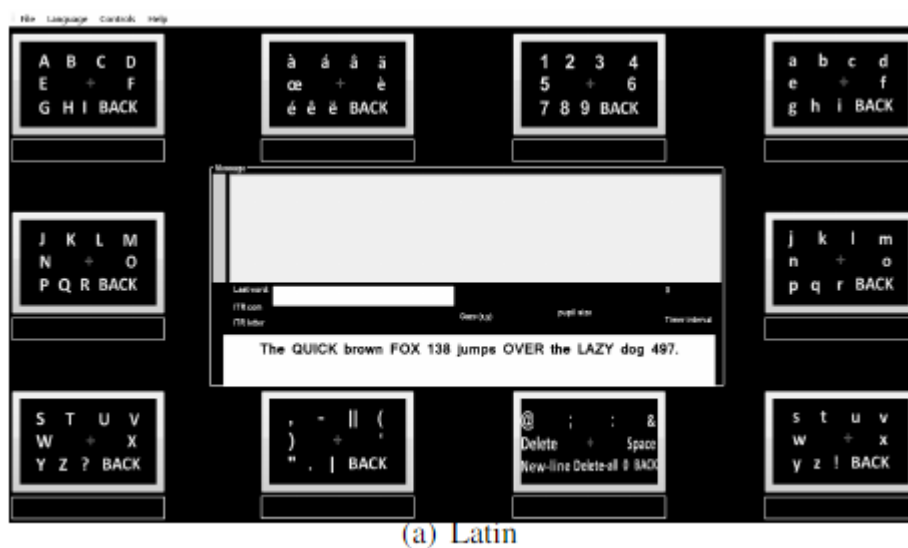


Figura 9 - Interface gráfica teclado virtual multilingual.

Fonte: CECOTTI *et al.*, 2019.

Eles apresentam resultados relativos à taxa de transferência de informação (ITR), que representa uma medida usada para avaliar a eficiência dos sistemas de comunicação. É ressaltado que não são utilizadas métricas de palavras por minutos entre os teclados, pois cada linguagem apresenta um número diferente de símbolos, e a quantidade deste pode variar consideravelmente para produzir uma mesma palavra. Apresentam o ITR no nível dos símbolos $57,61 \pm 14,14$ e $50,45 \pm 13,62$ bits/min para latim e bangla, enquanto está em $57,06 \pm 22,06$ e $45,19 \pm 15,34$ bits/min para latim e devanagari. Conclui-se que o *layout* latino apresenta melhores performances, demonstrando que o *layout* também deve ser pensado em relação a como as palavras podem ser representadas por meio dos símbolos presentes na língua em que foi desenvolvido, sendo necessárias possíveis adaptações de *layout* dependendo da língua.

Um teclado virtual também dirigido por controle ocular foi proposto especificamente para a linguagem portuguesa brasileira, onde é adotado pelos autores uma forma diferente de trabalhar com este tipo de controle, não requerendo a fixação do olhar em uma letra por vez, isto é, realizando todo caminho dirigido pelo olho e contabilizando os símbolos deste caminho (SILVA & VEIGA, 2020). Este tipo de abordagem pode ocasionar em erros de digitação das palavras, é proposto pelos autores que ao final da digitação o algoritmo liste as palavras aproximadas a digitada com possivelmente erro para seleção do usuário. O método adotado pelos autores baseia-se no cálculo de distância euclidiana, buscando efetivamente distância entre os conjuntos, no qual, por intermédio dessa distância e variações de trajeto, é possível determinar pontos do trajeto que, por sua vez, indicam os símbolos selecionados. Na Figura 10, ilustra-se a ideia de traçar os pontos com base nas direções observadas pelo usuário.

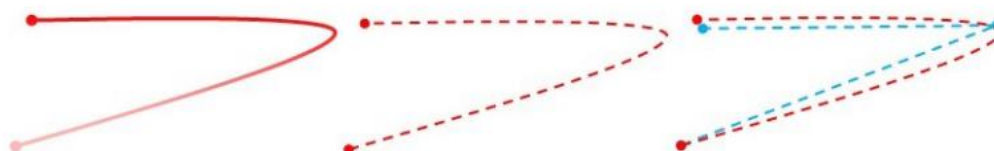
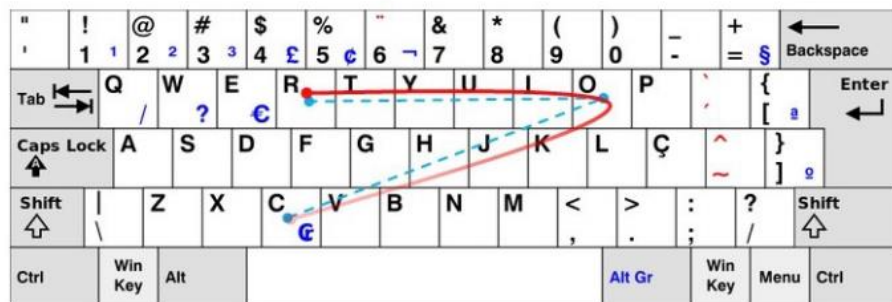


Figura 10 - Rastreamento de símbolos por rastreamento ocular contínuo.

Fonte: SILVA; VEIGA, 2020.

A partir dos testes com indivíduos adultos sem presença de problemas de visão, os autores trazem resultados da digitação de palavras pré-selecionadas, comparando o desempenho ocular e de um *mouse* comum, apresenta eficiência acima de 90% com o *mouse*, enquanto o rastreador ocular apresenta 50% de eficiência. Também são apresentados os resultados com relação ao tempo de digitação, onde termos frequentemente utilizados como “para”, “com”, “segundo” e “de” apresentam médias de tempos inferiores a 2 s. A abordagem adotada poderia ser melhor explorada por meio de *layouts* diferenciados do teclado QWERTY. Os autores citam essas limitações em suas conclusões.

Ainda em trabalhos recentes relacionados à ativação de teclados virtuais por rastreamento ocular e piscar dos olhos, existem abordagens que visam o total controle ocular, onde o acionador básico é o piscar do olho dirigente conforme trabalho apresentado por ISLAM *et al.* (2021). Para tal, é necessário, além do rastreamento ocular para controle horizontal da seleção do caractere, a detecção de fechamento do olho; neste trabalho, os autores abordam a relação de dois eixos, vertical e horizontal, traçados sobre o olho, onde o eixo vertical é aquele que determinará o fechamento do olho, pois é onde ocorre a variação conforme movimento das pálpebras. O rastreamento de posição ocorre pela esclera (região branca do olho) e por meio de algoritmos que processam em que posição está deslocada, realizando movimento inversamente proporcional no teclado virtual proposto é ilustrado na Figura 11.

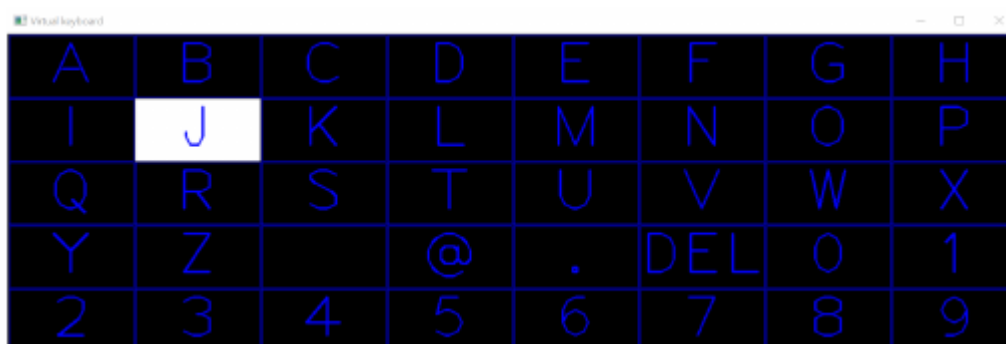


Figura 11 - Interface gráfica teclado virtual controlado por rastreamento ocular e piscar dos olhos.

Fonte: ISLAM *et al.*, 2021.

O teclado virtual pelo controle ocular total traz resultados relativos à velocidade de digitação com base em textos simplificados como “Hi There” em testes controlados, onde a melhor média apresentada é de até 2,35 wpm após algum treinamento. Os autores não abordam possíveis falhas relativas ao controle pelo piscar dos olhos, apenas sobre a métrica final da velocidade de digitação, com a qual não é possível entender a distinção entre comando incorreto ou apenas uma confusão linguística do usuário.

Algumas abordagens avançadas de detecção por meio de câmeras também são utilizadas na interface cérebro-computador, especialmente em relação ao eletrooculograma (EOG). No trabalho de DHILLON *et al.* (2009) utilizando uma técnica que a partir de eletrodos colocados em posições ao redor dos olhos do usuário, é possível detectar o movimento dos olhos, conforme ilustra a Figura 12.

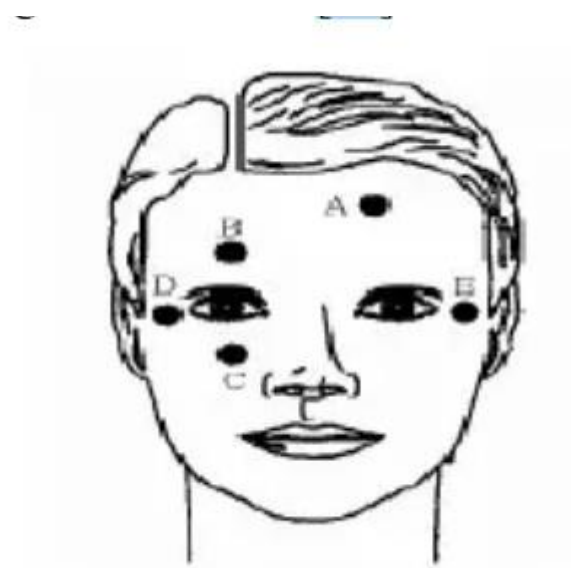


Figura 12 - Eletrodos aliados na face do usuário para captura dos sinais elétricos.

Fonte: DHILLON *et al.*, 2009.

A partir desta técnica, eles propõem o controle de um teclado virtual, onde o posicionamento no teclado se refere à detecção do movimento por meio da captura do sinal de EOG. Não é apresentada medição de eficiência desta técnica pelos autores, mas é possível inferir que este controle pode ser mais preciso que a interpretação de imagens de outros trabalhos relatados, pois fatores externos como luminosidade não produziram interferência na utilização do teclado.

Outras abordagens relatadas na literatura utilizam-se do eletroencefalograma, ao qual o princípio de coleta de sinais elétricos provenientes de atividade cerebral é aplicado. No projeto de MAYAUD *et al.* (2013), propõe-se um sistema denominado RoBIK, que realiza controle a partir de um *headset* composto por diversos eletrodos, como ilustrado na Figura 13.

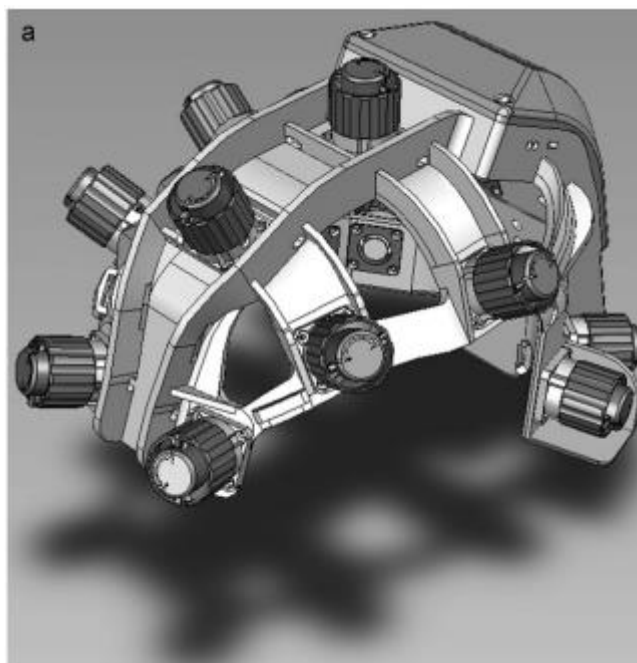


Figura 13 - *headset* composto por eletrodos para captura de sinais elétricos.

Fonte: MAYAUD *et al.*, 2013.

Diferentemente de outros projetos, o princípio de controle de um teclado virtual seria utilizando um *browser* Brainmium, que é um navegador da web autônomo no qual se navega em uma página da web a partir de uma estrutura de *software* que permite a execução e desenvolvimento de aplicações web baseadas na resposta do P300 e contempla paradigmas citados por CONGEDO *et al.* (2011) para realizar a seleção de elementos do *browser*. Os autores não fornecem dados relativos ao desempenho de digitação no *browser* utilizando o dispositivo de controle, que parece ser adequado para casos severos de paralisia cerebral.

Projetos mais recentes também voltados ao controle a partir do eletroencefalograma contemplam resultados relativos a desempenhos de digitação. GANNOUNI *et al.* (2022) propõem controlar as funcionalidades do sistema utilizando sinais de eletroencefalografia. A abordagem adotada neste trabalho também baseia-se na resposta do P300. Esta abordagem trata da exposição dos olhos a um estímulo externo que gera um potencial evocado na área cortical do cérebro, como o P300. Este tipo de atividade normalmente conta com um *delay* de 300 ms após a ocorrência dos estímulos visuais externos. Neste projeto, os autores propõem formas de controlar todo computador em si por meio da captura desses biopotenciais, e abordam sobre a digitação de comandos de *prompt*, ilustrada na Figura 14.

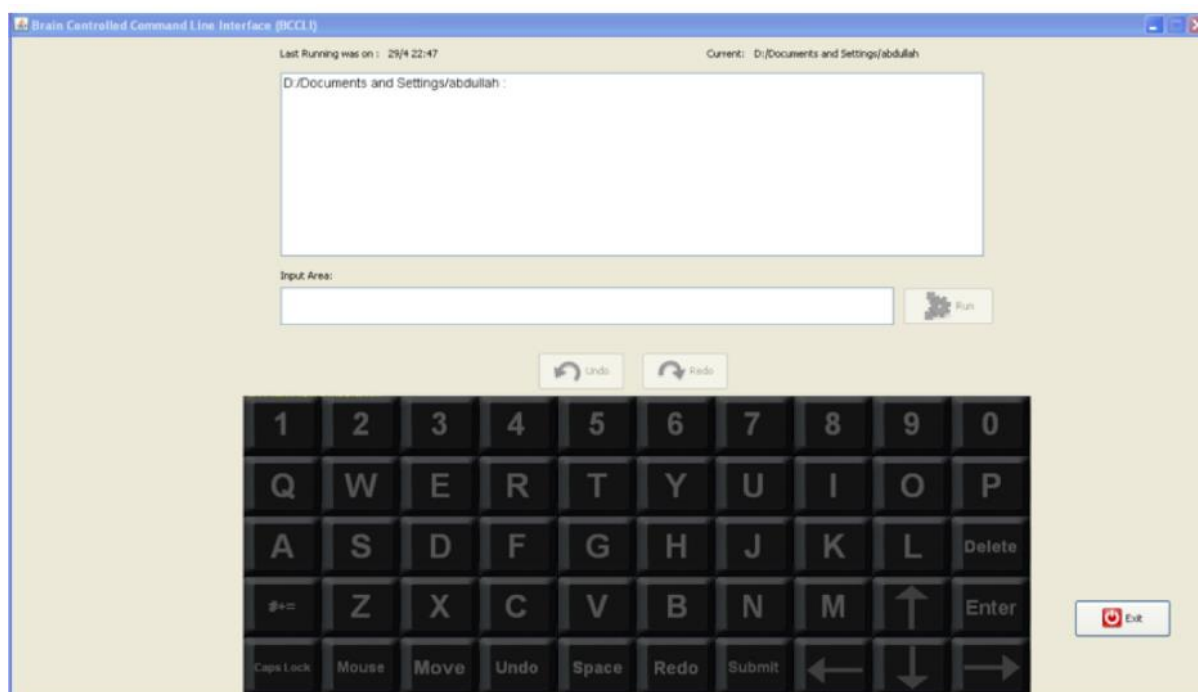


Figura 14 - Interface teclado virtual por comandos de *prompt*.

Fonte: GANNOUNI *et al.* (2022).

Os métodos utilizados pelos autores referem-se à utilização das APIs expostas do sistema operacional para desenvolvimento, como *KeyboardController* e *Command Line API*. Os resultados expostos pelos autores referem-se majoritariamente em relação à precisão do sistema, visto que isto ainda é ponto crucial nos sistemas que funcionam a partir da eletroencefalografia; contudo, é exposto que referente às funcionalidades de controle do computador focados da entrada textual, alcançou-se precisão média de 85% no modo de ortografia livre, que permite ao participante escrever 21 caracteres sem erros por minuto. Considerando uma média de 5 caracteres por palavra, isso representaria, em média, 4 wpm.

Controles alternativos aos *joysticks*, dispositivos oculares ou sinais elétricos, também pode ser utilizado o movimento da língua como efector principal. STRUIJK (2006) propôs uma interface que se utiliza do método de detecção baseado na lei de indução de Faraday para uma bobina e usa técnicas de indutância variável. A ideia é mudar a indutância de uma bobina de indução com núcleo de ar, movendo um material ferromagnético, preso à língua, para dentro do núcleo das bobinas, conforme ilustrado na Figura 15.

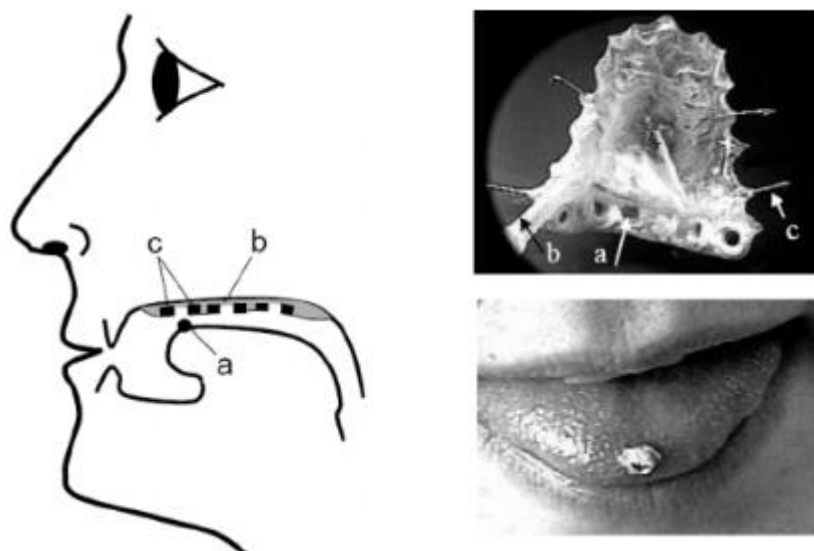


Figura 15 – Controle teclado virtual a partir da língua através de indutância variável.

Fonte: STRUIJK (2006, p.2594-2597).

Em uma configuração experimental, o autor propõe uma exibição visual mostrando o texto a ser digitado (branco) e o texto sendo digitado (preto). A exibição visual inclui uma figura da placa palatina com indicação dos caracteres relacionados aos 5 sensores. Uma corrente é aplicada às bobinas, que são conectadas em série na placa palatina. O sinal resultante é retificado, amplificado e amostrado. Intervalos de dados de 0,1 s são calculados e usados para delimitação após a remoção da linha de base para determinar se uma seleção é feita pelo sujeito. Um caractere selecionado aparece na exibição visual imediatamente após a detecção. A Figura 16 representa o fluxo completo até a digitação de um caractere.

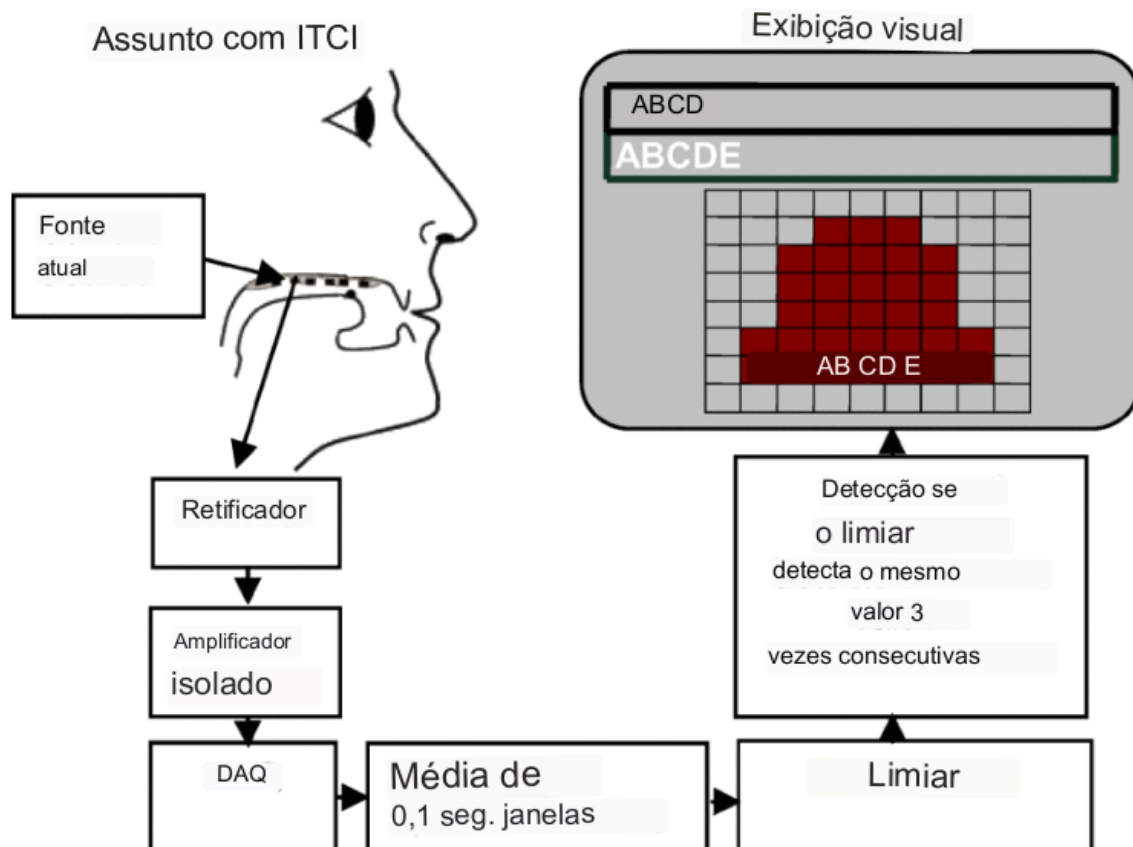


Figura 16 - Fluxograma até a digitação de um caractere por entrada realizada por controle bucal.

Fonte: STRUIJK (2006).

São apresentados resultados pertinentes a utilização do sistema por um usuário não treinado pelos autores, onde a escolha de um caractere tinha tempo total máximo de 2.9s levando em conta tempo de *delay* do algoritmo e o número de símbolos utilizados. Complementa que existem limitações em um sistema desse tipo, pois o espaço bucal é limitado, mas que com pequenas bobinas é possível abranger todo alfabeto latino. Porém, não são expostos resultados com todos os símbolos, nem algum tipo de comparação com métodos oculares obtidos com os usuários alvos deste sistema que apresentam deficiências motoras extremas, tais como paralisia cerebral ou esclerose múltipla.

Algumas interfaces de controle também são propostas para casos menos severos de origem neuromuscular, onde o propósito é fornecer uma simplificação aos teclados QWERTY tradicionais. ANDREAS *et al.* (2022) abordam a implementação de um controlador simplificado ilustrado na Figura 17, onde, por meio de dez acionadores controlados cada um por um dedo equivalente da pessoa, é possível realizar as entradas de todos os caracteres presentes no teclado. Desta maneira, o usuário não

necessita de movimentação de deslocamento do braço/mão, conseguindo realizar diversas ações por movimentos de pressão simples ou pressão longa.

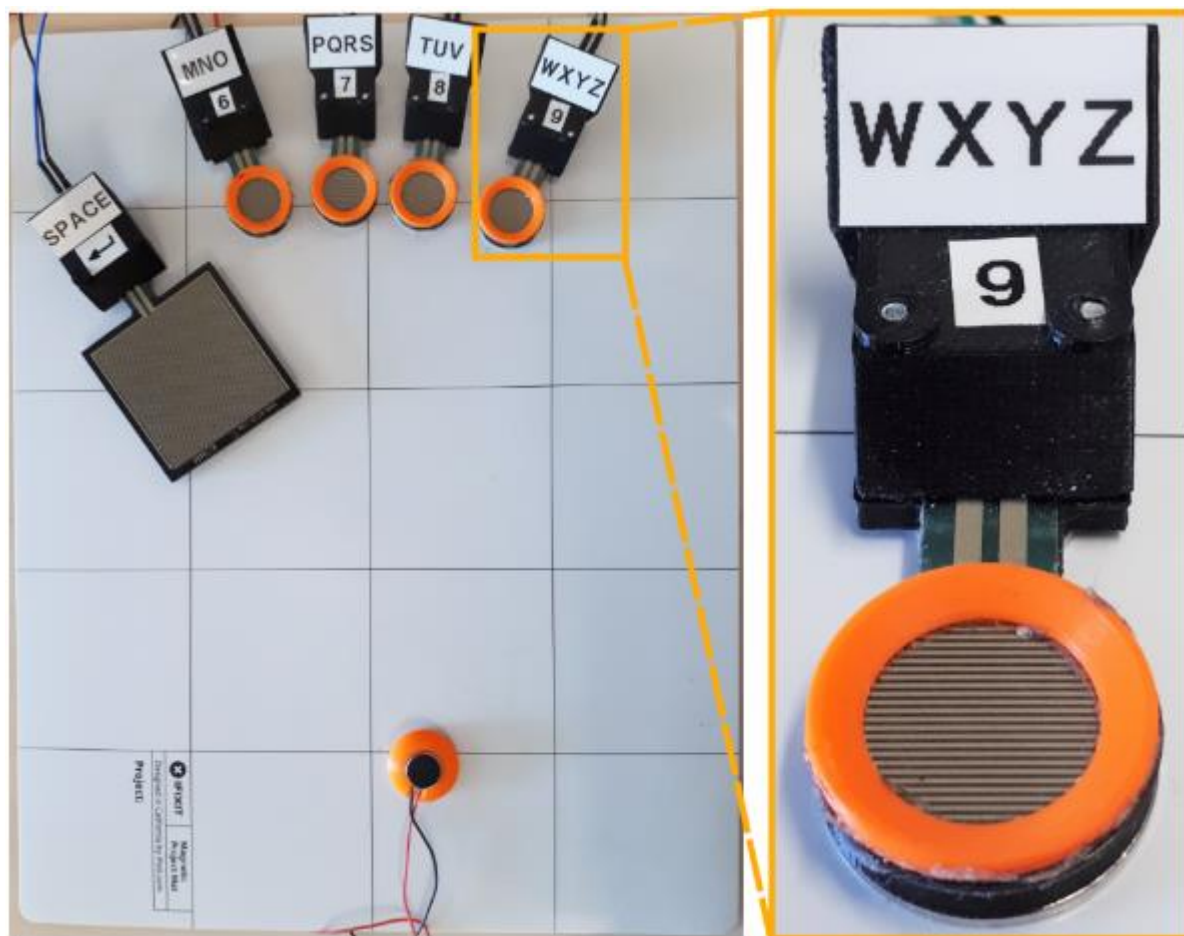


Figura 17 - Interface acionadora do teclado virtual.

Fonte: ANDREAS *et al.* (2022).

A interface contém pequenas adaptações para fornecer melhores recursos de usabilidade como, por exemplo, profundidade com bordas nos botões onde ficam os dedos, para evitar o escorregar deles. Antes do processo de digitação, propõe-se fatores de configuração e calibração da interface, como a sensibilidade dos acionadores e tempo da detecção da pressão longa, pois esses fatores podem variar conforme a deficiência motora. O teclado virtual é mostrado em tela e contém nível de opacidade sobre outras telas do sistema. Na Figura 18, ilustra-se o *layout* do teclado, onde se pode perceber que no centro é mostrada a palavra digitada seguida de um dicionário probabilístico da próxima palavra. O próprio usuário pode inserir novas palavras no dicionário.

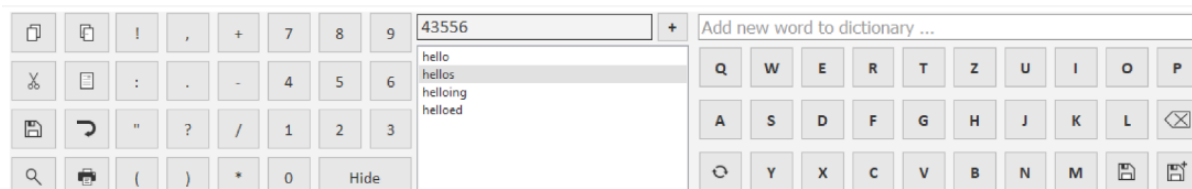


Figura 18 - Interface do teclado virtual proposto com sistema de dicionário de próxima palavra.

Fonte: ANDREAS *et al.* (2022).

Os testes foram realizados com três pessoas sem deficiência motora, as quais utilizaram um teclado convencional e o proposto pelos autores. Os testes foram realizados várias vezes com o propósito de aumentar a curva de aprendizado. Os autores relatam a utilização do *site* “10fastfingers.com” tal qual sugere frases para digitação com letras maiúsculas e minúsculas. Foram avaliadas as palavras digitadas e a ocorrência de erros. Os resultados do experimento indicam que, após ao menos três utilizações, os usuários atingiram média de 8,44 wpm. Os autores afirmam que fatores como a maior utilização e memorização dos comandos possam ampliar a eficiência do teclado, assim como o acréscimo de palavras ao dicionário de sugestão de palavras.

2.5 Princípios de Acessibilidade na Web

Softwares desenvolvidos em ambiente web que se preocupam em fornecer recursos de acessibilidade podem seguir diretrizes e padrões bem estabelecidos pelo World Wide Web Consortium (W3C) em suas Diretrizes de Acessibilidade para Conteúdo Web (WCAG) (ZAHRA, 2019). Estas servem para garantir que pessoas com deficiência tenham igualdade de acesso à informação e aos serviços disponíveis na web.

2.5.1 Perceptividade

A perceptividade refere-se à capacidade de percepção sobre os conteúdos apresentados na tela de um *site*. Efetivamente, como o conteúdo está apresentado, relacionado a cores, tamanhos de cores e fontes de textos. É importante que conteúdos não textuais apresentem versões textuais também.

2.5.2 Operatividade

Operatividade refere-se à capacidade de controle dentro do *site*, isto é, sobre os elementos interativos assim como a navegação entre as telas. Botões de navegação em lugares estratégicos e a possibilidade de operação de todo o *site* pelo teclado são formas fundamentais para pessoas com deficiência motora.

2.5.3 Compreensibilidade

O conteúdo apresentado deve ser claro no que quer transmitir, sem que apresente formas ambíguas de interpretação. A forma de linguagem deve ser padronizada e, se necessário, disponibilização de recursos de ajuda e documentação.

2.5.4 Robustez

Robustez refere-se à capacidade do conteúdo ser apresentado em diversas situações diferentes como, por exemplo, *layouts* de telas variadas, navegadores distintos e dispositivos variados como celulares, *notebooks* ou *desktops*.

2.5.4 Velocidade

O W3C busca assegurar que as páginas da web e as aplicações carreguem rapidamente e respondam de forma eficiente às interações dos usuários. Isso é fundamental para melhorar a experiência do usuário, reduzir a latência e promover uma navegação fluida e agradável.

3. METODOLOGIA

3.1 Tipo de Pesquisa

O estudo proposto caracteriza-se como uma pesquisa Quantitativa Descritiva e Exploratório-Experimental. Foi desenvolvida uma aplicação web com funcionalidades voltadas para escrita de textos e comunicação por meio de um teclado virtual direcionado a pessoas com deficiências motoras. A avaliação foi conduzida por meio de simulações automatizadas, nas quais foram quantificadas as médias de wpm que o software pode alcançar. Esses resultados poderão ser comparados com os de outros teclados virtuais assistivos disponíveis na literatura.

3.2 Estrutura do Projeto

O processo de levantamento de escopo realizado para este projeto foi fundamental para identificar os elementos essenciais necessários para a criação do teclado virtual proposto. É importante ressaltar que esse levantamento foi realizado antes do início do projeto, e que mudanças ocorreram ao decorrer do desenvolvimento.

Inicialmente, foram identificadas as necessidades relacionadas à predição textual do teclado:

1. Viabilidade Tecnológica: investigação dos recursos de tecnologia disponíveis para utilização e que se adequam ao projeto.
2. Disponibilização e Testes: compreensão dos fatores relacionados à forma como o serviço será disponibilizado para a aplicação web e como os testes dos recursos expostos serão executados.

Em seguida, foram avaliadas as necessidades de recursos dentro do aplicativo web:

1. Viabilidade Tecnológica: identificação dos recursos de tecnologia disponíveis para utilização e adequados ao projeto.
2. Definição das Telas: determinação das telas que estarão disponíveis para o usuário.

3. Definição dos Controladores: identificação dos controladores primários disponibilizados para controle do teclado virtual.
4. Mapeamento das Funções dos Controladores: compreensão das necessidades de mapeamento dos controladores definidos para utilização na aplicação.
5. Armazenamento de Dados: definição da necessidade de armazenamento dos dados e especificação de quais dados serão armazenados.

Por fim, foram avaliadas as necessidades de testes automatizados:

1. Levantamento dos Testes: identificação do que é necessário testar com base no problema que se propõe resolver.
2. Levantamento dos Dados Esperados dos Testes: definição dos tipos de dados aceitáveis como resultados dos testes e das métricas que serão validadas.
3. Interpretação dos Resultados: comparação dos resultados obtidos com outros projetos e determinação da viabilidade de disponibilização para o usuário final.

3.3 Desenvolvimento de Software Web (Site)

Este aplicativo web foi desenvolvido utilizando exclusivamente a linguagem de programação web JavaScript. Segundo pesquisas recentes, essa linguagem corresponde a 69,7% das mais comumente usadas em desenvolvimento web (UC BERKELEY EXTENSION, 2023). A portabilidade do aplicativo foi inicialmente projetada para resoluções de monitores HD. A escolha de inspirar-se no sistema Dasher de WARD & MACKAY (2002) foi motivada por suas características únicas e desempenho eficaz em comparação com as médias de velocidade de seleção (palavras por minuto) alcançadas por pessoas com deficiências motoras.

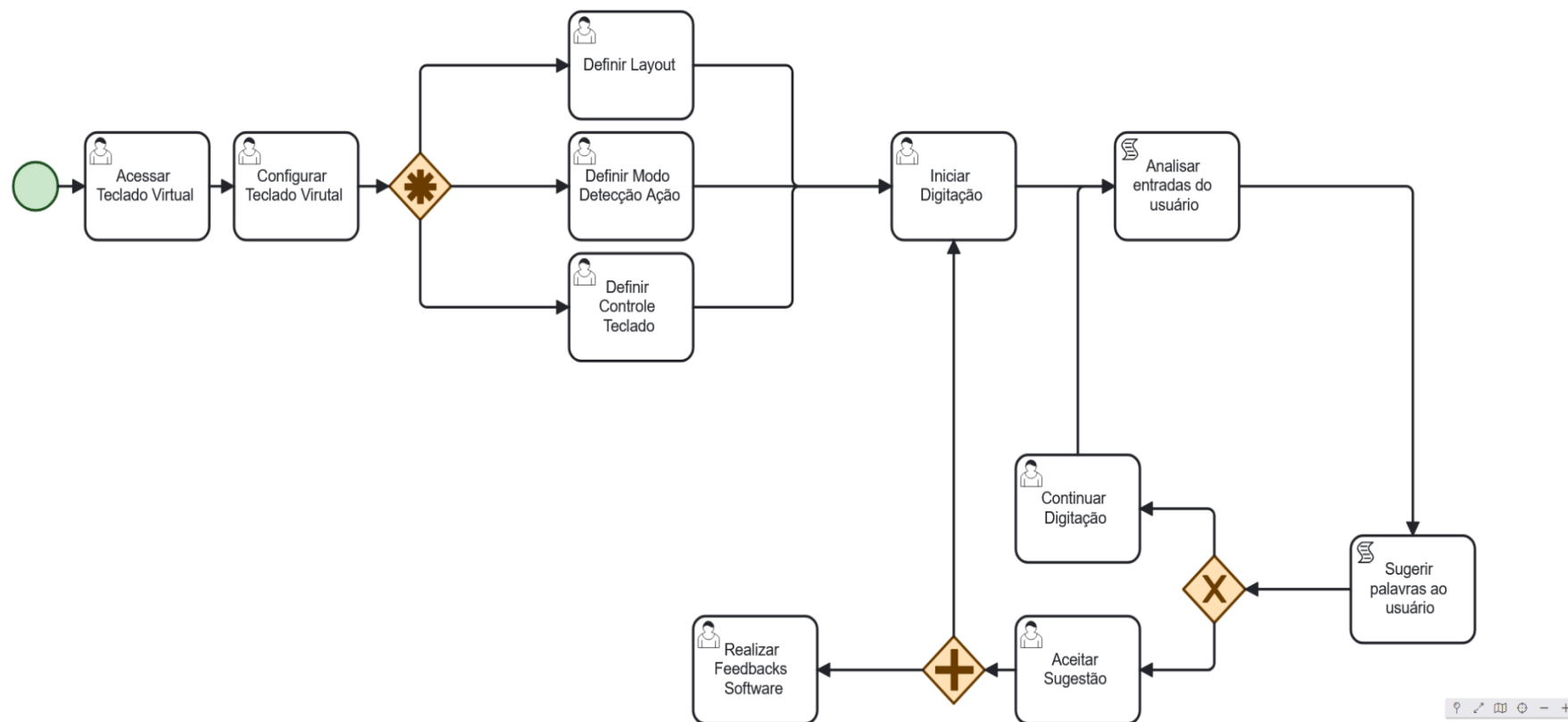


Figura 19 - Fluxograma geral do software.

Fonte: Autor.

Pode-se interpretar com base neste fluxograma (Figura 19) que o usuário recebe funções operacionais de acesso ao teclado, configurações (layouts, modos de uso e controlador), entrada de dados (realizar digitações) e aceitação de sugestões que são analisadas e sugeridas pelo sistema. A aplicação pode ser descrita tecnicamente da seguinte forma:

- Visão Geral: a aplicação tem como principal objetivo possibilitar a digitação por meio de um teclado virtual, onde o mesmo possa ser operado com controladores alternativos (joysticks, mouses e controle ocular) e seja customizado com funcionalidades;
- Arquitetura: Conforme mencionado a aplicação será arquitetada única e exclusivamente utilizando JavaScript, utilizando o framework Angular e a biblioteca de rastreamento ocular Webgazer;
- Funcionalidades: a aplicação traz a possibilidade de digitação de todos os caracteres do alfabeto, assim como símbolos numéricos e pontuações. Com base em entradas de usuário faz previsões do que o usuário está digitando, assim como contabiliza indicadores de desempenho durante a execução de entradas de texto. Também é possível realizar customizações sobre o teclado virtual;
- Especificações Técnicas: é necessário no mínimo uma máquina que suporte os navegadores web em suas versões mais atualizadas, necessário consulta por navegador utilizado;
- Interface do Usuário: são propostos três layouts, sendo um exclusivamente para feedbacks, outro para que seja possível customizar o teclado virtual tendo possibilidades de definir controladores, customização total da exibição das opções do teclado virtual e, por fim, a tela do teclado virtual em si que exibe a tela conforme customizada para digitação de texto;
- Segurança: é uma aplicação totalmente de livre acesso e não conta com informações sensíveis;
- Desempenho: a aplicação funciona em *front-end-side*, tendo processamento por máquina do usuário e não em servidor externo;

3.4 Conexão de Controladores

Este projeto visa poder atender de maneira nativa ao *site* suporte para dispositivos alternativos. Para tal, será utilizada a API HID que permite que após autorização do usuário sejam coletados dados de entrada dos dispositivos conectados via Bluetooth ou USB. Neste trabalho, os dados foram captados e mapeados para comandos que simulam a utilização do *mouse* ou teclado. Na Figura 20, ilustra-

se o diagrama da estrutura proposta para utilização de dispositivos incomuns em *sites web*.

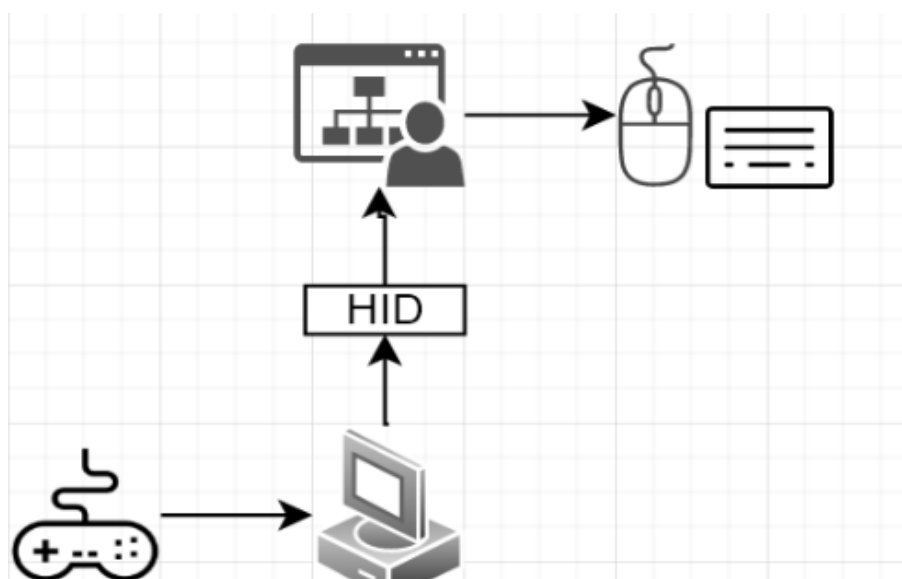


Figura 20 - Fluxo da conexão e utilização de um dispositivo incomum via HID.

Fonte: Autor.

Com relação aos métodos de mapeamento para *mouse*, foi realizado utilizando a conversão dos dados dos dispositivos para dois eixos (xy). É importante ressaltar que o próprio ponteiro do *mouse* não pode receber eventos programaticamente dos *browsers* por questões de segurança. A possível limitação seria que o *mouse* pode interagir com os elementos de tela por meio do evento de “*click*”; porém, neste trabalho, propõe-se um algoritmo de colisão entre os elementos onde este evento pode ser compensado de forma programática. Em sua primeira versão, será possível utilizar o teclado virtual, por meio de controles *joystick*, mouse e controles sensoriais como oculares e operados por variação de movimento:

Joycon (Modos *Joystick* e Variação de Movimento)

1. **Acessibilidade:** são controles pequenos e leves, o que os torna mais manejáveis para pessoas com deficiência motora. É possível controlá-los com movimentos sutis, o que pode ser mais conveniente para pessoas com limitações de mobilidade.
2. **Variedade de modos de uso:** possuem sensores de movimento e botões que podem ser mapeados para diferentes funções. Com isso, é possível adaptar-se às necessidades específicas da pessoa com deficiência motora.
3. **Compatibilidade:** são controles projetos para *videogames*, contudo é possível utilizá-lo em um computador.

4. Preço acessível: Comparadas as soluções de *hardware* especializado, os Joycons podem ser mais acessíveis financeiramente.
5. Disponibilidade: Os Joycons são amplamente disponíveis em muitas regiões, o que facilita a aquisição e a substituição em caso de necessidade.

Controles *joystick* padrão Dualshock

1. Ergonomia e familiaridade: são controles projetados com foco na ergonomia e no conforto, o que pode ser benéfico para pessoas com deficiência motora. Além disso, muitas pessoas já estão familiarizadas com esses controles devido à popularidade dos *videogames*, o que pode facilitar a adaptação.
2. Acessórios e adaptações: às empresas fabricantes destes dispositivos tem investido em acessórios e adaptações para tornar seus controles mais acessíveis a pessoas com deficiência. Esses acessórios podem incluir extensões, botões remotos, *joysticks* de tamanho diferenciado e outras modificações que se adequam às necessidades individuais.
3. Compatibilidade: são controles projetados para *videogames*, contudo é possível utilizá-lo em um computador.
4. Comunidade e suporte: ambas as plataformas têm comunidades ativas e recursos de suporte *online*, o que pode ser útil para obter orientações e soluções para adaptar os controles às necessidades da pessoa com deficiência.
5. Preço acessível: assim como os Joycons, esses controles também são amplamente disponíveis e podem ser mais acessíveis financeiramente em comparação com dispositivos especializados.

Controles de detecção ocular

1. Normalmente, um controle de teclado virtual operado por câmera pode conter três abordagens possíveis: rastreamento ocular, detecção do posicionamento ocular e controle pelo piscar dos olhos. Em primeira instância, propõe-se para esta aplicação o controle sensorial pela detecção do posicionamento, sendo utilizado devido à viabilidade de um pacote que realiza este tipo de ação conforme trabalho de PAPOUTSAKI et al. (2018) com resultados satisfatórios que possam atender o propósito do teclado virtual. O método de piscar de olhos não é adotado pela dificuldade de operatividade e as baixas médias conforme relatadas pelo trabalho de CECOTTI et al. (2019), já o método de *tracking* ocular exige grande processamento e no momento os pacotes disponíveis para linguagem utilizada ainda não fornecem precisão suficiente para operar um teclado virtual conforme desenhado por este projeto. Métodos que

impliquem na necessidade de acoplamento de dispositivos na face do usuário podem causar problemas em âmbito social e são necessários diversos testes com usuários físicos e avaliações de cenários que possam comprometer sua utilização; por isso, não serão abordados.

É importante notar que cada pessoa é única, e as necessidades individuais podem variar amplamente (WESSELS et al., 2003). Este trabalho visa ampliar a gama de controles suportados à medida que o projeto for difundido, assim como aceitar sugestões de usuários via tela de *feedback* que estará disponível na aplicação.

3.5 Sintetizador de Fala

Após digitações no teclado é possível por meio de funcionalidade do teclado virtual falar aquilo que foi digitado, sendo assim possível fornecer recursos de comunicação ao usuário com deficiência motora. A linguagem Javascript adotada, contempla o recurso de sintetizador de voz, onde a partir de uma entrada de texto uma voz em parte robotizada é acionada para realizar a ação da fala, adotamos essa função nativa para implementação da funcionalidade.

3.6 Indicadores de Desempenho

São coletados e armazenados em nuvem alguns indicadores de desempenho durante a utilização do *software* :

Contagem de Velocidade de Seleção ou Palavras por Minuto (WPM): após iniciado, será calculado quantas palavras o usuário digita a cada minuto no teclado virtual por meio da equação 1. Cada palavra considerada deve conter 5 caracteres. O contador de tempo será pausado em momentos de não utilização do usuário.

$$WPM = \frac{|C| - 1}{T} * 60 * \frac{1}{5} \quad (\text{equação 1})$$

Onde:

|C| : caracteres digitados, modular em fórmula pela não presença de numeração negativa no contexto;

T : tempo de execução do *software*, caso realize uma parada a contabilização deste tempo é parada também.

Contagem de Caracteres por Minuto (CPM): após iniciado o teclado virtual, será calculado cada caractere que o usuário digita a cada minuto (equação 2). O contador de tempo será pausado em momentos de não utilização do usuário.

$$CPM = \frac{|C| - 1}{T} * 60 \quad (\text{equação 2})$$

Onde:

|C| : caracteres digitados, modular em fórmula pela não presença de numeração negativa no contexto;

T : tempo de execução do *software*, caso realize uma parada a contabilização deste tempo é parada também.

Taxa de erros por Minuto (EPM): após iniciado o teclado virtual, será calculada cada ação de “Apagar” (equação 3). Ao final da utilização, será calculada a taxa. Os minutos utilizados no cálculo só são referentes aos momentos em que o *software* realmente estiver em uso.

$$EPM = \frac{|E| - 1}{T} * 60 \quad (\text{equação 3})$$

Onde:

|E| : contabilização da ação “Apagar”, modular em fórmula pela não presença de numeração negativa no contexto;

T : tempo de execução do *software*, caso realize uma parada a contabilização deste tempo é parada também.

Palavras Preditas por Minuto (PPM): após iniciado o teclado virtual, será calculado quantas palavras o usuário digita a cada minuto, caso selecione uma palavra predita (que será contabilizada) (equação 4). O contador de tempo será pausado em momentos de não utilização do usuário.

$$PPM = \frac{|PP| - 1}{T} * 60 \quad (\text{equação 4})$$

Onde:

|PP| : contabilização de uma palavra predita, não considerando palavras onde foi predita faltando apenas um caractere, modular em fórmula pela não presença de numeração negativa no contexto;

T : tempo de execução do *software*, caso realize uma parada a contabilização deste tempo é parada também.

Classificação Palavras Preditas por Minuto: após iniciado o teclado virtual, será classificado com quantos caracteres faltantes uma palavra foi predita durante a execução completa de um texto. Os dados serão armazenados conforme exemplo listado na Tabela 1.

Tabela 1 - Contabilização de com quantos caracteres faltantes uma palavra foi predita.

Palavra Digitada Usuário	Palavra Predita	Número Caracteres Faltantes
Principal	Principalmente	5

Fonte: Autor.

3.7 Armazenamento de *Feedbacks* e Desempenho

Os *feedbacks* enviados por usuários são armazenados em nuvem, sendo utilizado o armazenamento do Firebase da Google, visto que o mesmo contempla fácil conexão e armazenamento dos dados. Além do texto de livre digitação de *feedback* enviado pelos usuários, são coletados às seguintes informações do cache da aplicação:

1. *browser*: *Browser* e Versão do mesmo (ex: Firefox V.1200);
2. OS: Sistema Operacional;
3. *OS_version*: Versão do Sistema Operacional;
4. *device*: *Tablet*, *Mobile* ou *Desktop*;
5. *deviceId*: Identificador Único do dispositivo do usuário;
6. *applicationcontroller*: refere-se ao dispositivo utilizado pelo usuário para controlar o teclado virtual;

Durante a utilização da aplicação, os indicadores de desempenho serão enviados para o armazenamento em nuvem também, sempre as últimas 5 sessões de utilização, sendo compreendidos pelas seguintes informações:

1. Contagem de Palavras por Minuto;

2. Contagem de Caracteres por Minuto;
3. Taxa de erros por Minuto;
4. Palavras Preditas por Minuto;
5. Predição Ignoradas por Minuto;
6. Classificação Palavras Preditas por Minuto.

3.8 Testes Automatizados

Os testes automatizados são realizados utilizando um pacote específico da linguagem Javascript denominado Cypress (CYPRESS, 2023). Este *framework* pode emular a utilização de um usuário real, interagindo com os elementos em tela assim como um usuário o faria. Não é possível realizar simulações dos perfis dos usuários, principalmente se considerar deficiências físicas específicas. Contudo, é de extrema importância que antes de ser disponibilizado ao público, o *software* contenha este tipo de teste automatizado, pois se não for validado pela ferramenta dificilmente o usuário final terá uma boa experiência de uso.

A ideia central dos testes automatizados do teclado virtual é simular a digitação do teclado virtual como um usuário faria. Durante a rodada do teste a aplicação é aberta no *browser* (Edge), um texto definido como parâmetro de entrada por *script* é definido como objetivo da digitação utilizando o teclado, a simulação identifica a letra da sequência e faz a captura, por meio dessas entradas a aplicação assim como se comportaria para um usuário convencional realiza as sugestões de palavras, o teste automatizado verifica as sugestões também, onde tendo uma sugestão da palavra atual da sequência também realiza a captura. Uma randomização de erro via *script* é executada a cada seleção, tendo um fator de 1/100, quando atingido uma seleção incorreta é realizada, e através disso a automatização tem que executar as ações de limpeza de última digitação. Outros dois parâmetros são configurados nos cenários de testes, sendo relacionados a velocidade do controle do teclado virtual, sendo estes:

Velocidade do Ponteiro:

A aplicação poderá ser controlada por *mouse* ou *joysticks*, a utilização destes dispositivos interage com elemento visual em tela para realizar ações no teclado virtual. Este parâmetro está relacionado à velocidade com que o elemento visual se moverá pelo teclado virtual, sendo a métrica relacionada a quantos pixels na tela o elemento visual se moverá com base no movimento do dispositivo controlador.

Delay Sensorial:

Controles operados por câmera, ou por movimentos podem ocasionar erros de captura, por este motivo ao realizar alguma ação no teclado, necessitamos de um tempo de espera e indicar visualmente ao usuário que aquela ação será contabilizada. Este parâmetro se refere a quanto tempo a aplicação irá aguardar para que seja realizada a contabilização da ação, aplicado em métrica temporal de milissegundos. A Tabela 2 exemplifica a relação dos valores que serão usados nos cenários de testes.

Tabela 2 - Cenários de Testes Mapeados.

Tamanho do Texto (palavras)	Velocidade do Ponteiro (Pixels por movimento)	Delay Sensorial (ms)
1800	600	500
1800	1200	2000

Fonte: Autor.

3.9 Predição de Palavras

O método adotado neste trabalho utiliza um modelo preditivo estático baseado no "Corpus do Português" de DAVIES (2011). Este corpus é uma coleção extensa de textos em português que serve como um recurso para análise linguística e modelagem de linguagem. O modelo preditivo foi alimentado com dados extraídos deste corpus, que foram então importados para um banco de dados NoSQL em memória da aplicação web. Para otimizar o desempenho e a relevância das sugestões, foi aplicada uma métrica de corte baseada na frequência das palavras. A relevância das palavras foi determinada identificando aquelas que aparecem com maior frequência no corpus, enquanto termos menos comuns, como jargões técnicos ou palavras específicas de determinados domínios, foram considerados menos relevantes para o objetivo do teclado virtual. Apenas palavras com 25.000 ou mais ocorrências foram incluídas no banco de dados, garantindo que o modelo ofereça sugestões de palavras amplamente reconhecidas e utilizadas.

O banco de dados aplica um método de ranqueamento para selecionar as seis palavras mais prováveis com base nos dados de entrada do usuário, que são as letras digitadas. Essas palavras são então apresentadas ao usuário na interface web como sugestões. Quando uma palavra é selecionada pelo usuário, ela

recebe um incremento unitário em seu ranking, o que melhora sua posição nas futuras sugestões. Palavras que não estão inicialmente presentes na base de dados são adicionadas de forma que não afetam imediatamente o algoritmo de sugestão, mas são posicionadas de maneira a serem rapidamente promovidas no ranking se forem comuns no vocabulário do usuário. Além disso, a predição é utilizada para reorganizar a posição das letras exibidas na interface. As letras são organizadas com base nas palavras mais prováveis, que são ranqueadas a partir das entradas do usuário. Como ilustrado na Figura 21, a entrada "Cac" resultaria na exibição das seis palavras mais prováveis, e a ordem das letras refletiria a frequência dessas letras nas 20 palavras mais prováveis.

Para uma compreensão mais aprofundada sobre modelos preditivos e a aplicação de corpora na modelagem de linguagem, é recomendável consultar a obra de DAVIES (2011) intitulada *Corpus do Português*. Adicionalmente, a leitura de trabalhos acadêmicos sobre o uso de corpora em sistemas de predição e sugestão de palavras pode fornecer uma visão mais completa sobre o tema.

Base Dados:		Entrada Usuário: Cac	
1.	Cachorro	Sugestões Aplicação	
2.	Cacto		
3.	Cacau		
4.	Cachoeira		
5.	Cactos		
6.	Cachecol		
7.	Cachimbo		
8.	Cacique		
9.	Caciquear		
10.	Cachimbar		
11.	Cactaceae		
12.	Cachaça		
13.	Cachorro-quente		
14.	Cactáceas		
15.	Cachopa		
16.	Cacete		
17.	Cacareco		
18.	Cachê		
19.	Cachaço		
20.	Cacimba		
		1.	A
		2.	Cachorro
		3.	O
		4.	Cacto
		5.	C
		6.	E
		7.	R
		8.	H
		9.	Cachoeira
		10.	I
		11.	Cacau
		12.	U
		13.	Cactos
		14.	Cachecol

Figura 21 - Exemplificação algoritmo preditivo.

Fonte: Autor.

4. RESULTADOS

O primeiro *layout* do teclado virtual proposto apresentado na Figura 23, inspirado principalmente no conceito dinâmico do já citado Dasher, onde as letras posicionadas à direita se deslocam para a esquerda em direção ao cursor representado pelo retângulo vermelho da imagem, algumas ações estáticas são apresentadas no setor esquerdo do *layout*:

1. Limpar Toda Digitação: limpa toda digitação já contabilizada no campo de texto acima do teclado.
2. Vogais: altera as letras apresentadas no teclado para mostrar apenas vogais.
3. Limpar Último: limpa última letra digitada pelo usuário.
4. Consoantes: altera as letras apresentadas no teclado para mostrar apenas consoantes.
5. Falar Texto Completo: o texto digitado é reproduzido pela saída de som do computador por uma voz sintetizada padrão do navegador.
6. Inserir Espaço: insere espaço no texto digitado.
7. Mostrar Mais: por limitações do layout são demonstradas um número limitado de letras, então a ação serve para que o *layout* apresente as letras que ainda não foram apresentadas em tela.

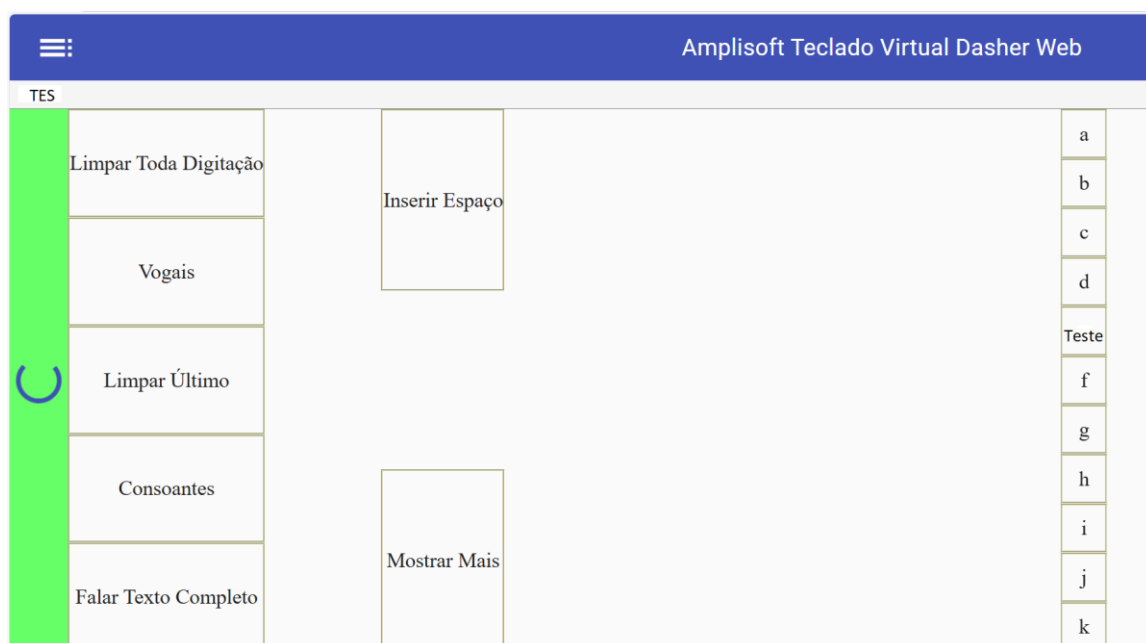


Figura 23 – Primeiro *layout* proposto.
Fonte: Autor.

Na sequência, foram executados os testes automatizados que simulam a utilização do usuário conforme os parâmetros de entrada especificados, como o tamanho do texto, a velocidade de movimentação para os controles *joystick* e o *delay* de iterações para os controles sensoriais. A Tabela 3 apresenta os resultados obtidos com base nos indicadores de desempenho previamente estabelecidos. Cada cenário foi repetido 5 vezes para determinar os ganhos relativos à atualização do *ranking* do algoritmo preditivo. É importante observar que não é possível simular com precisão os erros por minuto que uma pessoa poderia cometer, pois esse é um fator variável influenciado por diversos elementos externos. Por padrão, foi considerada uma taxa randomizada de um erro a cada 100 letras digitadas.

Tabela 3 - Resultados Primários Testes Automatizados

Controle	Loop	VP	wpm	cpm	ppm	epm
Joystick	0	600	7.33	36.65	3.63	0.55
Joystick	1	600	7.37	36,85	3.66	0.55
Joystick	2	600	7.44	37.2	3.71	0.59
Joystick	3	600	7.41	37.05	3.69	0.62
Joystick	4	600	7.49	37.45	3.75	0.57
Joystick	0	1200	13.71	68.55	7.89	0.73
Joystick	1	1200	13.73	68.65	7.90	0.71
Joystick	2	1200	13.76	68.8	7.92	0.73
Joystick	3	1200	13.78	68.9	7.94	0.68
Joystick	4	1200	13.84	69.2	7.96	0.72

VP: Velocidade Ponteiro; WPM: Palavras por Minuto; CPM: Caracteres por Minuto; PPM: Palavras Preditas por Minuto; EPM: Erros por minuto;
Fonte: Autor.

Após analisar os dados, foi tomada a decisão de não implementar as lógicas de testes para os controles sensoriais que ainda estavam pendentes. Isso ocorreu porque os indicadores demonstraram que a aplicação estava com médias mais baixas do que as esperadas, comparativamente à abordagem de inspiração, que possibilitou encontrar médias de até 23 wpm com usuários treinados. Neste momento, realizou-se uma mudança no *layout*, visando à melhoria dos indicadores. Na seção de discussão, são apresentadas as causas das médias encontradas e a motivação para o desenho de novos *layout*.

Desenhamos dois novos *layouts* padrões para aplicação, sendo o primeiro deles disposto de duas linhas conforme a Figura 24.

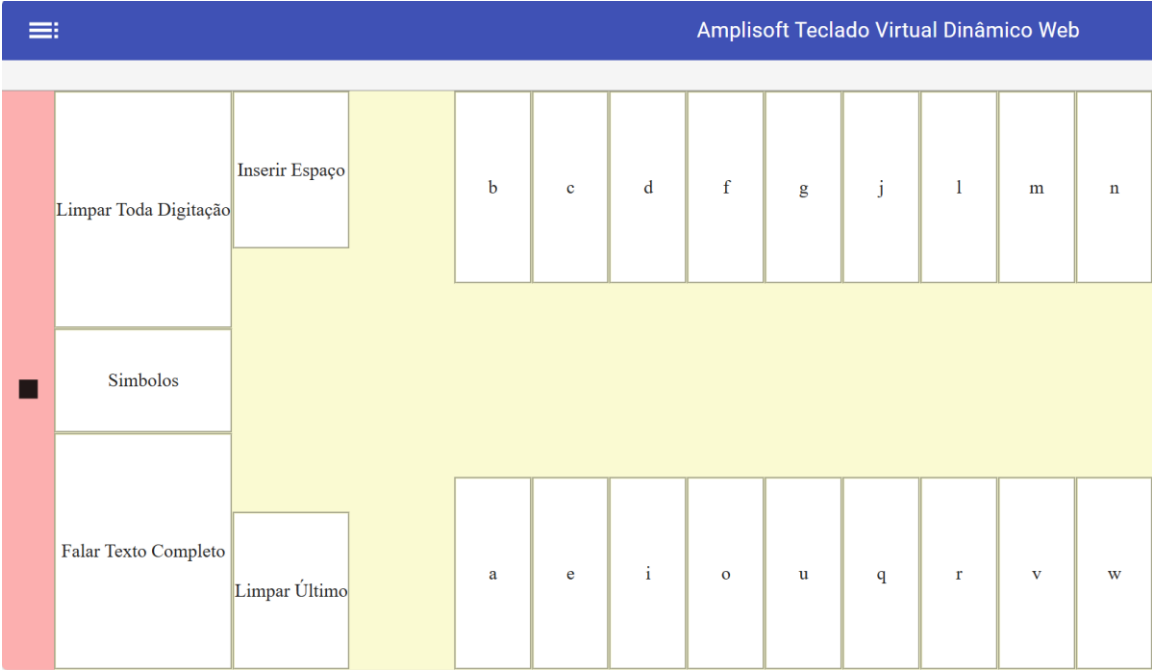


Figura 24 – Layout padrão duas linhas.

Fonte: Autor.

Em sequência conforme a Figura 25 o desenho do layout de quatro linhas.

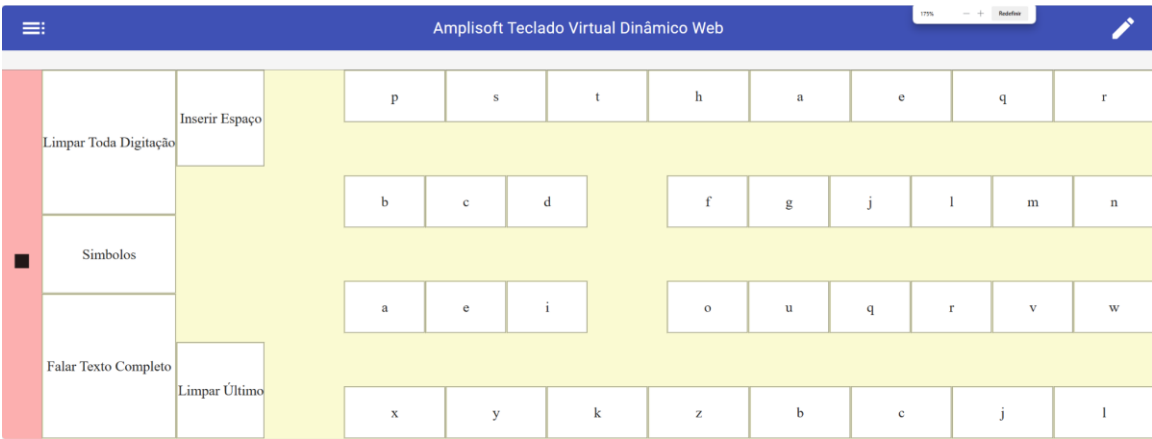


Figura 25 – Layout padrão quatro linhas.

Fonte: Autor.

A utilização do teclado acontece de formas configuráveis conforme demonstrado na Figura 26:

1. Dispositivos Suportados: possibilitar ao usuário definir o controlador alternativo do teclado virtual, por padrão é possível realizar o controle com *mouse* ou um dos listados.

2. Detecção de Ação: configura como os controles operados por *joysticks* ou *mouses* irão interagir com o teclado virtual, sendo a captura das ações possível pelo contato com o elemento ou após uma faixa de tempo.
3. Delay Detecção Ação: Este campo configura o tempo que o teclado irá esperar para contabilizar a ação caso o tipo da detecção seja por contagem de tempo, onde durante este tempo um indicativo visual é demonstrado para o usuário para indicar a “captura” da ação.
4. Tipo Layout: Determina qual *layout* do teclado virtual será demonstrado na tela do teclado virtual, por padrão, tendo modos vertical e horizontal.

Configurações Gerais	Dispositivos Suportados
Tipo Layout Layout Customizado	☐ Joystick Nintendo Joycon Direito
Detecção Ação Contagem Tempo	☐ Joystick Nintendo Joycon Esquerdo
Delay detecção ação (MS)* 2000	☐ Joystick DualShock
	☐ Sensorial Nintendo Joycon Direito
	☐ Sensorial Nintendo Joycon Esquerdo
	☐ Controle Ocular

Figura 26 - Tela configurações controles finalizadas.

Fonte: Autor.

Exemplificando melhor essas configurações, o campo “Tipo de Layout” da Figura 26, determina se a aplicação irá demonstrar o *layout* da Figura 24 ou 25. Com relação ao campo “Detecção Ação” ele tem duas variantes de opção:

- Contagem Tempo: Ao posicionar o ponteiro sobre uma das ações disponíveis no teclado o sistema apenas irá contabilizar a ação após uma faixa de tempo que é configurada no campo “Delay detecção ação (MS)” (padrão 2 segundos), ilustramos na Figura 27 esse tipo de captura:

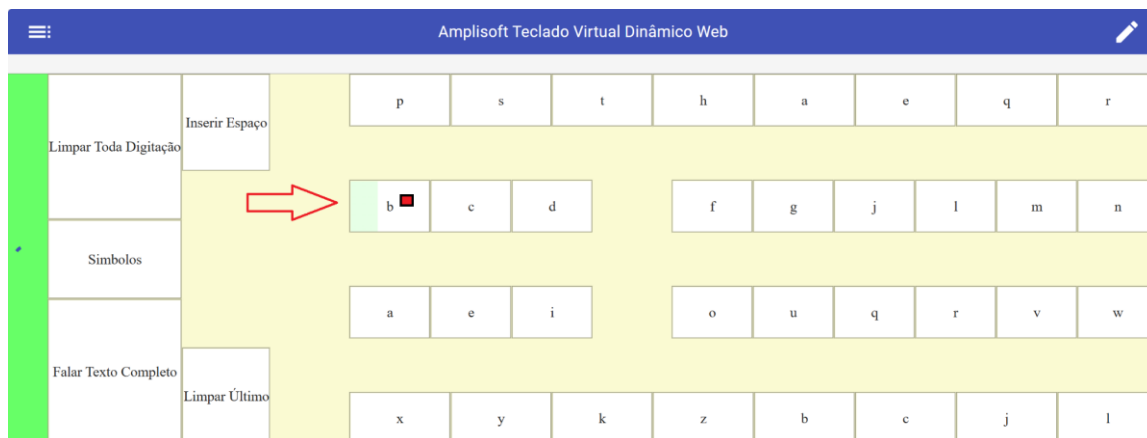


Figura 27 - Captura da ação por contagem de tempo.

Fonte: Autor.

Conforme observamos na Figura 27, ao posicionar o ponteiro (controlado por *mouse* ou dispositivo alternativo) sobre um elemento do teclado ele realiza o feedback que a ação estará sendo contabilizada, alterando a cor da caixa da ação para a cor verde, após o tempo configurado a ação é contabilizada.

- Ao Contato: Imediatamente ao posicionar o ponteiro sobre uma ação ela é contabilizada pelo sistema, podemos observar na Figura 28 o comportamento.

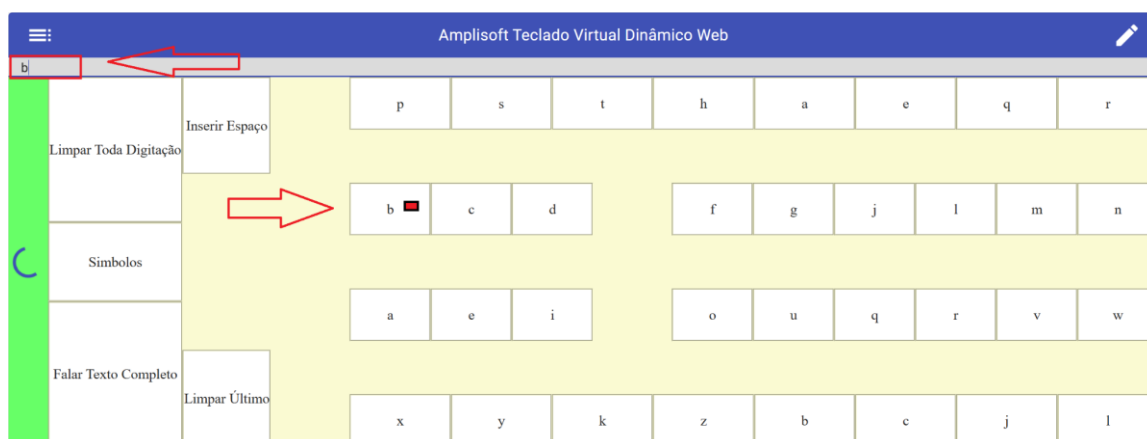


Figura 28 - Captura da ação de forma imediata.

Fonte: Autor.

Conforme observado na Figura 28 pelas setas indicadas, ao posicionar o ponteiro sobre a letra “b” ela foi imediatamente computada no campo de indicação do que foi digitado pelo sistema, identificado pela seta superior.

Em ambos modelos de “Detecção Ação” ao realizar entradas de letras o sistema irá no campo superior indicar a digitação do usuário até o momento, e por sequência reorganizar o teclado, mantendo símbolos mais prováveis nas

primeiras colunas do *layout*, assim como as sugestões de palavras. Na Figura 29 ilustramos esta situação.

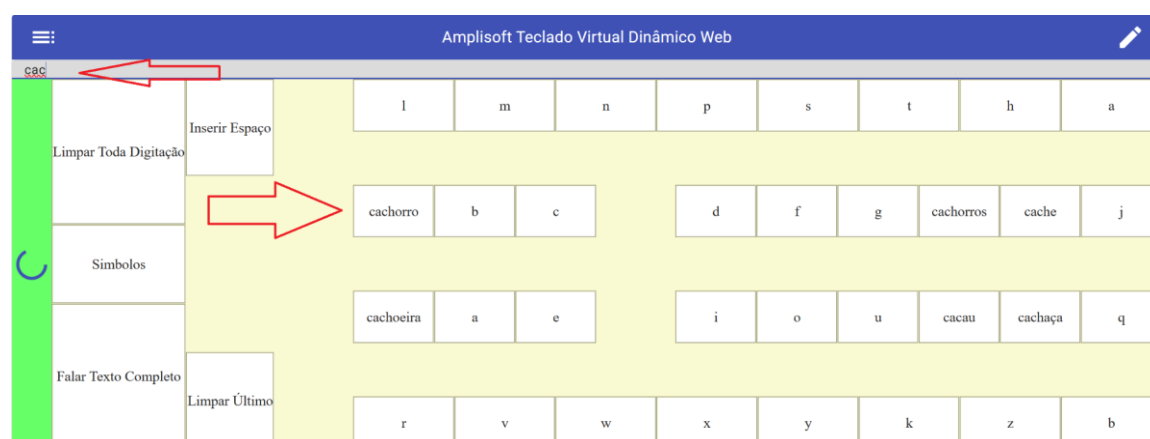


Figura 29 - Reorganização do teclado.

Fonte: Autor.

Então conforme vemos na Figura 29, através da entrada prévia “cac” indicado pela seta vermelha superior, o sistema reorganizou as letras e realizou sugestões de palavras conforme a seta vermelha inferior apontando para sugestão “cachorro”.

Além da contabilização das letras e palavras algumas ações estáticas estão disponíveis no teclado virtual sendo elas:

1. Limpar Toda Digitação: Limpa toda digitação já executada pelo usuário, redefinindo o campo da entrada do que foi digitado para vazio.
2. Inserir Espaço: Insere espaço no campo de entrada do que foi digitado pelo usuário.
3. Falar Texto Completo: O sintetizador de voz padrão do browser irá falar aquilo que foi digitado no campo de entrada do que foi digitado pelo usuário, não conseguimos demonstrar nesta dissertação este recurso de forma visual mas ele pode ser validado pelo acesso gratuito da aplicação pelo link <https://juniorschen.github.io/Amplisoft-Teclado-Virtual-Dinamico>.
4. Limpar Último: Limpa o último caractere digitado no campo de entrada do que foi digitado pelo usuário.
5. Símbolos: Altera para que o teclado faça exibição dos números e caracteres especiais, assim como demonstrado pelos retângulos vermelhos na Figura 30.

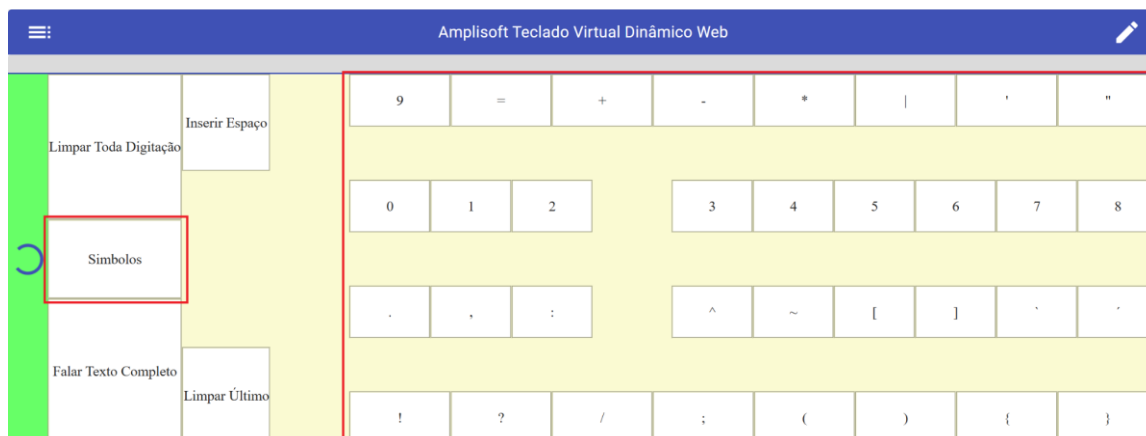


Figura 30 - Exibição dos números e símbolos.

Fonte: Autor.

- Dispositivos suportados: Por padrão o teclado pode ser controlado pelo mouse convencional, existem diversos dispositivos assistivos que plugados via USB podem controlar o mouse e por sequência interagir com o teclado virtual, contudo alguns dispositivos alternativos são propostos para controle da aplicação como relatado na seção 3.4 da metodologia, então o campo de “Dispositivos suportados” das configurações da Figura 26 é utilizado para que a aplicação entenda qual destes dispositivos será usado, onde quando configurado é possível controlar a aplicação por meio destes dispositivos, não é possível demonstrar este comportamento na dissertação, porém aplicação se encontra disponível para uso através da url <<https://juniorschen.github.io/Amplisoft-Teclado-Virtual-Dinamico>> sem nenhum tipo e custo, e a utilização destes tipos de controles podem ser verificados.

Acerca dos resultados dos testes automatizados temos primeiramente demonstrado na Tabela 4, os testes do *layout* de duas linhas operado por *joystick*.

Tabela 4 - Resultados dos Testes Automatizados de *Layout* Vertical dos Controles Joystick

Controle	Loop	VP	wpm	cpm	ppm	epm
Joystick	0	600	21.70	108.54	10.93	1.44
Joystick	1	600	21.83	109.19	10.74	0.96
Joystick	2	600	24.13	120.65	11.78	1.58
Joystick	3	600	25.20	126.03	12.47	1.20
Joystick	4	600	26.37	131.87	12.97	0.95
Joystick	0	1200	31.07	155.37	15.22	1.90
Joystick	1	1200	32.86	164.30	16.20	1.47
Joystick	2	1200	32.70	163.51	16.08	1.56
Joystick	3	1200	32.86	164.32	16.57	1.52
Joystick	4	1200	35.30	176.53	17.34	0.25

VP: Velocidade Ponteiro; WPM: Palavras por Minuto; CPM: Caracteres por Minuto; PPM: Palavras Preditas por Minuto; EPM: Erros por minuto;
Fonte: Autor.

Os resultados dos testes automatizados do *layout* de duas linhas operado por controles sensoriais são exibidos na Tabela 5.

Tabela 5 - Resultados dos Testes Automatizados de *Layout* Vertical dos Controles Sensoriais

Controle	Loop	Delay	wpm	cpm	ppm	epm
Sensorial	0	2000	5.48	27.40	2.72	0.46
Sensorial	1	2000	5.68	28.42	2.76	0.46
Sensorial	2	2000	5.78	28.90	2.87	0.42
Sensorial	3	2000	5.81	29.05	2.90	0.40
Sensorial	4	2000	5.97	29.85	3.03	0.36
Sensorial	0	500	15.74	78.74	7.77	0.68
Sensorial	1	500	15.32	76.61	7.69	1.11
Sensorial	2	500	15.72	78.62	7.77	0.57
Sensorial	3	500	15.79	78.96	7.88	0.57
Sensorial	4	500	15.45	77.27	7.67	0.45

Delay: Tempo em milissegundos; WPM: Palavras por Minuto; CPM: Caracteres por Minuto; PPM: Palavras Preditas por Minuto; EPM: Erros por minuto;
Fonte: Autor.

Os resultados dos testes automatizados do *layout* de quatro linhas operado por controles *joystick* são exibidos na Tabela 6.

Tabela 6 - Resultados dos Testes Automatizados de *Layout* Horizontal dos Controles

Controle	Loop	VP	wpm	cpm	ppm	epm
Joystick	0	600	17.88	89.40	8.75	0.65
Joystick	1	600	17.16	85.81	8.48	0.88
Joystick	2	600	17.44	87.22	8.57	0
Joystick	3	600	19.08	95.40	9.37	0.71
Joystick	4	600	19.13	95.66	9.40	0.69
Joystick	0	1200	20.86	104.31	10.35	1.21
Joystick	1	1200	21.04	105.22	10.35	1.06
Joystick	2	1200	21.21	106.05	10.65	1.09
Joystick	3	1200	28.17	140.86	14.04	1.91
Joystick	4	1200	30.41	152.08	14.94	0.65

Joystick;

VP : Velocidade Ponteiro;; WPM: Palavras por Minuto; CPM: Caracteres por Minuto; PPM: Palavras Preditas por Minuto; EPM: Erros por minuto;

Fonte: Autor.

Por fim apresentamos os resultados dos testes automatizados do *layout* de quatro linhas operado por controles sensoriais na Tabela 7.

Tabela 7 - Resultados dos Testes Automatizados de Layout Horizontal dos Controles Sensoriais

Controle	Loop	Delay	wpm	cpm	ppm	epm
Sensorial	0	2000	5.48	27.40	2.72	0.46
Sensorial	1	2000	5.68	28.42	2.76	0.46
Sensorial	2	2000	5.78	28.90	2.87	0.42
Sensorial	3	2000	5.81	29.05	2.90	0.40
Sensorial	4	2000	5.97	29.85	3.03	0.36
Sensorial	0	500	15.74	78.74	7.77	0.68
Sensorial	1	500	15.32	76.61	7.69	1.11
Sensorial	2	500	15.72	78.62	7.77	0.57
Sensorial	3	500	15.79	78.96	7.88	0.57
Sensorial	4	500	15.45	77.27	7.67	0.45

Delay: Tempo em milissegundos; WPM: Palavras por Minuto; CPM: Caracteres por Minuto; PPM: Palavras Preditas por Minuto; EPM: Erros por minuto;
 Fonte: Autor.

Complementando temos também a tela onde os usuários podem enviar *feedbacks* relacionados à utilização do teclado virtual e sugestões de melhorias conforme a Figura 31, sendo um campo livre digitação.

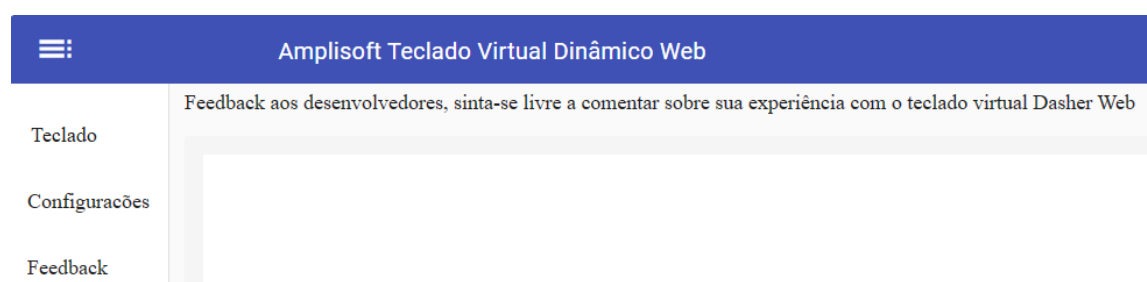


Figura 31 - Tela *feedback* finalizada.
 Fonte: Autor.

Adicionamos como funcionalidade extra a possibilidade de customização de *layout*, onde o usuário pode mover e redimensionar cada bloco de ação livremente. Para realizar essas mudanças na tela do teclado virtual adicionamos um botão superior direito que quando acionado habilita o modo de edição como podemos observar na Figura 32, após as alteração o usuário aciona o botão de salvar e o layout fica salvo conforme customizado, é possível retornar para os padrões nas configurações pela opção “Tipo do Layout” vista anteriormente.

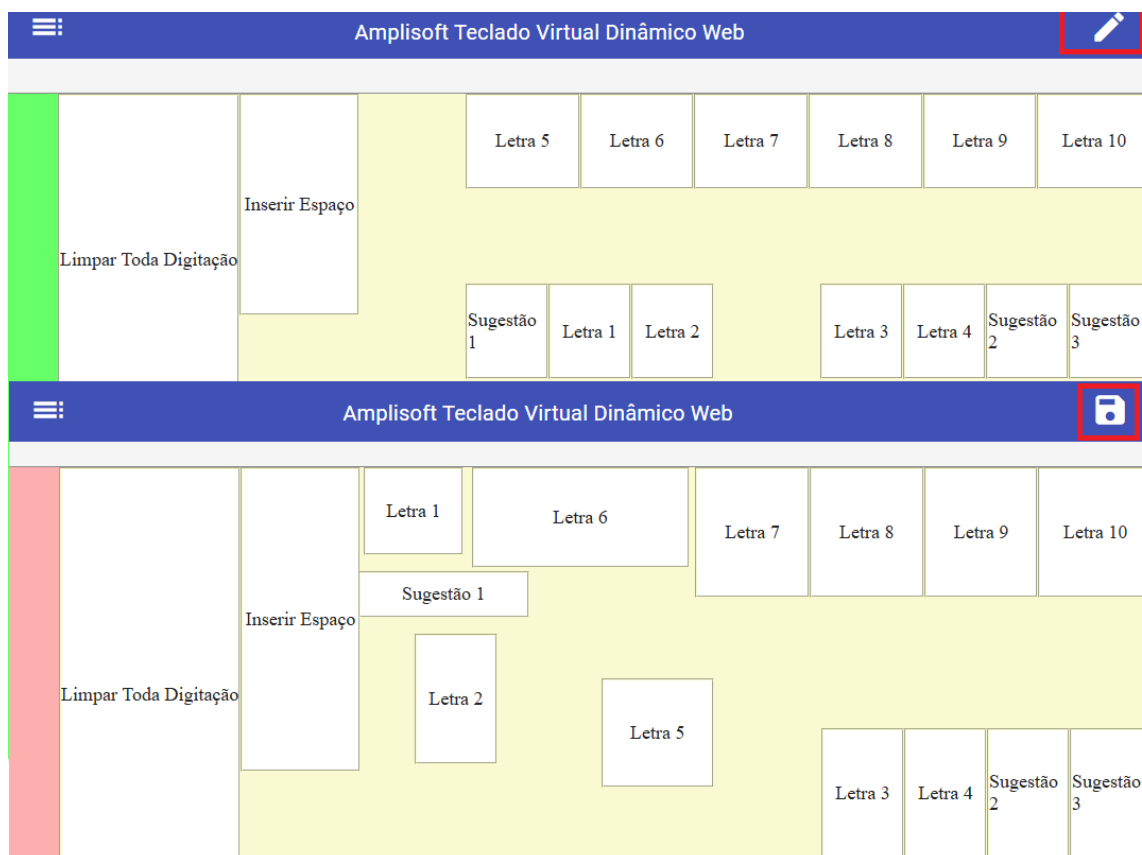


Figura 32 - Customização de Layout.

Fonte: Auto

5. DISCUSSÃO

Inicia-se a discussão dos resultados apresentados, lembrando que não é possível realizar simulações precisas dos perfis dos usuários, especialmente ao considerar deficiências físicas específicas. Os testes automatizados e os resultados apresentados tratam do valor que a aplicação pode entregar ao usuário. No desenvolvimento de *software*, esses testes devem sempre ser executados antes do contato do usuário com a aplicação.

5.1 Metodologia

Apresentou-se diversas abordagens durante o desenvolvimento do aplicativo, como possibilidades de configurações em todos os aspectos, variando de como o *layout* é exibido e como ele pode ser controlado por diversos dispositivos. Os métodos adotados para o desenvolvimento levaram em conta as melhores funcionalidades já disponíveis para utilização sem custos agregados, evidentemente levando em conta os usuários alvos, pessoas que possuem algum tipo de deficiência motora. É um desafio enorme prever as necessidades específicas de cada pessoa; contudo, o melhor método primário é a validação por meio dos testes automatizados tendo uma base sólida para viabilizar as funcionalidades existentes desenvolvidas.

O desenvolvimento da aplicação teve como base única e exclusivamente a linguagem de programação Javascript, sendo disponibilizada em ambiente web. Dentre os projetos que visitamos, apenas HENZEN & NOHAMA (2016) se apoiaram neste tipo de ambiente também. Outros métodos adotados como de WARD & MACKAY (2002) e JORDAN *et al.* (2020) apresentam o desenvolvimento a nível do sistema operacional, as diferenças entre estes métodos normalmente está vinculada aos níveis de controle, onde aqueles desenvolvidos a nível do sistema operacional tem maiores vantagens na manipulação de funções nativas de sistema, que não são disponibilizadas ao *browser* do ambiente web por questões de segurança. No entanto, ao utilizar a api HID podemos capturar os dados de dispositivos conectados via Bluetooth ou USB, onde por meio destes temos acesso a todo necessário para portar dispositivos alternativos ao mouse e teclado, e ao mesmo tempo temos vantagem do ambiente web onde podemos liberar atualizações a aplicação sem depender de ações manuais do usuário tornando este procedimento transparente. Também ressaltamos que atualmente a linguagem Javascript é uma das mais utilizadas atualmente, tendo uma comunidade ativa, e

recebendo constantemente atualizações com adições de novos recursos e melhorias nos já existentes, fornecendo uma flexibilidade enorme aos desenvolvedores que optam pelo desenvolvimento em web, assim como possibilidade de utilização das aplicações em computadores, tablets e celulares.

Relacionado aos métodos para coleta de indicadores de desempenho optamos por nos apoiar na medição da eficácia de digitação em relação ao tempo, que são utilizadas de forma geral para medição de performance de teclados virtuais, como podemos observar nos trabalhos de GOMIDE et al (2016) e WARD & MACKAY (2002). Apresentamos também um método de coleta de *feedbacks* que não foi relatado pelos outros autores que acompanhamos, este indicador se torna complementar aos de desempenho, visto que por meio dele o usuário pode se expressar e transmitir os pontos positivos e negativos que tiveram durante a utilização da aplicação, fatores que podem ser relevantes visto que as deficiências motoras são adversas, e os indicadores numéricos não podem expressar este tipo de situação.

5.2 Resultados

Conforme apresentado na primeira coleta de dados automatizados na Tabela 3, pode-se perceber que os indicadores de desempenho do *layout* da Figura 23 apresentaram médias inferiores ao projeto Dasher de WARD & MACKAY (2002). Ao analisar os *logs* de cada ação que a ferramenta de testes gerou, foi possível perceber que a complexidade das ações "Mostrar Mais", "Consoantes" e "Vogais" impactou negativamente a performance. Isso ocorreu porque o simulador de usuário precisava realizar diversas navegações para atender o texto proposto para digitação. Quando contabilizado o tempo dessas navegações, ficou evidente que elas eram um fator de extremo impacto nos indicadores coletados. O aplicativo Dasher original não possui esse tipo de ação, apresentando todas as letras para captura do usuário conforme demonstrado na Figura 4. No entanto, essa abordagem não foi difundida devido à complexidade gerada. Tentou-se expor as ações em grandes espaços na tela para manter a premissa dinâmica de maneira simplificada, o que comprometeu a performance devido à necessidade de mais ações.

Desenhamos dois modelos padrões de *layout* para a aplicação, um disposto de duas linhas Figura 24 e outro de quatro linhas Figura 25, o propósito central dessas duas abordagens foi ter um *layout* com menos elementos em tela concentrando as informações apenas em duas linhas, e também um *layout* um pouco mais familiar ao

padrão QWERTY disposto por quatro linhas. Os resultados dos *layouts* foram demonstrados nas Tabelas 4 e 6. O layout quatro linhas teve desempenho inferior, devido a necessidade de navegação extra quando necessário ações mais distantes. Contudo, os resultados ainda foram satisfatórios, considerando que essa abordagem poderá ser mais amigável ao usuário, que são aspectos que os testes automatizados não conseguem simular. Decidimos então além dos modelos padrões possibilitar que o usuário possa customizar o *layout* Figura 32 da melhor forma para que possa atender às suas necessidades, tendo em mente que diversos fatores, principalmente relacionados ao controle motor, devem afetar qual será o melhor modelo.

Os resultados dos testes de controles operados por *joystick* mostram que a aplicação pode superar as médias de digitação de um teclado convencional operado por uma pessoa sem deficiência motora em ambiente simulado, atingindo médias de até 35 wpm, conforme apresentado na Tabela 4. Um teclado convencional atinge médias de 30 wpm e o teclado Dasher usado como inspiração, até 23 wpm (WARD & MACKAY, 2002; HOSTE & SIGNER, 2013).

Quanto aos resultados dos testes operados por controle sensorial mostram que esse tipo de controle realmente apresenta performances inferiores aos operados por *joystick*. Contudo, é importante ressaltar que esses controles normalmente serão utilizados em casos mais extremos, onde as médias encontradas em ambiente simulado, conforme descrito na Tabela 5, de até 15 wpm, podem ser promissoras.

Temos também a interpretação por meio das tabelas dos impactos dos indicadores em e ppm, onde em todos os casos apresentados nas tabelas 4, 5, 6 e 7 podemos perceber que a taxa de wpm tende a aumentar a cada incremento de loop devido ao algoritmo de predição textual que incrementa o ranking de palavras a cada execução, em contrapartida temos casos os loops superiores apresentaram menor média de wpm que execuções anteriores, ao analisarmos o indicador epm destes registros percebemos que sua variação de ocorrência aumentou sendo, mostrando assim que as vantagens da predição de palavras, tendo em média apresentada taxas próximas a 50% de melhoria de eficiência e também como taxas mais altas do indicador de epm podem afetar o desempenho do teclado.

5.3 Comparativo com abordagens existentes

Podemos traçar pequenos paralelos com trabalhos visitados nas referências bibliográficas, onde comumente todas as abordagens se referem a formas de solucionar problemas de pessoas com deficiências motoras, propondo algoritmos de predição textual, variados desenhos de layouts do teclado virtual e métodos de controles distintos. Contudo as abordagens acabam se distinguindo diversamente, sejam pelos diferentes métodos adotados para construção do teclado virtual, ou pela forma como a eficiência é medida e também o propósito final da aplicação, sendo assim não é possível realizar comparações diretas entre os trabalhos.

Se tratando do Dasher WARD & MACKAY (2002) que foi utilizado como referência para nosso trabalho, temos diversas semelhanças, sendo elas, a forma dinâmica da exibição do layout, operar o teclado por meio de controles alternativos e coleta de indicadores de desempenho semelhantes. Contudo mesmo para a abordagem deles não é viável realizar comparações visto que o DASHER foi testado com usuários reais e treinados atingindo médias de 23 wpm. Enquanto nossa abordagem foi conduzida através de testes automatizados atingindo médias de 35 wpm, onde o cenário que esse desempenho foi atingido se distingue.

5.4 Limitações

Como já discutido anteriormente, os principais fatores que limitam este trabalho são cenários em que usuários reais serão colocados para uso da aplicação, visto que todos os testes foram realizados em ambiente controlado automatizado, as principais limitações que compreendemos são:

1. Testes Automatizados versus Testes com Usuários Reais: Os testes automatizados não podem simular adequadamente o uso humano, como preferências pessoais e dificuldades específicas de cada deficiência motora. É essencial realizar testes com usuários reais para obter uma compreensão mais precisa da eficácia e usabilidade do teclado virtual.
2. Variabilidade das Deficiências Motoras: As deficiências motoras variam amplamente entre os indivíduos, e um dispositivo que funciona bem para uma pessoa pode não ser adequado para outra.
3. Fatores Cognitivos e Emocionais: A aceitação e a eficácia do teclado também dependem de fatores cognitivos e emocionais que não foram abordados nos testes

automatizados. A motivação, a curva de aprendizado e a frustração dos usuários são aspectos que precisam ser considerados em estudos futuros.

5.5 Trabalhos Futuros

Para avançar na pesquisa e desenvolvimento de teclados virtuais assistivos, traçamos as seguintes direções para melhoria e continuidade do projeto no futuro:

1. **Elaboração de Protocolos:** Tendo em vista que em testes automatizados obteve-se resultados satisfatórios o próximo passo seria a distribuição e acompanhamento de testes com usuários reais; no entanto, quando se leva em conta fatores como a variedade de deficiências motoras e fatores cognitivos deve-se primeiro levantar protocolos de como serão aplicados esses testes antes de sua execução.
2. **Testes com Usuários Reais:** Testes com usuários reais utilizando os protocolos estabelecidos para coletar dados detalhados sobre a usabilidade, eficiência e aceitação do teclado virtual, este tipo de teste deve ser antes aprovado pelo Comitê de Ética em Pesquisa.
3. **Integração com Outras Tecnologias Assistivas:** Explorar a integração do teclado virtual com outras tecnologias assistivas, como sistemas de reconhecimento de voz e dispositivos de rastreamento ocular, para proporcionar uma experiência de uso mais rica e acessível.
4. **Melhorias no Aplicativo:** Após a fase de testes e coleta de *feedbacks* é esperado que diversas melhorias sejam identificadas para execução e melhor aceitação do aplicativo.

6. CONCLUSÕES

A conclusão deste trabalho reflete a concretização bem-sucedida dos objetivos estabelecidos, evidenciando a eficácia das soluções desenvolvidas para a melhoria da interação com teclados virtuais. Este projeto visou não apenas o avanço técnico na criação de controles alternativos e funcionalidades de predição de palavras, mas também a avaliação do desempenho do software através de testes automatizados. Ao revisar os resultados obtidos, observamos que todas as metas específicas foram atingidas, contribuindo significativamente para a acessibilidade e usabilidade de teclados assistivos. A seguir, detalha-se a realização dos objetivos específicos desenvolvimento:

1. Validação da Ferramenta Via Testes: Por meio de testes automatizados de máquina, verificou-se a aplicação atinge médias de digitação de até 35 palavras por minuto, que supera a média de 30 palavras por minuto dos teclados convencionais, assim como as médias relatadas de 23 palavras por minuto do teclado virtual assistivo Dasher, permitindo afirmar que o aplicativo pode ser testado com usuários reais que apresentam deficiências motoras.
2. Desenvolvimento de Controles Alternativos: Foram desenvolvidos meios de controles alternativos que permitem a interação com o teclado virtual utilizando diferentes dispositivos HID, como joysticks de diferentes modelos e modos de operação, assim como controle ocular e a possibilidade de uso de qualquer dispositivo que controle um mouse convencional.
3. Implementação de Predição de Palavras: A funcionalidade de predição de palavras foi integrada ao teclado virtual, melhorando significativamente a velocidade de digitação. A predição não apenas sugere as palavras mais prováveis com base nas entradas do usuário, mas também reorganiza a interface para otimizar a eficiência da digitação.
4. Coleta de *Feedback*: Um método para coleta de feedback foi disponibilizado, permitindo aos usuários enviarem suas sugestões e problemas encontrados.
5. Estabelecimento de Indicadores de Desempenho: Foram estabelecidas e adicionadas formas de coleta dos indicadores de desempenho (wpm, ppm, cpm, epm, ipm).

7. CONTRIBUIÇÕES DO TRABALHO

Este trabalho apresenta contribuições importantes no campo dos teclados virtuais assistivos e da tecnologia assistiva em geral, que podem ser classificadas nas seguintes categorias:

7.1 Contribuições Tecnológicas

Desenvolvimento de *Software*: A criação de um teclado virtual dinâmico em plataforma web que supera as médias de digitação de teclados convencionais em condições específicas.

Portabilidade e Atualização: O projeto foi desenvolvido em um ambiente web, permitindo atualizações constantes sem necessidade de ação por parte do usuário e garantindo a portabilidade para novos dispositivos HID (Human Interface Device).

7.2 Contribuições Comunicacionais

Acessibilidade e Recurso de Fala: A adição deste teclado virtual em ambiente web poderá auxiliar usuários na sua comunicação diária, seja pela digitação ou por meio do recurso de fala auxiliando a comunicação no dia a dia, melhorando significativamente a interação social, especialmente para pessoas com deficiência motora.

***Feedback* e Personalização:** A implementação de funcionalidades para coleta de *feedback* e personalização do *layout* do teclado, permitindo ajustes que atendem melhor às necessidades individuais dos usuários.

7.3 Contribuições Assistivas

Apoio a Deficiências Motoras: Fornecimento de uma ferramenta assistiva eficiente que permitirá a indivíduos com deficiências motoras severas utilizar um teclado virtual com maior eficácia e eficiência.

Modelos Preditivos: Utilização de modelos preditivos para melhorar a velocidade e precisão da digitação, com potencial para futuras melhorias por meio da incorporação de inteligência artificial para análise de contexto.

Por fim, essas contribuições podem fornecer uma base sólida para futuras pesquisas e aplicações práticas, sendo acessível no repositório de código GitHub dos autores <<https://github.com/juniorschen/Amplisoft-Teclado-Virtual-Dinamico>>, ou será disponibilizada a quem desejar, a partir de autorização da Hotmilk/PUCPR.

REFERÊNCIAS

UC BERKLEY EXTENSION. 11 Most In-Demand Programming Languages. [S.I.] 2023. Disponível em < <https://bootcamp.berkeley.edu/blog/most-in-demand-programming-languages/> >. Acesso em: 13 maio. 2023.

PENSANDO O DIREITO. A internet como fator de inclusão. [S.I.] 2015. Disponível em <<http://pensando.mj.gov.br/marcocivil/pauta/a-internet-como-fator-de-inclusao/>>. Acesso em: 13 maio. 2023.

CYPRESS. Javascript Component Testing E2E Testing Framework. Disponível em <<https://www.cypress.io/>>. Acesso em: 13 maio. 2023.

KEMP, S. DIGITAL 2022: GLOBAL OVERVIEW REPORT. DATA REPORTAL. [S.I.] Janeiro de 2022. Disponível em <<https://datareportal.com/reports/digital-2022-global-overview-report>>. Acesso em: maio.2023

ZAHERA, S. Accessibility Principles. Developed by the Education and Outreach Working Group (EOWG). 2019, p.1-1. Disponível em <<https://www.w3.org/WAI/fundamentals/accessibility-principles/>>. Acesso em: 13 maio. 2023.

MATTHEW, E. Syntactic pre-processing in single-word prediction for disabled people. Ph.D. Thesis, Univ. of Bristol, England, 1996.

PREECE, J., ROGERS, Y., SHARP, H. Interaction Design, beyond human-computer interaction. Wiley. 2th edition. 2007, p.1-34.

WESSELS, R., DIJCKS, B., SOEDE, M., GELDERBLOM, G J. and De WITTE, L. Non-use of provided assistive technology devices: A literature overview. Technol. & Disabili. Vol 15, 2003, p.231-238. doi: 10.3233/TAD-2003-15404

YASUOKA, K; YASUOKA, M. "On the Prehistory of QWERTY". ZINBUN. N° 42, 2011, p.161–174. doi: 10.14989/139379

LYNDS J.S. DARCI TOO-a computer input device for people with disabilities. IEEE, Proceedings of the Johns Hopkins National Search for Computing Applications to Assist Persons with Disabilities. 2002, p.493-493. doi: 10.1109/CAAPWD.1992.217430

WARD, D. J.,MACKAY, D. J. C. (2002) Fast hands-free writing by gaze direction. Nature.Vol 418, 2002, p.838-838. doi: 10.1038/418838a

WARD, D. J., BLACKWELL, A. F., & MACKAY, D. J. C. (2000). Dasher—a data entry interface using continuous gestures and language models. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology* (p. 129–137). Association for Computing Machinery. doi: 10.1145/354401.354427

HOSTE, L., SIGNER, B. SpeeG2: a speech-and gesture-based interface for efficient controller-free text input. In: Epps J, Chen F, Oviatt S, Mase K, editors. Multimodal Interaction. Proceedings of the 15th ACM on International Conference on Multimodal Interaction. 2013, Sydney, Australia. New York: ACM. 2013, New York, p.213-213. doi: 10.1145/2522848.2522861

H. CECOTTI, Y. K. MEENA, B. BHUSHAN, A. DUTTA, G. PRASAD "A multiscript gaze-based assistive virtual keyboard," 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 2019, p.1306-1309, doi: 10.1109/EMBC.2019.8856446.

R. A. SILVA, A. C. P. VEIGA. Algorithm for decoding visual gestures for an assistive virtual keyboard. IEEE Latin America Transactions, vol. 18, no.11, p.1909-1916, 2020, doi: 10.1109/TLA.2020.9398632.

R. ISLAM, S. RAHMAN, A. SARKR. "Computer Vision Based Eye Gaze Controlled Virtual Keyboard for People with Quadriplegia," 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), Rajshahi, Bangladesh, 2021, p.1-6, doi: 10.1109/ACMI53878.2021.9528213.

H. S. DHILLON, R. SINGLA, N. S. REKHI, R. JHA. EOG and EMG based virtual keyboard: A brain-computer interface. 2009 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, 2009, p.259-262, doi: 10.1109/ICCSIT.2009.5234951.

L. N. S. ANDREASEN STRUIJK. An Inductive Tongue Computer Interface for Control of Computers and Assistive Devices. IEEE Transactions on Biomedical Engineering, vol. 53, no. 12, p.2594-2597, 2006, doi: 10.1109/TBME.2006.880871.

L. MAYAUD, S. FILIPE, L. PETEGNIEF, O. ROCHECOUSTE, M. CONGEDO. Robust Braincomputer interface for virtual Keyboard (RoBIK): project results. Innovation and Research in BioMedical engineering, 2013, p.131-138. doi: 10.1016/j.irbm.2013.01.013

M. JORAN, N.N.N. GUILHERME, B. ALCEU, N. PERCY. Virtual keyboard with the prediction of words for children with cerebral palsy. Comput. Methods Prog. Biomed. 2020. doi: 10.1016/j.cmpb.2020.105402

M. CONGEDO, M. GOYAT, N. TARRIN, G. IONESCU, L. VARNET, B Rivet. Brain Invaders: a prototype of an open-source P300-based video game working with theOpenViBEplatform.In: Proceedings ofthe 5th InternationalBrainComputer Interface Conference; 2011, p.280–283.

S. GANNOUNI, K. BELWAFI, M.R. AL-SULMI, M.D. AL-FARHOOD, O.A. AL-OBAID, A.M. AL-AWADH, H. ABOALSAMH, A. BELGHITH. A Brain Controlled Command-Line Interface to Enhance the Accessibility of Severe Motor Disabled People to Personal Computers. Brain Sci. 2022. doi: 10.3390/brainsci120709

D. ANDREAS, H. SIX, A. BLIEK, P. BECKERLE. Design and Implementation of a Personalizable Alternative Mouse and Keyboard Interface for Individuals with Limited Upper Limb Mobility. Multimodal Technol. Interact. 2022. doi: 10.3390/mti612010

GOMIDE, R. S., LOJA, L. F. B., LEMOS, R. P., flôres, E. L., MELO, F. R., TEIXEIRA, R. A. G. A new concept of assistive virtual keyboards based on a systematic review of text entry optimization techniques. Research on Biomedical Engineering. 2016, p.176–198. doi: 10.1590/2446-4740.01715

COOK, A. M., HUSSEY, S. M. Assistive Technologies: Principles and Practices. Mosby. 5th edition. 2020, p.1-15. doi: 10.1093/occmed/kqab177

MOLINA, A. J., RIVERA, O., GÓMEZ, I. Measuring performance of virtual keyboards based on cyclic scanning. In: Calinescu R, Liberal F, Marin M, Herrero LP, Turro C, Popescu M, editors. Hoboken: IEEE. 2009, p.1357-1360. doi: 10.1109/ICAS.2009.35

HAMARASHID, H. K., SAEED, S. A., RASHID, T. A. A comprehensive review and evaluation on text predictive and entertainment systems. Soft Comput. 2022, p.34-35. doi: 10.48550/arXiv.2201.10623

DARGAN, S., KUMAR, M., AYAGARI M. R., KUMAR, G. A
Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning. Archives of Computational Methods in Engineering. 2019, p.1071–1092. doi: 10.1007/s11831-019-09344-w

GHOSH. Designing an efficient virtual keyboard for text composition in Bengali. In: Tripathi S, editor. Human Computer Interaction. India HCI 2011: Proceedings of the 3rd International Conference on Human Computer Interaction. 2011, Bangalore, India. New York: ACM. 2011. p.84-84. doi: 10.1145/2407796.2407809

WOBBROCK. Measures of text entry performance. In: MacKenzie IS, Tanaka-Ishii K, editors. Text entry systems. Burlington: Morgan Kaufmann. 2007. p.47-74. doi: 10.1016/B978-012373591-1/50003-6

NICOLAU H., GUERREIRO T., LUCAS D., JORGE J. Mobile text-entry and visual demands: reusing and optimizing current solutions. Universal Access in the Information Society. 2013, p.291-301. doi: 10.1007/s10209-013-0319-y

SOUZA F., NOGUEIRA R., LOTUFO R. (2020). BERTimbau: Pretrained BERT Models for Brazilian Portuguese. In: Cerri, R., Prati, R.C. (eds) Intelligent Systems. BRACIS 2020. Lecture Notes in Computer Science(), vol 12319. Springer, Cham. doi: 10.1007/978-3-030-61377-8_28

DEVLIN J., CHANG M.W., LEE K., TOUTANOVA K. BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018. doi: 10.48550/arXiv.1810.04805

HENZEN A., NOHAMA P. "Adaptable virtual keyboard and mouse for people with special needs," 2016 Future Technologies Conference (FTC). San Francisco, CA, USA. 2016, p.1357-1360. doi: 10.1109/FTC.2016.7821782

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, č.; POLOSUKHIN, I. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017, p.6000–6010. doi: 10.5555/3294996

S. SZETO. REHABILITATION ENGINEERING AND ASSISTIVE TECHNOLOGY. Biomedical Engineering, 2015, p.211-254. doi: 10.1016/B978-0-12-238662-6.50007-0

GARAY-VITORIA, N., ABASCAL, J. Text prediction systems: A survey. *Universal Access in the Information Society*. Vol. 4. 2006. p.188-203. doi: 10.1007/s10209-005-0005

DAVIES M. (2011) Google Books Corpus. (Based on Google Books n-grams). Available online at <http://www.english-corpora.org/googlebooks/>. Based on: Jean-Baptiste Michel*, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden*. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science* 331 (2011) [Published online ahead of print 12/16/2010].

A. PAPOUTSAKI, A. GOKASLAN, J. TOMPKIN, Y. HE, J. HUANG, "The eye of the typer: a benchmark and analysis of gaze behavior during typing," in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ACM, 2018, p. 16.

P. M. FITTS. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*. 1954. doi: 10.1037/h00553

LÖWGREN, J.; STOLTERMAN, E. *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge: MIT Press, 2004.

POLACEK, O.; SPORKA, A. J.; SLAVIK, P. Text input for motor-impaired people. *Universal Access in the Information Society*, v. 16, p. 51-72, 2017.

WELTON, T.; CUNNINGHAM, J. P.; GAO, H.; YUILLE, A. L. A brain–computer interface for the Dasher alternative text entry system. *Universal Access in the Information Society*, v. 15, p. 77-83, 2016.

BIGHAM, J. P.; LADNER, R. E.; BORODIN, Y. The need for inclusive design in the age of wireless and wearable technology. *Journal of Assistive Technologies*, v. 11, n. 2, p. 63-78, 2017.

KAYE, H. S.; YEAGER, P.; REED, M. Web accessibility for people with disabilities: opportunities for social and work integration. *Journal of Disability Policy Studies*, v. 21, n. 1, p. 1-8, 2011.

LAZZARO, J.; KLEINER, B. The expanding role of assistive technologies in facilitating work for individuals with disabilities. *Journal of Assistive Technologies*, v. 13, n. 4, p. 203-217, 2019.

SZETO, G. The ergonomics of keyboard and mouse use in people with motor disabilities. *Ergonomics in Design*, v. 23, n. 1, p. 8-13, 2015.

BARKER, R.; KRUGER, S.; WATKINS, K. The role of assistive technology in promoting inclusion for people with disabilities. *Disability & Society*, v. 32, n. 3, p. 396-410, 2017.

SCHUR, L.; KRUSE, D.; BLANCK, P. *People with Disabilities: Sidelined or Mainstreamed?* Cambridge University Press, 2013.

VANDERHEIDEN, G. C. *Fundamentals of Accessible Design in the Digital World*. CRC Press, 2000.