



CENTRO UNIVERSITÁRIO SENAC  
CAMPUS SANTO AMARO

RODRIGO CARVALHO QUAGLIO MAGALHÃES

JOGOS ELETRÔNICOS: COMPUTADOR DE BAIXO CUSTO PARA PROMOÇÃO  
DA AUTONOMIA E INCLUSÃO SOCIAL - CRIANÇAS COM DIFÍCULDADES  
COGNITIVAS.

SÃO PAULO

2023

Rodrigo Carvalho Quaglio Magalhães

JOGOS ELETRÔNICOS: COMPUTADOR DE BAIXO CUSTO PARA PROMOÇÃO  
DA AUTONOMIA E INCLUSÃO SOCIAL - CRIANÇAS COM DIFICULDADES  
COGNITIVAS.

Trabalho de conclusão de curso apresentado  
ao Centro Universitário Senac – Santo  
Amaro, como exigência parcial para  
obtenção do grau de Bacharel em  
Engenharia da Computação, sob orientação  
dos professores Me. Sergio Tavares, Dra.  
Keli C. Vido.

SÃO PAULO

2023

## **AGRADECIMENTOS**

Gostaria de agradecer primeiramente aos meus pais, Alessandra Ruschel Carvalho Magalhães e Ricardo Quaglio Magalhães, que sempre me apoiaram do início ao fim, dando todo suporte nessa longa caminhada.

Agradeço imensamente aos meus amigos Gustavo de Castro, Matheus Fellone e Rodrigo Gomes por estarem ao meu lado em todos os momentos que precisei, me dando forças e conselhos, não só na faculdade como na vida. Além disso, um agradecimento especial para Beatriz Duque, que me ajudou na escolha do tema, acompanhou tudo de perto e incentivou de todas as formas para que fosse possível a realização desse trabalho. A todos, meu muito obrigado. Sem vocês, nada disso seria concluído, amo vocês.

Agradeço também ao Prof. Ms. Sergio Tavares, por abraçar a ideia e pela orientação com todo suporte e dicas durante as reuniões para o desenvolvimento do projeto.

E por fim, ao Senac e professores, por contribuir no meu desenvolvimento como profissional, proporcionando conhecimento e experiencias que levarei para vida toda.

## **RESUMO**

Com a evolução constante da tecnologia no mundo todo, cada vez mais os olhares da sociedade se voltam a encontrar soluções para problemas que a humanidade enfrenta há anos. A pouca inclusão social e autonomia para PCD, seja por ignorância da sociedade ou ferramentas disponíveis para auxiliar no desenvolvimento deles durante o dia a dia, acaba sendo objeto de estudo e realização nos tempos atuais. Com isso, esse trabalho visa utilizar o uso da chamada tecnologia assistiva para promover a autonomia e inclusão social de crianças com dificuldades cognitivas, com foco na Síndrome de *Down*. A forma que será realizado esse trabalho é por meio de jogos eletrônicos que sejam educativos e que tragam diversão, com um sistema que recompense todo conhecimento e aprendizado que a criança com Síndrome de *Down* adquirir. A plataforma onde o jogo será desenvolvido será um computador de baixo custo (*Raspberry PI*) que possa ser usado por hospitais, clínicas, professores e famílias que não possuem um poder aquisitivo elevado, buscando uma maior acessibilidade para todos.

**Palavras-chave:** Tecnologia assistiva; Síndrome de *Down*; Jogos eletrônicos; *Raspberry PI*; baixo custo.

## **ABSTRACT**

According to the uninterrupted evolution of technology around the world, the eyes of society are turning increasingly to seek the solution humanity's problems has been facing for years. The lack of social inclusion and autonomy for PCD's, whether due to society's ignorance or tools available to help in their development during their daily routines, ends up being the focus on study and realization nowadays. Thus, this report aims to describe how the assistive technology can promote autonomy and social inclusion of children with cognitive difficulties, particularly focused on *Down syndrome*. This project will be carried out through electronic games that are educational and enjoyable, that provide rewards for all knowledge that a child with *Down Syndrome* acquires. A platform where the game will be developed will be a low-cost computer (*Raspberry PI*) that can be used by hospitals, clinics, teachers, and families that do not have a high purchasing power, providing for everyone accessibility.

**Keywords:** Assistive technology; *Down's syndrome*; Electronic games; *Raspberry PI*; low cost.

## **LISTA DE FIGURAS**

Figura 2.1 - Ilustração das características de pessoas com Síndrome de Down .....	18
Figura 2.2 - Imagens do aplicativo meu ABC Down .....	33
Figura 2.3 - <i>PlayTable</i> .....	34
Figura 2.4 - <i>Raspberry PI model 3</i> .....	35
Figura 3.1 - Fluxograma da terceira parte do projeto focado nos jogos.....	39
Figura 3.2 - Assistente virtual e suas expressões.....	44
Figura 3.3 – Exemplo de uso do Canva.....	45
Figura 3.4 - Tela de código padrão para <i>Pygame</i> .....	47
Figura 3.5 -Exemplo de chamada de arquivo de imagem e som.....	47
Figura 3.6 - Identificar pasta atual .....	48
Figura 3.7 - Funções do <i>Pygame</i> .....	48
Figura 3.8 - Integração com controle .....	49
Figura 3.9 - Diagrama de sequência da plataforma de jogo inicial .....	50
Figura 3.10 - Classe de Ambiente .....	50
Figura 3.11 - Classe Jogo .....	51
Figura 3.12 - Vetor para chamar os jogos escolhidos.....	51
Figura 3.13 - <i>Loop</i> de início da plataforma de jogo.....	52
Figura 3.14 - Condições com as funções de evento do <i>Pygame</i> .....	52
Figura 3.15 - Condição do primeiro menu.....	53
Figura 3.16 - Condição de escolha para escolher ambiente.....	53
Figura 3.17 - Estrutura de condição para selecionar o jogo.....	54
Figura 3.18 - Algoritmo para determinação de letras que serão usadas .....	56
Figura 3.19 - Início da função de iniciar o jogo .....	56
Figura 3.20 - Função que adiciona o status de revelação .....	57
Figura 3.21 - Função que faz associação de letra e cartas desenhadas.....	57
Figura 3.22 - Funções de revelar e cobrir carta .....	58
Figura 3.23 - Função que faz a animação de revelar e cobrir a carta.....	58
Figura 3.24 - Chamada de imagens .....	58
Figura 3.25 - Variáveis importantes .....	59
Figura 3.26 - Chamada de funções para inicialização do jogo .....	59

Figura 3.27 - Exemplo de condição de direção.....	60
Figura 3.28 - Algoritmo de seleção de carta .....	60
Figura 3.29 - Função de vitória .....	61
Figura 3.30 - Diagrama de Classes .....	62
Figura 3.31 - Instancias de classes e sprites .....	62
Figura 3.32 - Exemplo de condição de avanço do tutorial .....	63
Figura 3.33 - Vetor com as sujeiras e seu posicionamento .....	63
Figura 3.34 - Gerando os valores de posicionamento .....	64
Figura 3.35 - Juntando o vetor de sujeira com o vetor de valores .....	64
Figura 3.36 - Condições de movimento da escova .....	65
Figura 3.37 - Condição para finalizar a limpeza de uma sujeira .....	65
Figura 3.38 - Final do jogo exibindo o troféu.....	66
Figura 3.39 - Classe de objeto imagem .....	67
Figura 3.40 - Criar os brinquedos .....	67
Figura 3.41 - Função de desenhar brinquedos na tela .....	68
Figura 3.42 - Condição de desenho para tela aproximada .....	68
Figura 3.43 - Função de identificação do <i>mouse</i> pressionado.....	69
Figura 3.44 - Função de identificação do <i>mouse</i> solto .....	69
Figura 3.45 - Função de identificação do <i>mouse</i> sendo arrastado .....	70
Figura 3.46 - Função de vitória que exibe troféu ao final .....	70
Figura 4.1 - Tela inicial e dos ambientes.....	74
Figura 4.2 - Telas de escolha e tutorial.....	76
Figura 4.3 - Telas de acertos, erros e final. ....	77
Figura 4.4 - Telas de início e tutorial inicial.....	78
Figura 4.5 - Mini game e tela pós finalizar escovação. ....	79
Figura 4.6 - Telas de início mostrando o quarto e aproximando do armário...80	80
Figura 4.7 - Telas de acerto de cada etapa mais o final onde tudo está organizado.....	81
Figura 4.8 - Jogos inicializados a partir de executável.....	83

## **LISTA DE TABELAS**

Quadro 2.1 - Descrição de elementos para criação de jogos .....	28
Quadro 2.2 - Especificações da <i>Raspberry PI</i> model 3 .....	35
Quadro 3.1 - Orçamento .....	40

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>12</b>
1.1. Contextualização .....	12
1.2. Problema de pesquisa .....	13
1.3. Justificativa .....	13
1.4. Objetivos .....	14
1.4.1. Objetivo Geral .....	14
1.4.2. Objetivos Específicos .....	14
<b>2 REFERENCIAL TEÓRICO.....</b>	<b>15</b>
2.1 Introdução as deficiências intelectuais .....	15
2.1.1 Características da deficiência intelectual .....	16
2.2 Síndrome de <i>Down</i> na história .....	16
2.2.1 Definição da Síndrome de <i>Down</i> .....	17
2.2.1.1 Características Físicas.....	18
2.2.2 Dificuldades dos Indivíduos com Síndrome de <i>Down</i> .....	19
2.2.2.1 Processos cognitivos dos portadores da SD.....	20
2.2.2.2 Dificuldade com a memória de curto prazo.....	20
2.2.2.3 Dificuldades de concentração .....	20
2.2.2.4 Dificuldade de Linguagem.....	21
2.2.3 Estímulos para indivíduos com Síndrome de <i>Down</i> .....	22
2.3 Jogos eletrônicos e o desenvolvimento cognitivo.....	24
2.3.1 Importância dos jogos eletrônicos .....	24
2.3.2 <i>Serious games e digital game-based learning</i> .....	26
2.3.3 Elementos para criação de jogos .....	27
2.4 Tecnologia assistiva .....	29
2.5 Jogos para crianças com Síndrome de <i>Down</i> .....	30

2.6 Trabalhos relacionados .....	32
2.7 Sobre a <i>Raspberry PI</i> .....	34
<b>3 METODOLOGIA .....</b>	<b>36</b>
3.1 Abordagem do método de pesquisa.....	36
3.1.1.1 Pesquisas quanto ao objetivo .....	37
3.1.1.2 Pesquisas quanto ao procedimento.....	37
3.1.1.3 Fonte da Pesquisa .....	38
3.2 Visão geral do projeto.....	38
3.2.1 Primeira etapa: Referencial Teórico e pesquisa exploratória .....	38
3.2.2 Segunda etapa: Desenvolvimento do jogo na <i>Raspberry PI</i> .....	39
3.3 Orçamento.....	40
3.4 Materiais.....	41
3.5 Softwares e linguagens .....	41
3.5.1 <i>Python</i> .....	42
3.5.2 <i>Pygame</i> .....	42
3.5.3 Visual Studio Code .....	43
3.6 Design, paleta de cores e sons. ....	43
3.6.1 <i>Canva</i> .....	45
3.7 Desenvolvimento do protótipo de jogos.....	45
3.7.1 Estruturação de programação usando <i>Pygame</i> .....	46
3.7.2 Fundamentos para a plataforma de jogos inicial .....	49
3.7.3 Fundamentos para escolha do jogo da Memória .....	54
3.7.4 Fundamentos para escolha do jogo de Escovar os Dentes .....	61
3.7.5 Fundamentos para escolha do jogo de arrumar o quarto.....	66
3.8 Planejamento de testes .....	71
3.8.1 Testes durante o desenvolvimento ou teste Alfa.....	71
3.8.2 Teste de funcionalidade .....	71
3.8.3 Teste de jogabilidade .....	72

<b>4    RESULTADOS E DISCUSSÕES .....</b>	<b>72</b>
4.1 Resultados após fases teste Alfa e teste de funcionalidade.....	72
4.1.1 Plataforma de jogo inicial .....	73
4.1.2 Jogo da memória.....	75
4.1.3 Jogo de escovar os dentes.....	77
4.1.4 Jogo de arrumar o quarto .....	80
4.2 Mudanças realizadas após os testes de funcionalidade.....	82
4.3 Expansão de utilização para outros sistemas operacionais .....	82
<b>5    CONSIDERAÇÕES FINAIS .....</b>	<b>84</b>
5.1 Dificuldades.....	85
5.2 Trabalhos futuros.....	85

## 1 Introdução

### 1.1. Contextualização

A partir de estudos científicos ligados ao desenvolvimento do cérebro humano desde o nascimento até sua morte, Herbert J. e Patrícia S. (1978) afirmaram que, o estímulo cognitivo em crianças é fundamental para o futuro na vida adulta e impactando diretamente na sua habilidade em linguagens (tanto fala quanto leitura), em ter raciocínio lógico, realizar operações matemáticas e até mesmo no seu controle emocional. Porém, com a dificuldade que várias crianças possuem no aprendizado, sejam elas físicas ou mentais, devido a deficiências que afetam o cérebro, é possível identificar peculiaridades para cada tipo, tornando assim a necessidade de desenvolver diferentes métodos de ensino para estimular da melhor forma possível e proporcionar uma vida adulta saudável.

Um dos métodos mais utilizados e eficientes é o de jogos focados em desenvolver e estimular as habilidades cognitivas em crianças. Com a chegada da tecnologia, observar-se excelentes resultados com uma maior profundidade nos jogos cognitivos eletrônicos. Pois, além de propor desafios que são ligados ao uso da memória, raciocínio lógico, criatividade, linguagem, entre outros, há formas de implementar vários sistemas de recompensas que são mais impactantes e interessantes para as crianças, já que se utiliza muito da imaginação e criatividade, estimulando o aprendizado e concentração.

Dessa maneira, o projeto regente propõe o desenvolvimento de um videogame, utilizando um computador de baixo custo, e jogos eletrônicos que auxiliam no desenvolvimento de crianças com deficiências cognitivas, estimulando suas habilidades linguísticas, emocionais, lógicas e matemáticas, criando um sistema onde elas se sintam recompensadas por estarem jogando, tornando a atividade natural e divertida.

## 1.2. Problema de pesquisa

Com a tecnologia cada vez mais, presente e tendo um papel de relevância na acessibilidade, esse trabalho selecionou como problema de pesquisa: É possível diminuir o custo ao promover aprendizado e inclusão social em crianças com deficiência cognitiva utilizando jogos eletrônicos em computadores de baixo custo?

## 1.3. Justificativa

A utilização de jogos eletrônicos para o estímulo cognitivo se torna cada vez mais presente no aprendizado de crianças, pois podem aprimorar percepção, resposta motora e habilidades cognitivas (BEDIOU, 2017) e sendo um importante aliado ao tema acessibilidade já que, há uma facilidade maior em introduzir pessoas com algum grau de deficiência cognitiva, com o uso da tecnologia assistiva (TA), por meio de algo divertido, focado e prático (ISMAILI, 2018).

Raskind (2000) Afirma que utilizando a persistência e o uso de tecnologias assistivas, pode melhorar e muito as crianças com dificuldades cognitivas atingir seus potenciais, mesmo que ela apresente dificuldades grandes no processo de aprendizagem. Além disso, pode promover uma independência maior das crianças se usar essas tecnologias como ajuda, com o incentivo dos pais e professores, onde as crianças com deficiência depositam sua confiança (RASKIND, 2000).

Para os portadores de Síndrome de *Down*, sua dificuldade está relacionada em reter informações em mente no curto e longo prazo, o que afeta a linguagem expressiva e receptiva, ou seja, no processamento linguístico (o que afeta leitura, escrita, matemática etc.) e em sua audição, como descrito nos estudos realizados por Bower e Hayes (1994). Porém, como foi investigado por Caycho e colaboradores (1991) por mais que existam dificuldades no processo de aprendizado em portadores de Síndrome de *Down*, eles possuem capacidade de desenvolver princípios cognitivos, como o de fazer contagens matemáticas, o que implica em uma chance de aprimoramento relativamente alta.

Enfim, esse trabalho busca apresentar maneiras que auxiliem o desenvolvimento de crianças com deficiências cognitivas, por meio de jogos eletrônicos, utilizando um computador de baixo-custo para serem usados em instituições ou até mesmo para os pais que não dispõem de recursos financeiros, apoiando-se em métodos estudados para trazer uma experiência simples e divertida, trazendo uma autonomia e inclusão social para a crianças.

#### 1.4. Objetivos

##### 1.4.1. Objetivo Geral

O objetivo geral desse projeto é o desenvolvimento de jogos eletrônicos que auxiliem e estimulem o aprendizado de crianças com dificuldades cognitivas - Síndrome de *Down*, utilizando um computador de baixo custo.

##### 1.4.2. Objetivos Específicos

- 1) Promover autonomia através da acessibilidade para estimular o aprendizado em crianças com SD e incluir elas socialmente.
- 2) Investigar por meio de uma revisão bibliográfica os possíveis métodos em jogos de estímulo cognitivo e inclusão social em crianças com Síndrome de *Down* (audições, visuais, linguagem etc.).
- 3) Analisar o uso de computador de baixo custo como o console para os jogos.
- 4) Desenvolver uma interação entre *hardware* e *software* no computador de baixo custo.

## 2 Referencial teórico

O referencial teórico tem como objetivo introduzir determinadas definições que levaram a concepção do trabalho. Para melhor compreensão do tema, o capítulo foi desmembrado em 3 partes – Importância dos jogos no desenvolvimento das crianças e adolescentes, definição e caracterização da Síndrome de *Down* e por último, como os *games* podem ajudar no desenvolvimento da SD.

### 2.1 Introdução as deficiências intelectuais

O foco do projeto não é aprofundar-se no estudo de doenças cognitivas, mas somente conceituá-las para melhor entendimento do trabalho.

A deficiência intelectual não é considerada uma doença ou um transtorno psiquiátrico, e sim um ou mais fatores que causam prejuízo das funções cognitivas que acompanham o desenvolvimento diferente do cérebro. (HONORA & FRIZANCO, 2008, p. 103)

Um dos trabalhos mais completos na definição dos transtornos mentais é O DSM-5 (Manual Diagnóstico e Estatístico de Transtornos Mentais) da APA (Associação Americana de Psiquiatria), que foi revisado em 2013.

Um transtorno mental é uma Síndrome caracterizada por perturbação clinicamente significativa na cognição, na regulação emocional ou no comportamento de um indivíduo que reflete uma disfunção nos processos psicológicos, biológicos ou de desenvolvimento subjacentes ao funcionamento mental. Transtornos mentais estão frequentemente associados a sofrimento ou incapacidade significativos que afetam atividades sociais, profissionais ou outras atividades importantes. Uma resposta esperada ou aprovada culturalmente a um estressor ou perda comum, como a morte de um ente querido, não constitui transtorno mental. Desvios sociais de comportamento (p. ex., de natureza política, religiosa ou sexual) e conflitos que são basicamente referentes ao indivíduo e à sociedade não são transtornos mentais a menos que o desvio ou conflito seja o resultado de uma disfunção no indivíduo, conforme descrito. (DSM-5 2013)

### 2.1.1 Características da deficiência intelectual

Normalmente as crianças com deficiência intelectual, apresentam um nível cognitivo baixo, dependem das outras pessoas para fazerem tarefas simples do dia a dia, possuem dificuldades de compreensão na realização de certas atividades, segundo o Instituto Neurosaber<sup>1</sup>.

Falando das crianças especificamente, elas possuem um tempo de aprendizagem diferente das crianças da mesma idade, precisando muitas vezes um acompanhamento especial e especializado.

Exemplificando alguns transtornos mentais ou atrasos cognitivos são: transtorno do X frágil<sup>2</sup>, TDAH<sup>3</sup>, autismo<sup>4</sup>, Síndrome de RETT<sup>5</sup>, Síndrome de *Down*, entre outros.

No caso da Síndrome de *Down*, que será o foco nesse trabalho, a deficiência intelectual está ligada a uma Síndrome genética.

### 2.2 Síndrome de *Down* na história

É um pouco difícil precisar uma data exata para os primeiros aparecimentos de Síndrome de *Down*. Na antiguidade, existem vários relatos que crianças ou adultos com alguma deficiência intelectual ou cognitiva eram abandonadas para morrer, ou mesmo mortos.

A primeira prova científica que temos é por volta do ano de 5.200 AC, através da descoberta de um esqueleto na Ilha das Rosas na Califórnia. Ao analisar o esqueleto, principalmente o crânio possuía características compatíveis com o que

<sup>1</sup>Instituto Neurosaber – Instituto que compartilha conhecimento sobre Neuroaprendizagem e Desenvolvimento Infantil. – [www.neurosaber.com.br](http://www.neurosaber.com.br) acessado 01 de junho 2021

<sup>2</sup> Transtorno do X Fragil: Condição genética e hereditária, ligada a desordem do cromossomo X

<sup>3</sup> TDAH (Transtorno do déficit de atenção e hiperatividade): Caracterizada por sintomas de desatenção, inquietude e impulsividade

<sup>4</sup> Autismo ou TEA (Transtorno do Espectro do Autismo): Condição que se caracteriza por déficit na comunicação social e comportamental

<sup>5</sup> Síndrome de RETT: Distúrbio genético e de neurodesenvolvimento raro que ocorre principalmente em meninas, causado pela mutação de um gene ou genes necessários para o desenvolvimento do cérebro.

hoje entendemos com indivíduos desta Síndrome – Face e olhos achatados, ossos e dentes pequenos.

Algumas culturas da história nos contam que a Síndrome de Down esteve presente em diversos continentes, e foram retratadas principalmente em forma de esculturas nos tempos mais remotos ou em pinturas a partir da idade moderna.

O pediatra inglês John Langdon Haydon *Down* foi o primeiro médico a descrever o fenótipo caracterizado por bochechas proeminentes, fenda palpebral oblíqua e estreita, lábios grossos, língua grande e grossa, nariz pequeno, alteração vascular e deficiência intelectual, em 1866 (DOWN, 1866; PATTERSON; COSTA, 2005). Dando o nome a esta Síndrome.

### 2.2.1 Definição da Síndrome de *Down*

A Síndrome de *Down* (SD), apesar de maior parte da população mundial saber da existência dessa deficiência, ainda há pouca inclusão social recorrente da vasta desinformação e preconceitos referentes a essa deficiência.

A primeira vez que SD foi descrita e definida foi em 1866 por John Langdon *Down*, no artigo *Down Syndrome and its pathogenesis: considerations about genetic determinismo*.

Não é de conhecimento científico o motivo da Síndrome de *Down* acontecer, porém sabe-se que ela ocorre quando há uma alteração inadequada nos cromossomos<sup>6</sup> do par 21, formando assim uma célula de 47 e outra de 45 cromossomos totais e não 46 em ambos. (TUNES; PIANTINO, 2006). Esse fenômeno é conhecido por “(...) trissomia<sup>7</sup> dos cromossomos 21 (...)”, ou seja, ao invés da duplicação dessa divisão cromossômica, ela é triplicada, e atinge 95% da população dos portadores de Síndrome de *Down*. Existem mais 2 tipos que podem ser identificados em portadores de Síndrome de *Down*, uma delas é a translocação

---

<sup>6</sup> Cromossomos: Longas sequencias de DNA que contém diversos genes e outras sequências de nucleotídeos

<sup>7</sup> Trissomia – distúrbio genético em que uma pessoa tem 3 cópias de um cromossomo em vez de 2

cromossômica, que está presente, em média, entre 3 e 4% dos casos de Síndrome de *Down*, que é como a trissomia, porém o cromossomo é ligado pelo braço longo de outro. A outra anomalia, é conhecida como mosaicismo, em que, é formada duas linhagens celulares, uma com 46 cromossomos e a outra possuindo 47 cromossomos (trissomia) (BULL; Committes on genetics, 2011).

A incidência de nascidos vivos com SD é estimada em 1 a cada 650 a 1000 gestações, independente de etnia, gênero ou classe social (MALT et al. 2013). Na primeira metade do século XX, a expectativa de vida das pessoas com SD era até o início da adolescência e na sua maioria institucionalizadas. Devido aos avanços na área da saúde e as melhores condições de vida em geral, essa expectativa aumentou para 60 a 65 anos de idade, e quase todas as pessoas com SD vivem com suas famílias (ZHU et al. 2013).

Devido a estas características a Síndrome de *Down* não é uma doença que pode ser tratada ou curada. Ela é uma condição da pessoa, onde se deve estimulá-la para obter o seu melhor desenvolvimento.

#### 2.2.1.1 Características Físicas

As pessoas com Síndrome de *Down* são facilmente identificadas pelas suas características física, como podemos ver na figura 1. Claro com menor ou maior intensidade em cada indivíduo.

Figura 2.1 - Ilustração das características de pessoas com Síndrome de *Down*



Fonte: disponível em <http://www.movimentoDown.org.br>

### 2.2.2 Dificuldades dos Indivíduos com Síndrome de *Down*

Os indivíduos com Síndrome de *Down* enfrentam vários problemas que influenciam no seu aprendizado durante cada fase da vida, desde o seu nascimento até a vida adulta. Entre as principais dificuldades podemos elencar:

- Dificuldade de se comunicarem – problemas com a fala e linguagem
- Dificuldades auditivas e visuais
- Problemas motores
- Problemas com a memória de curto prazo.
- Períodos de concentração menor que as demais crianças

Essas dificuldades que eles trazem consigo podem gerar obstáculos para inclusão deles na sociedade, principalmente quando criança, o que leva a um atraso em seu desenvolvimento para a vida adulta. Para Voivodic (2013) a Síndrome de *Down* possui como característica a deficiência mental, o que leva a um atraso em todas as áreas de desenvolvimento do indivíduo, sendo as principais habilidades afetas a audição, oratória, cognitiva, social e problemas de comunicação que independe da fala.

As pessoas com SD apresentam potencial de se desenvolver pela ampla neuroplasticidade que pode ser estimulada desde os primeiros meses de vida pela família, profissionais de saúde e educação (SILVA; KLEINHANS, 2006; BULL, 2011).

De acordo com o objetivo deste projeto de estimular a capacidade cognitivas dos portadores de SD, será aprofundado a dificuldade de linguagem e problemas de concentração e memória a curto prazo.

### 2.2.2.1 Processos cognitivos dos portadores da SD

O cérebro do indivíduo de SD apresentam algumas alterações físicas, estruturais e funcionais.

Flores (1991) e Trancoso (1997) descreveram o cérebro de um portador de SD com um volume e tamanho menor do que indivíduos normais. Além de anomalias nos neurônios, o que afetaria áreas do sistema nervoso responsável pela estrutura e na comunicação das redes neuronais. O que seria responsável pela fala, memória curto prazo e atenção

Na idade escolar as crianças terão dificuldades na compreensão de regras e instruções, apresentarão um vocabulário mais restrito, o que poderá atrapalhar o seu convívio social.

### 2.2.2.2 Dificuldade com a memória de curto prazo

A memória de curto-prazo (MCP) é considerada como a habilidade de “manter na mente” uma informação por um período curto (JARROLD; BADDELEY, 2001).

Baddeley (2003) ainda conclui que, a memória de curto prazo está diretamente ligada ao entendimento das informações recentes para auxiliar a memória de longo prazo na recuperação de informações já aprendidas. Este processo tem como objetivo permitir a realização de atividades e influenciam na capacidade de pensamento.

### 2.2.2.3 Dificuldades de concentração

Os portadores de SD tem mais dificuldade de fixar a atenção por períodos mais longos em uma só atividade, eles perdem a sua concentração rapidamente. Também tem dificuldade de realizar duas tarefas ao mesmo tempo, o que irá atrapalhar o seu acompanhamento de uma aula, onde precisa ouvir o que o

professor está ensinando, entender e muitas vezes escrever (TRONCOSO; CERRO, 1999).

#### 2.2.2.4 Dificuldade de Linguagem

Autores como Buckley e Bird (1994) dizem que as características no desenvolvimento cognitivo e linguísticos em crianças com 5 anos de idade, mostra uma alta dificuldade em compreensão linguísticas, o que resulta em dificuldade para interpretar textos, reconhecer regras e sintáticas da língua, como também afeta a fala, pois com o vocabulário reduzido, as expressões dessas crianças acabam sendo transmitidas de outras formas e não apenas pela linguagem. Ainda segundo Buckley e Bird (1994) a linguagem acaba trazendo outras dificuldades, já que é diretamente ligada ao desenvolvimento de outras habilidades cognitivas, com a compreensão sendo afetada, a interpretação e o raciocínio para desenvolver pensamentos e memorizá-los acaba levando mais tempo e complexadas.

Segundo o Ph.D kumin, Libby no seu livro *Early Communication Skills for Children with Down Syndrome: A Guide for Parents and Professionals*, 2012. Para desenvolver habilidades de fala e linguagem, as crianças precisam de algumas habilidades sensoriais e perceptivas básicas. Entre estas são as habilidades para ver, ouvir, tocar, provar e cheirar coisas e as pessoas em seu ambiente. Mas, eles também incluem dois sentidos menos conhecidos: 1) propriocepção, que é o sentido que informa o corpo da posição dos músculos e 2) o sentido vestibular, que é o sentido do movimento que nos permite manter o equilíbrio.

Um outro fator que influencia na linguagem as crianças com Síndrome de *Down* podem apresentar dificuldades no tato, o que está intimamente ligado ao desenvolvimento da linguagem. São dificuldades que têm a ver com a consciência sensorial, a hipossensibilidade ao toque, a hipersensibilidade ao toque ou a combinação desses problemas como descrito por Libby (2012).

Apesar de pesquisas indicarem que o domínio da memória verbal ser limitada em indivíduos com Síndrome de *Down* (COUZENS; CUSKELLY; HAYNES, 2011), é

possível estimular e aprimorar a capacidade de memória auditória verbal com intervenções (CONNERS, 2008).

Estudos feitos por Caselli (1998) Chapman, Hesketh e Kistler (2002), Laws e Bishop (2003) afirmam que há uma diferença entre as habilidades receptivas e expressivas, onde a expressiva é mais afetada. Isso pode gerar até mesmo frustrações nos indivíduos com Síndrome de *Down*, já que eles não conseguem se expressar da mesma forma que recebem as informações, levando a necessidade de uma atenção maior aos educadores para conduzir e evitar esse sentimento em crianças (FIDLER; NADEL, 2007).

### 2.2.3 Estímulos para indivíduos com Síndrome de *Down*

O movimento *Down*<sup>8</sup> traz um guia completo de como podemos estimular os portadores de Síndrome de *Down*, a partir do momento que os pais recebem o diagnóstico de Síndrome de *Down* no seu bebê, já devem procurar ajuda especializada. Quando antes iniciar a estimulação, melhor será o seu desenvolvimento e inclusão na sociedade.

A estimulação precoce pode ser feita através de carinhos, brinquedos, esportes e jogos, agindo como um complemento do desenvolvimento e educação das crianças.

Todas as dificuldades cognitivas e de linguagem que descritas acima, influencia na parte de estimulação da linguagem e fala.

Algumas ações podem influenciar no desenvolvimento da criança, segundo o movimento *Down*:

- Em inglês existe uma expressão que é muito apropriada para auxiliar a criança na sua forma de se expressar “*Take your time*” – que seria em tradução livre - leve o seu tempo. O que quer dizer espere o tempo de a criança responder, mesmo que isso demore um pouco mais que a

---

<sup>8</sup> Movimento *Down* – Instituto criado para atender crianças com Síndrome de *Down* [www.movimentoDown.com.br](http://www.movimentoDown.com.br) acessado 20 de maio de 2021

maioria das crianças, mas também é importante que você escute e preste atenção.

- Não use palavras difíceis, procure falar uma linguagem cotidiana que a criança esteja acostumada de maneira direta.
- Solicite que a criança repita o que você falar, para assegurar o entendimento.
- Explique de maneira clara, use recursos áudio visuais, se necessário.
- Estimule a criança a formar frases, evitando expressões monossilábicas.
- brincadeiras e jogos que envolvam imagens, palavras que estimulem a criança a formar palavras é uma ótima maneira de desenvolver a sua fala e linguagem
- A introdução de novas palavras pode ser feita através de brincadeiras onde a criança pode trabalhar de maneira lúdica a parte de relacionar e de memória – escolhendo a figura correta, falando o nome do objeto e relacionando com outros objetos.
- Ao ensinar algo novo assegure se que reteve o aprendizado anterior, sempre voltando ao que foi aprendido.
- As atividades devem ser curtas e oferecer recompensas, para que os portadores de SD se sintam estimulados a terminarem.
- O uso do computador facilita muito o desenvolvimento da linguagem escrita e falada, então fazer atividades no computador é uma poderosa ferramenta.

Por isso, é importante buscar formas de promover a inclusão social de crianças com Síndrome de *Down*, pois elas são capazes de aprender, através de estímulos que ajudem a desenvolver suas habilidades cognitivas. E como jogos podem divertir e ensinar, uma fonte ainda limitada talvez devido aos custos elevados. O que leva ao tema do projeto, que é desenvolver jogos eletrônicos utilizando computador de baixo-custo justamente para incentivar e promover inclusão social dessas crianças.

## 2.3 Jogos eletrônicos e o desenvolvimento cognitivo

Atualmente, com o avanço da tecnologia, mais pessoas estão recorrendo a diversão aos jogos eletrônicos, segundo a pesquisa realizada pela Pesquisa Game Brasil (2020) 73,4% dos brasileiros jogaram algum tipo de jogo eletrônico, não importando qual plataforma. Isso ocorre por conta de os jogos eletrônicos estarem mais acessíveis à medida que vão evoluindo, graças ao número gigante de empresas (como a Nintendo) que se disponibilizam em desenvolvê-los e anunciar-los para a sociedade, além da nova onda de jogos *mobile*, que podem ser jogados a qualquer hora em qualquer lugar, o que acaba proporcionando um aumento no tempo que se passa jogando, principalmente as crianças e adolescentes.

### 2.3.1 Importância dos jogos eletrônicos

Segundo Gros (2003) uma das portas de entrada ao fantástico mundo da tecnologia, por crianças e adolescentes, é dada por meio de jogos eletrônicos, pois o primeiro contato vem justamente através de um vídeo *game*.

Um dos fatores para isso ocorrer é que o mundo dos jogos eletrônicos engloba diversas formas de se jogar e leva cada jogador a aventuras diferentes, com objetivos únicos, o que os tornam atraentes aos olhos de quem assiste e proporciona experiências únicas para quem está jogando. Existem diversos estilos de jogos, tendo para todos os tipos de gostos. Alguns principais são: ação, aventura, estratégia, simulação, esportes, entre outros. Com isso chegam-se conclusões de que jogos podem sim ter aspectos educacionais, como Kubiaki destaca:

Os jogos são desenvolvidos para lazer e diversão, mas também podem ser utilizados com finalidade educacional por trazerem implícitos aspectos pedagógicos que ajudarão o aluno a construir ou pôr em prática conhecimentos, e trazer o desafio à fantasia e à curiosidade [...] (KUBIAKI, 2015, p. 19).

Dessa forma os jogos eletrônicos se viabilizam como um grande aliado ao desenvolvimento e aprendizado de crianças, conseguindo ampliar a capacidade

cognitiva e intelectual por meio de conteúdos estratégicos e importantes (GROS, 2003).

Schwartz diz em seu livro que (2014, p.36):

Há uma infinidade de jogos que testam memória e outras competências cognitivas, portanto ajudam a desenvolver o cérebro como se estivéssemos numa academia. Ou seja, não só existem jogos desenhados para ajudar em processos de ensino e aprendizagem como alguns títulos aparentemente fora do universo educacional podem ser criativamente adotados por professores e alunos.

As principais habilidades que os jogos eletrônicos são capazes de desenvolver segundo Pinto e Ferreira (2005) são: habilidades motoras; desenvolvimento de discernimento visual e atenção seletiva; desenvolvimento de lógica indutiva; desenvolvimento cognitivo em aspectos científicos e técnicos; e o desenvolvimento de habilidades complexas e indução de descobertas. Além disso, segundo Rizzi (1994) eles podem auxiliar no desenvolvimento de habilidades sociais, como respeito, solidariedade, cooperação, obediência, responsabilidade e iniciativa.

Johnson (2005) ainda constata algo interessante, pois enquanto os jogadores estão sendo incentivados a tomar decisões, fazer escolhas que podem mudar o rumo do jogo, já que as regras são apresentadas conforme o jogo segue, o jogador acaba aprendendo de forma natural jogando.

Crianças, muitas vezes, encontram dificuldades na hora de enxergar ou visualizar contextos fora da realidade que ela vive, o que torna o aprendizado de certas coisas mais difíceis e causa uma falta de interesse ou desvio de atenção do que é necessário aprender na escola, por exemplo. E nisso os jogos eletrônicos são ótimos, pois como Shaffer e Clinton (2004) afirmam:

Jogos são poderosas ferramentas para o aprendizado porque eles possibilitam criar mundos virtuais, facilitando a o entendimento das situações sociais, fortalecendo a identidade, compartilhando valores e formas de pensamentos das comunidades. (SHAFFER; CLINTON 2004, p.7).

Dessa forma, utilizar jogos eletrônicos para ensinar e criar interesse no que está sendo passado e fixar conteúdo nos estudantes, acaba sendo útil e traz um

leque de possibilidades para aprimorar o aprendizado. Segundo McClarty et al (2012) “Jogos são ferramentas importantes para engajar e ajudar os estudantes a mergulharem completamente no ambiente de aprendizado, focando na atividade que estão envolvidos. ”.

### 2.3.2 *Serious games e digital game-based learning*

Ao pensar em jogos eletrônicos para estudantes, ou seja, como uma forma educativa para desenvolver habilidades cognitivas, existem os chamados *serious games*. Que, segundo autores como Prensky (2003), Gee (2003) e Bellotti (2010) são jogos onde o foco é ensinar e não apenas divertir ou entreter.

Esse termo foi usado pela primeira vez por Abt (1970) para diferenciar jogos por diversão de jogos para aprender, e no contexto digital usado inicialmente por David Rejeski e Bem Sawyer em 2002. Para autores como Papastergiou (2009) e Sung, Hwang (2013) *serious games* pode proporcionar mais interesses e interações na hora de aprender, o que induz os estudantes a adquirir e compartilhar pensamentos, conhecimentos e experiências. Burguillo (2010).

Como *serious games* abrange áreas diversas, a forma de se referenciar aos jogos educacionais pedagógicos é chamado de *Digital game-based learning*, que segundo Prensky (2001) é a forma perfeita de juntar a mudança na forma de aprender no atualmente com a evolução da tecnologia e de gerar motivação para estudantes que nasceram em meio a tecnologia, Prensky (2001) ainda complementa que:

“Atualmente os estudantes de ensino fundamental que passarão pela faculdade, carrega com eles seu próprio *Game Boys*, *Handicams*, Celulares, CD portátil e *MP3 players*, *Pagers*, *Laptops* e conexão à internet. ”

Na época atual nos mostra exatamente esse ponto, pois a tecnologia está presente na rotina de todos os estudantes, a partir de seu celular, de vídeo *games*

portáteis ou não, internet móvel. Claro que devemos levar em consideração as condições financeiras de cada aluno, o que definirá o grau de utilização de cada um.

Dessa forma o desenvolvimento dos *Digital game-based Learning* é definido, tendo como objetivo, gerar interesse e motivação para os estudantes na hora de adquirir conhecimento (ERHEL; JAMET, 2013).

Obviamente, muitos educadores como (EGENFELDT-NIELSEN, 2006; FELICIA, 2009; WASTIAU et al. 2009; HWANG; WU, 2012), enxergam o uso de jogos como algo complicado para ensinar, pois não conseguem aplicar de maneira correta durante as aulas em sala de aula, já que o interesse de criança acaba sendo ligado a diversão e não ao aprendizado muitas vezes. Porém, Deubel (2006), diz que:

*Digital game-based learning* (DGBL), a união do conteúdo educacional com jogos de computador ou online, tem o potencial para uma grande variedade de aplicativos educacionais, desde que gerenciados adequadamente. Simplificando: motiva em virtude de ser divertido. É versátil, pode ser usado para ensinar quase todos os assuntos ou habilidades e, quando usado corretamente, é extremamente eficaz. Além do mais, seu uso é apoiado pela teoria construtivista, que exige envolvimento ativo e aprendizagem experiencial.

Se os jogos desenvolvidos forem bem fundamentados e usados corretamente, a efetividade é enorme, o que eleva a importância de estudos de como desenvolver jogos digitais de forma educacional, pois podem ser aliados indispensáveis no futuro para professores, assim como previu Prensky (2001).

### 2.3.3 Elementos para criação de jogos

Para Balasubramanian e Wilson (2006), que se basearam em estudos realizados no passado por Glazier (1973), Prensky (2001) e Rasmussen (2001) os jogos digitais possuem uma base que é necessária para todo tipo de jogos, sendo elas: 1) personagem; 2) regras do jogo; 3) metas e objetivos; 4) quebra-cabeças ou desafios; 5) Narrativa ou História; 6) interações dos personagens; 7) estratégias e 9) resultados.

Quadro 2.1 - Descrição de elementos para criação de jogos

Elementos	Descrição
personagem	Quem o jogador é dentro do jogo, ou seja, o personagem que o jogador irá controlar e fazer com que se sinta imerso no mundo do jogo.
Regras	O que define exatamente os objetivos e os conceitos do seu jogo, com o que o jogador pode ou não fazer.
Metas e objetivos	A motivação do jogador em estar jogando, seja procurando algo, construindo, ou apenas explorando a área, mas é o que estimula cada vez mais o jogador a jogar.
Quebra-cabeças ou desafios	Os tipos de desafio os jogadores irão encontrar para atingir os objetivos e metas proposto anteriormente
Narrativa ou História	Como em uma novela ou filme, jogos possuem suas próprias histórias e é o que faz os jogadores serem parte do universo do jogo, trazendo curiosidades. Alguns jogos dão possibilidade de jogos terem o desenrolar da história diferente a partir da decisão de cada jogador.
Interações entre personagens	São formas do jogo de criar seu próprio universo, para o jogador não se sentir sozinho. Geralmente são controlados pelo próprio jogo (ou seja, pelo computador).
Estratégias	Possíveis jeitos de jogar o jogo. Há jogos onde cada jogador faz sua estratégia, porém muitos são mais lineares
Resultados	Os impactos, benefícios e recompensas que o jogador terá por ter concluído uma missão ou por ter finalizado o jogo.

Fonte: Adaptado de Balasubramanian e Wilson (2006)

A partir dos elementos de criação de jogos descrita no quadro 1, podemos entender melhor a forma que os jogos são estruturados e desenvolvidos. Assim, podemos aplicar essa estruturação e englobar os requisitos necessários para estimulação cognitiva em crianças com Síndrome de *Down*, de forma educativa como Prensky (2001) pensava, além de gerar motivação como descrito por Erhel e Jamet (2013).

## 2.4 Tecnologia assistiva

A tecnologia assistiva (TA) é o termo usado atualmente para o uso de tecnologia para auxiliar e promover a inclusão de PCD's (Pessoas Com Deficiência), seja no trabalho, na escola, em casa ou em qualquer outro ambiente (GALVÃO, 2009).

Assim como Manzini (2005) descreve:

Os recursos de tecnologia assistiva estão muito próximos do nosso dia a dia. Ora eles nos causam impacto devido à tecnologia que apresentam, ora passam quase despercebidos. Para exemplificar, podemos chamar de tecnologia assistiva uma bengala, utilizada por nossos avós para proporcionar conforto e segurança no momento de caminhar, bem como um aparelho de amplificação utilizado por uma pessoa com surdez moderada ou mesmo veículo adaptado para uma pessoa com deficiência.

No Brasil a tecnologia assistiva é conceitualizada como:

Tecnologia Assistiva é uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social. (BRASIL, 2007).

Com a chegada do uso comum de computadores como forma de tecnologia assistiva, as chamadas TIC (Tecnologias de informação e comunicação) ganham espaços, podendo ser utilizadas como TA ou por meio de TA, assim como Galvão; Hazard, Rezende (2007) dizem:

Utilizamos as TIC como Tecnologia Assistiva quando o próprio computador é a ajuda técnica para atingir um determinado objetivo. Por exemplo, o computador utilizado como caderno eletrônico, para o indivíduo que não consegue escrever no caderno comum de papel. Por outro lado, as TIC são utilizadas por meio de Tecnologia Assistiva, quando o objetivo final desejado é a utilização do próprio computador, para o que são necessárias determinadas ajudas técnicas que permitam ou facilitem esta tarefa. Por exemplo, adaptações de teclado, de mouse, software especiais etc.

Juntamente com esse conceito, existem uma série de possibilidades que a tecnologia assistiva atinge, tendo sua classificação feita pela Norma Internacional

ISO 9999. No total, são 11 classes usadas para classificação, onde o uso de jogos eletrônicos para promover a inclusão social e desenvolvimento cognitivo segue as classes 05 “ajuda para treino de capacidades” classe 21 “ajudas para a comunicação, informação e sinalização” e a classe 30 “ajudas para a recreação” (ISO 9999, 2002). o que possibilita o estudo com o desenvolvimento de jogos eletrônicos utilizando um computador de baixo custo, pois se encaixa como uma tecnologia assistiva e pode promover autonomia e inclusão social para indivíduos com Síndrome de *Down*.

## 2.5 Jogos para crianças com Síndrome de *Down*

Crianças com Síndrome de *Down* necessitam de estímulos que agradem e mantenham o interesse para que impulsionem sua vontade de continuar realizando tarefas, principalmente as que são ligadas ao seu aprendizado. Para Gilmore e Cuskelly (2014) o planejamento educacional necessita o envolvimento direto do estudante, alinhando os interesses e níveis de habilidade de cada criança, além disso é importante despertar o sentimento de conquista por estar aprendendo, por meio de recompensas, o que aumenta o engajamento e sua motivação em sua aprendizagem. Como os jogos eletrônicos causam um envolvimento maior pela sua imersão, diminuindo as chances de fracassar e vincula o aprendizado aos objetivos e funções que são necessários para iniciar um novo domínio em alguma habilidade (JENKINS, 2003), é possível ver a utilização de vídeo games para engajar crianças com Síndrome de *Down* a aprender.

Segundo Fidler (2005) crianças com Síndrome de *Down* possuem facilidade no processamento visual-espacial, o que inclui a memória visual, integrações visuo-motoras e imitação visual. Como os jogos eletrônicos são uma combinação de imagens e sons, eles são ótimos para prender a atenção das crianças com SD, pois são bastante visuais, algo que estudiosos como Abbeduto (2003), Buckley, Bird, Sacks e Archer (2006) afirma ser uma área de aprendizado forte delas. Dessa forma, utilizar jogos com tarefas que utilizem visual-espacial irá estimular a atenção da criança e ajudar manter a vontade de aprender, aprimorando sua cognição. (AWH; VOGEL; OH, 2006; LEPSIEN; NOBRE; 2006).

Alguns estudos já realizados afirmam que algumas áreas de conhecimentos podem ser aperfeiçoadas em jogos, como matemática, física e linguagem (HAYS, 2005; KE, 2009), áreas nas quais indivíduos com Síndrome de *Down* acaba tendo um elevado nível de dificuldade, pois são áreas ligadas diretamente a habilidades cognitivas e linguísticas.

O desenvolvimento de jogos eletrônicos capazes de manter o foco de crianças com Síndrome de *Down* é um desafio, pois elas necessitam ter objetivos e recompensas que sejam interessantes o suficiente e que interligue o aprendizado por meio pedagógico e a diversão que toda criança necessita (VIGOTSKY, 1997)

Outro ponto de dificuldade, mas necessário para o desenvolvimento de jogos eletrônicos para crianças com Síndrome de *Down* é a adaptação dentro de cada jogo levando em consideração as características, necessidades e capacidades de cada indivíduo. Pois cada indivíduo com DS possui características diferentes um dos outros, tornando indispensável o acompanhamento dos pais e educadores na hora do aprendizado. (COMISSÃO EUROPEIA, 1999b).

Hanghøj e Brund (2010, p.17) afirmam:

O ensino baseado em jogos pode ser entendido como uma série complexa de escolhas pedagógicas, práticas e processos de construção de significado, que podem ser analisados através das noções complementares de papéis do professor, modalidades de jogo e posicionamento.

Para aprimorar o desenvolvimento das habilidades cognitivas, linguagens e memória, em crianças com Síndrome *Down*, os jogos eletrônicos precisam utilizar da repetição tanto em som quanto visual, recompensar essas crianças por estarem aprendendo, elogiar e incentivar são fundamentais para que os acertos e erros sejam compreendidos, e sempre relacionar o que está sendo aprendido com a realidade vivida por essa criança. (VIGOTSKY, 1997; GILMORE; CUSKELLY, 2014)

“A chave para estabelecer o nível de dificuldade no ponto onde o aluno pode ser desafiado e cumprir a tarefa com um suporte moderado”. (JALONGO, 2007, p. 401).

Segundo o portal *Incluo*, criado por Leonardo e Carolina Gontijo para promover inclusão social de indivíduos com Síndrome de *Down*, os games virtuais para SD devem estimular a coordenação motora fina, auxiliam no processamento de

informações e fazem conexões associativas da aprendizagem, trabalhando o processo de memorização, raciocínio lógico, autodomínio, proatividade dentre outras performances. Um dos exemplos citados é o jogo *Genius*, que estimula a atenção, a memória e a noção de cores.

## 2.6 Trabalhos relacionados

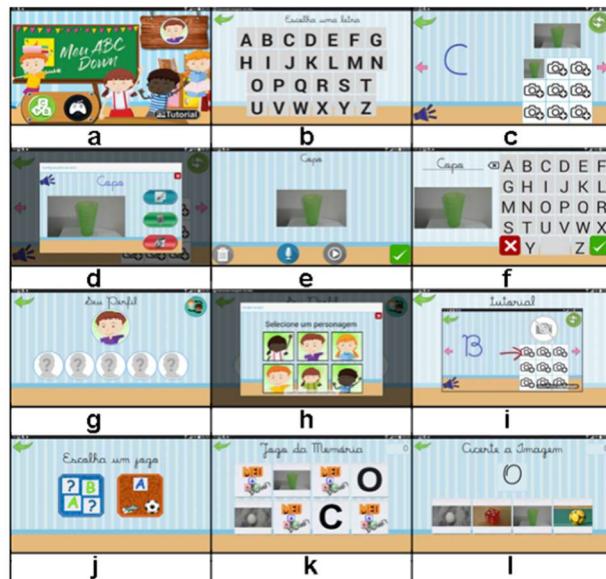
Atualmente, existem modalidades de jogos, com intuito de desenvolver habilidades cognitivas em crianças com Síndrome de *Down*, que utiliza diversas plataformas diferentes, como jogos *mobiles*, jogos que utiliza realidade virtual ou simplesmente jogos em formato de tabuleiro.

Na área de aplicativos mobiles, existe o chamado “o Meu ABC *Down*”, desenvolvido por Josimar e Rháleff (2017). Esse jogo está disponível para Android, onde a ideia é construir uma galeria multimidia personalizada com sons, imagens e textos, levando em consideração o contexto-social da própria criança. Nas palavras dos autores:

[..] a criança pode (i) capturar uma foto de um objeto a partir da câmera do tablet; em seguida (ii) escrever o nome deste objeto se utilizando de um teclado virtual que segue a ordem alfabética ABCDEF (adequada para o período de alfabetização) e; por fim (iii) gravar o áudio correspondente à imagem do objeto com o microfone do tablet. (JOSIMAR; RHÁLEFF,2017).

Além disso, existem dois jogos inclusos, sendo um “jogo da memória” e um jogo de “acerte a imagem”, que estimulam a memória e a coordenação motora respectivamente.

Figura 2.2 - Imagens do aplicativo meu ABC Down



Fonte: Disponível em <https://repositorio.ufpb.br/jspui/handle/123456789/3354>

O projeto *PlayTable*, criado pela empresa *PlayMove*, é atualmente o mais bem desenvolvido com o foco em utilizar a Ludopedagogia<sup>9</sup> para desenvolver habilidades cognitivas e motoras de crianças. Como afirma o dono da *PlayMove*, Marlon Souza (2016):

Através de pesquisas foi verificado que o aprendizado por meio dos jogos e aplicativos lúdicos facilita o desenvolvimento do raciocínio lógico, a memorização, a atenção, a paciência, a criatividade, a resolução de problemas, as linguagens de expressão e a coordenação motora fina das crianças.

A *PlayTable* funciona a partir de uma tela que reconhece o toque humano e de alguns objetos como plástico, metal e feltro etc. Além disso, os jogos desenvolvidos para a plataforma são feitos por outras empresas, aumentando o número de jogos disponíveis, que, por sua vez, podem ser jogos tanto para crianças que não possuem deficiências como para as que possuem.

---

<sup>9</sup> Ludopedagogia: é um segmento da Pedagogia dedicado a estudar a influência do elemento lúdico na educação. Ela é uma ferramenta para propósitos pedagógicos dentro das diretrizes educacionais vigentes.

Figura 2.3 - *PlayTable*



Fonte: Site PlayTable (2022)

## 2.7 Sobre a *Raspberry PI*

A *Raspberry PI* é um computador de baixo-custo, desenvolvida pela *Raspberry PI foundation*, que é uma fundação de caridade localizada no Reino Unido, com foco em educar e facilitar o uso da computação e ensinar a pessoas de qualquer idade, o uso da computação, por meio de programação (Raspberrypi, 2012). Ela utiliza-se do seu próprio sistema operacional baseado no kernel do Linux, chamado de *Raspbian*<sup>10</sup>, porém ela comporta diversos sistemas operacionais também, como Ubuntu, Windows, Solar, etc. Além disso, a um leque de linguagens que ela suporta, sendo as mais conhecidas: C/C++, Java e *Python* (Raspberrypi, 2012).

---

<sup>10</sup> **Raspbian OS** é uma distribuição Linux criada para rodar nos *Raspberry Pi*. Derivada do Debian, essa distro é considerada o sistema operacional padrão do computador da *Raspberry Foundation*.

Figura 2.4 - *Raspberry PI* model 3



Fonte: Site [Raspberrypi.org](https://www.raspberrypi.org) (2022)

Quadro 2.2 - Especificações da *Raspberry PI* model 3

CPU	Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
RAM	1GB RAM
Networking	BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
Internet	100 Base Ethernet
GPIO	40-pin extended GPIO
Portas	4 USB 2.0, 1 saída de audio 3.5mm estereo e HDMI
Armazenamento	Micro SD port for loading your operating system and storing data

Fonte: Adaptado do site *Raspberry PI* (2022)

### 3 Metodologia

O presente trabalho tem como objetivo demonstrar a viabilidade do desenvolvimento de um console utilizando um computador de baixo custo atendendo o público de baixa renda e com dificuldades cognitivas: Como este seria um universo amplo, tomou-se a decisão de direcionar as crianças com Síndrome de *Down* promovendo sua inclusão social e desenvolvendo a sua capacidade cognitiva. Dessa forma, além da revisão bibliográfica completa sobre a Síndrome de *Down* e os jogos eletrônicos apresentados nos capítulos anteriores, em conjunto com a pesquisa exploratória, conseguimos ter uma visão geral do que seria necessário para o desenvolvimento do projeto.

#### 3.1 Abordagem do método de pesquisa

A escolha do método de pesquisa foi fundamental para levantarmos as principais características e necessidades de desenvolvimento das crianças com Síndrome de *Down*.

Foi utilizado o método de pesquisa qualitativo exploratório.

Baseado nesta pesquisa, foi possível determinar como o jogo deveria se comportar ao ser jogado, e quais comandos deveríamos colocar para o melhor entendimento, e definir a parte de design

Segundo Minayo (2001), a pesquisa qualitativa é uma pesquisa ampla onde se foca em significados, motivos, aspirações, crenças, valores e atitudes, o que remete a uma análise mais profunda das relações, dos processos e dos fenômenos que não podem ser simplesmente reduzidos em números.

GIL (2007) classificou as pesquisas quanto aos seus objetivos e procedimentos.

### 3.1.1.1 Pesquisas quanto ao objetivo

- Pesquisa exploratória - O objetivo deste tipo de pesquisa é fazer um levantamento dos problemas, onde serão levantadas e construídas diversas hipóteses. Entre as pesquisas exploratórias elencadas por GIL (2007), encontramos
  - 1) levantamento bibliográfico;
  - 2) análise de exemplos que estimulem a compreensão
- Pesquisa descritiva - Este tipo de pesquisa exige um conhecimento prévio do entrevistador para poder determinar quais são os fatos a serem pesquisados. Alguns exemplos de pesquisa descritiva: estudos de caso, análise documental, pesquisa ex-post-facto. Segundo Triviños (1987), uma das críticas a este tipo de pesquisa pode haver uma análise anterior que pode vir a distorcer certos fatos, gerando dados imprecisos.
- Pesquisa explicativa – O foco é a identificação de fatores que indicam e geram os fatos. Este tipo de pesquisa vai buscar o porquê dos dados e/ou informações, com a análise dos resultados Gil (2007).

### 3.1.1.2 Pesquisas quanto ao procedimento.

- Pesquisa Bibliográfica: Realizada através de pesquisas de materiais já existentes, entre elas: livros, artigos científicos e pesquisas na Internet
- Pesquisa Documental: pode-se confundir com a anterior, mas a diferença está na sua natureza, estes documentos não tiveram nenhuma análise previa. Alguns exemplos são documentos de arquivos, de instituições do Governo, Igrejas, sindicatos etc. ou que foram emitidos, mas podem ser reanalizados como relatórios de resultados de empresas;
- Levantamento: Este tipo de procedimento é do questionamento direto do público-alvo, definido previamente, normalmente em amostras com

grande número de respondentes. Muito utilizado em pesquisas quantitativas.

### 3.1.1.3 Fonte da Pesquisa

A pesquisa foi realizada em portais especializados em portadores de SD, como :Movimento *Down*, e Mano *Down*, projetos acadêmicos, entre outros.

## 3.2 Visão geral do projeto

A realização do projeto foi em duas etapas macro. A primeira etapa realizou-se a pesquisa exploratória para fazer o levantamento para confecção dos temas a serem abortados. Onde coletou-se dados sobre comportamentos, interesses, práticas, estímulos etc. para ser possível desenvolver jogos pedagógicos que prendam a atenção dessas crianças. Na segunda etapa, traduziu-se o que foi aprendido na pesquisa, em linguagem de jogo.

### 3.2.1 Primeira etapa: Referencial Teórico e pesquisa exploratória

O ponto de partida do projeto foi a pesquisa exploratória através do referencial teórico, que está dividido em 3 etapas:

- 1) Deficiências cognitivas em crianças com Síndrome de *Down*
- 2) Jogos eletrônicos para desenvolvimento cognitivo
- 3) Jogos eletrônicos para crianças com Síndrome de *Down*

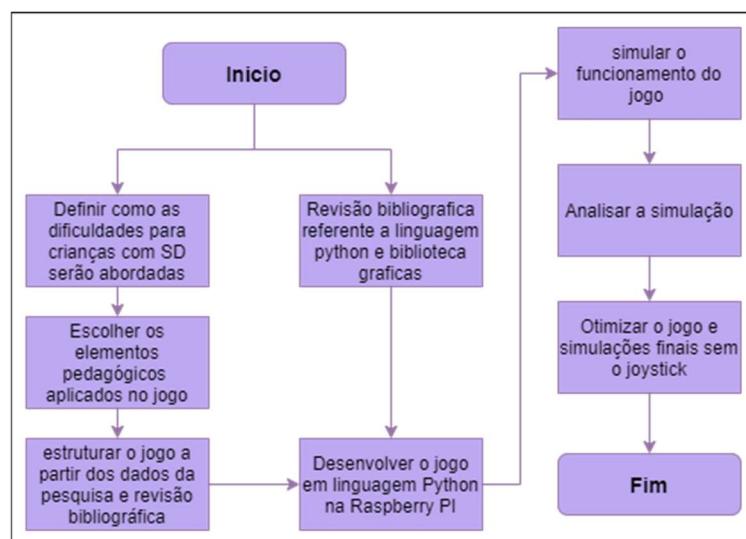
Através deste levantamento foi possível definir as hipóteses para os temas que iremos abordar nos jogos eletrônicos a serem desenvolvidos.

### 3.2.2 Segunda etapa: Desenvolvimento do jogo na *Raspberry PI*

Já na segunda etapa desenvolveu-se o jogo eletrônico para criança com Síndrome de *Down*. Para o jogo utilizou-se recursos de programação já conhecidos a partir da linguagem *Python*, na plataforma de um computador de baixo-custo (*Raspberry PI*) para processar os dados e apresentar as imagens em um monitor ou TV para que seja possível o usuário visualizar e jogar. O detalhamento do jogo encontra-se nos próximos capítulos.

A seguir, o fluxograma da terceira etapa utilizado para melhor entendimento do caminho seguido para o desenvolvimento do jogo eletrônico para crianças com Síndrome de *Down*.

Figura 3.1 - Fluxograma da segunda parte do projeto focado nos jogos



Fonte: Criado pelo Autor (2022)

Descrição das atividades realizadas:

- Primeira: Definição das dificuldades das crianças com SD que seriam abordadas no jogo e de que forma poderíamos estimulá-las. Além da revisão bibliográfica sobre linguagem *Python*, *Raspberry PI* e bibliotecas gráficas para encontrar a melhor forma para os desenvolvimentos de jogos.

- Segunda: Escolha dos métodos pedagógicos que foram estudados e mesclar com os elementos que são usados para criação de jogos.
- Terceira: Definição da estrutura do jogo, como fases, as cores, personalização, recompensas para cada objetivo alcançado, e como o assistente virtual seria utilizado.
- Quarta: O desenvolvimento do jogo a partir da linguagem *Python* e com a biblioteca gráfica escolhida, com a *Raspberry PI* de plataforma
- Quinta: Simular o funcionamento do jogo conforme for avançando no desenvolvimento.
- Sexta: Analisar se o jogo está bem estruturado, com fases e recompensas funcionais, intuitivas e prontas para ensinar.
- Sétima: Corrigir possíveis erros e trazer melhorias aos jogos.

### 3.3 Orçamento

O orçamento faz parte da execução do protótipo. A seguir, uma tabela simples do quanto foi destinado para cada produto que foram utilizados para o protótipo.

Quadro 3.1 - Orçamento

Produtos	Preço
<i>Raspberry PI 3 model B</i>	R\$635,00
Periféricos	R\$60,00
componentes	R\$70,00
Softwares	R\$0,00
MicroSD 32GB	R\$50,00
Total:	R\$815,00

Fonte: Criado pelo Autor (2022)

Os periféricos são referentes ao que é usado para interagir com o jogo. Que para esse trabalho foi utilizado um *mouse* e um teclado. Já os componentes são

cabos e a fonte que energiza a Raspberry PI que é usada para o funcionamento, incluindo a necessidade de se ter componentes reservas para caso houvesse alguma falha na construção. Os *softwares* utilizados para o desenvolvimento tiveram custo zero e o de edição de imagens também foi sem custos. O custo da Raspberry PI é uma média dos principais preços encontrados no mercado para esse produto, mais a adição de um microSD para maior capacidade de armazenamento.

### 3.4 Materiais

Para o funcionamento do protótipo a ser apresentado, o hardware escolhido como console foi a *Raspberry PI 3 Model B*. Como descrito, no referencial teórico, a quantidade de memória RAM de 1GB e clock 1,4 GHz foi o suficiente para processar a plataforma e os jogos que fossem desenvolvidos. Além disso, foi adicionado um cartão micro SD de 32 GB para armazenamento do sistema operacional *Raspbian* e os jogos. A *Raspberry PI* foi alimentada por uma fonte de 5 Volts e 3 Âmpères.

### 3.5 Softwares e linguagens

Com o *hardware* definido, a linguagem escolhida para o desenvolvimento dos jogos foi a linguagem *Python*. Na escolha da linguagem levou-se em consideração a o suporte da linguagem na *Raspberry PI*, através da IDE Thony, que já vem instalada juntamente o sistema operacional *Raspbian*. A partir disso, foi utilizada uma biblioteca em *Python* para desenvolver os jogos, chamada de *Pygame*.

No início, para desenvolver os jogos em linguagem *Python*, foi utilizado o Visual Studio Code, no sistema operacional Windows, a sua utilização é devido a maior familiarização com a plataforma, além de o Windows possuir maior suporte para *softwares* de edições de imagens, facilitando o desenvolvimento. E, após o desenvolvimento, os testes foram feitos inicialmente em uma máquina virtual com sistema operacional *Raspbian* (o mesmo que utilizado na *Raspberry PI*) para alguns ajustes e, em seguida, passadas para a *Raspberry PI* para os testes.

### 3.5.1 Python

*Python* é uma linguagem de programação geralmente utilizada para o desenvolvimento de sites, criação de softwares com orientação a objetos, análises de dados, *machine learning* etc. É considerada uma linguagem de alto nível e de fácil compreensão pela sua simplicidade. Por ser *open-source*<sup>11</sup> possui uma comunidade muito ativa e uma vasta gama de módulos e bibliotecas que ajudam no desenvolvimento de projetos mais complexos.

### 3.5.2 Pygame

Segundo a definição em seu site (<https://www.Pygame.org/>), o *Pygame* foi desenvolvido por Pete Shinners, e é um conjunto de módulos em *Python* designado para codificar jogos, adicionando as funcionalidades excelentes da biblioteca SDL, o que permite a criação de jogos e programas multimídia utilizando diversas ferramentas na linguagem *Python*. O *Pygame* é um projeto com licença GPL (*General Public License*). Ele é *open-source*, ou seja, é possível compartilhar e desenvolver jogos sem custos, que possui uma verdadeira portabilidade se tratando de sistemas operacionais, pois funciona nas principais, como: *Windows*, *MacOS* e *Linux*, mas também em outras não tão populares como Solaris e QNX. Ele utiliza de linguagem C a Assembly para ser mais otimizado e funcional.

Por ser uma biblioteca *open-source* e possuir uma licença de disponibilidade pública, além de sua facilidade e rapidez, o *Pygame* tornou-se a escolha que mais se encaixava ao objetivo, de ser um projeto de baixo-custo.

---

<sup>11</sup> O termo em inglês “**open source**” significa “código aberto” e se refere ao código-fonte de um site ou aplicativo. Nesse sentido, sua linguagem de programação pode ser vista por qualquer um, que pode adaptá-la para objetivos variados

### 3.5.3 Visual Studio Code

O Visual Studio Code é uma plataforma de edição de códigos, que tem como característica ser leve e de fácil acesso, pois ele possui disponibilidade de uso em diversos sistemas operacionais diferentes, como Windows, macOS e Linux. Sua principal característica é o suporte disponível para as diferentes linguagens de programação. Além de já vir com algumas linguagens em sua instalação, seu diferencial é a possibilidade de instalar extensões e compiladores de linguagem diferentes, facilitando o acesso a programação em uma única plataforma.

Visto isso, sua utilização no desenvolvimento dos jogos foi necessária, já que o suporte para *Python* é de uma grande eficácia.

## 3.6 Design, paleta de cores e sons.

A escolha do design, cores e sons é parte importante para os estudos e desenvolvimento de jogos, para definir como o jogo irá transmitir suas características e trará os jogadores para o seu ambiente. Para esse objetivo, os detalhes de maiores relevâncias vão além da narrativa e se encontram na escolha correta de design e de sons.

Para crianças que possuem deficiências, Landowska (2018), em sua contribuição no capítulo 8, para o livro guia “*guidelines for supporting children with disabilities*”, lista algumas características que os jogos digitais precisam seguir e a importância de alinhá-las a fim de obter os melhores resultados. Baseando-se nessas características em conjunto com as dificuldades enfrentadas pelas crianças com Síndrome de *Down*, os jogos seguiram determinadas diretrizes na escolha de design, cores e sons:

- Tela de fundo simples e com pouco detalhamento, com diferenciação entre o principal e o fundo.
- Cores primárias para objetivos principais, com cores vividas e claras.
- Interações intuitivas e de fácil memorização.
- Recuperação de erros.

- Fluidez sendo estática e linear, sem muitas opções diferentes para seguir.
- Sons que tragam retornos positivos juntamente com imagens para complementar.
- Som de fundo calmo, para trazer ambientação.
- Um assistente virtual, possuindo voz humana com tom expressivo, faces apropriadas e não-antropomórfico. (HERRING, 2017)

Dessa forma, a assistente virtual que está presente em todos os jogos foi criada com base nos itens listados acima. Ela possui expressões simples de felicidade, neutralidade e fala, como pode ser observado a seguir:

Figura 3.2 - Assistente virtual e suas expressões



Fonte: Criado pelo Autor (2022)

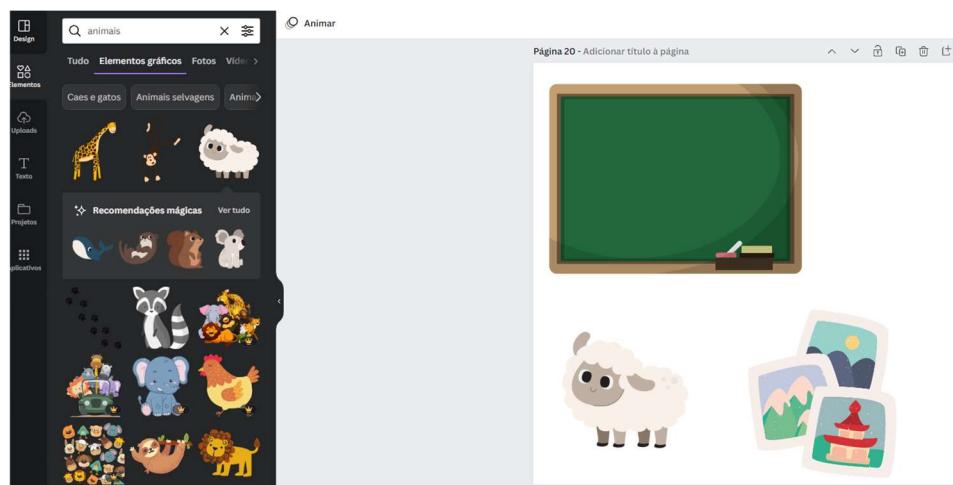
Assim, os designs utilizados nos jogos vieram de locais que disponibilizam imagens gratuitas para uso e editá-los conforme a necessidade. Da mesma forma que os sons de efeitos presentes nos jogos.

A música principal escolhida possui características que acalmam, relaxam e não agride a audição, foi desenvolvida do zero.

### 3.6.1 Canva

O Canva é uma ferramenta *online*, utilizada para criação de designs e que possuem uma vasta biblioteca de imagens já prontas e prontas para serem editadas. O diferencial do Canva, é ele disponibilizar uma parte dos designs de forma gratuita, ou seja, todos os direitos de imagem são de uso do próprio usuário, possibilitando utilizar para edições e adicionar em qualquer projeto.

Figura 3.3 – Exemplo de uso do Canva



Fonte: <https://www.canva.com>

O Canva foi utilizado para todo o design que estão presentes nos jogos, por ser uma ferramenta simples e de fácil acesso, não exige do usuário saber fundamentos avançados de design. Assim foi possível ir para o desenvolvimento do protótipo de jogos.

### 3.7 Desenvolvimento do protótipo de jogos

Os jogos foram estruturados com base nos estudos realizados com os elementos para criação de jogos descrita no referencial teórico. Assim, para o desenvolvimento do protótipo, ele foi dividido em duas etapas:

- Primeira etapa: O desenvolvimento de uma plataforma inicial onde pode selecionar o ambiente que a criança jogará (Escola ou Casa);
- Segunda etapa: O desenvolvimento dos jogos de acordo com o ambiente escolhido.

Para o ambiente Escola, foi desenvolvido o jogo da memória visando o aprimoramento da linguagem e memória cognitiva, através de elementos visuais, área na qual as crianças com Síndrome de *Down* possuem facilidade. Já para o ambiente Casa, foi desenvolvido um jogo que simula o passo a passo a ser seguido para criança escovar os dentes, auxiliando os pais no ensino de tarefas básicas, necessárias para higienização, além de proporcionar a inclusão social dessas crianças.

É importante que todos os jogos possuam formas de encorajar essas crianças com Síndrome de *Down* a continuar jogando, que ensine, de forma didática, porém simples, como realizar cada tarefa e o porquê de realizá-las, mostrando o progresso delas de forma visual e com muitos elogios. Dessa forma, a implementação de um assistente virtual para ensinar sobre os jogos, fazer os elogios para os acertos e encorajar quando houver erros, é essencial para facilitar a imersão da criança.

Além disso, o desenvolvimento de um sistema de recompensa, para estimular a criança querer continuar jogando e a jogar cada vez mais, pois a repetição é parte do processo.

### 3.7.1 Estruturação de programação usando *Pygame*

Como apresentado nas secções 3.5.1 e 3.5.2, toda programação dos jogos foram desenvolvidas em *Python* utilizando a biblioteca disponível chamada *Pygame*. Para utilizar o *Pygame*, é necessário fazer a importação da biblioteca a cada início de código e inicializá-la utilizando *Pygame.init()*. Dessa forma, a base para todos os jogos que foram desenvolvidos possui a seguinte estrutura:

Figura 3.4 - Tela de código padrão para Pygame.

```
#Importação de bliblioteca
import pygame
from pygame.locals import*

#Inicialização do pygame
pygame.init()

#Variaveis de Largura e altura de tela
largura=1000
altura=800

#Inicialização do display de tela
tela = pygame.display.set_mode((largura,altura))

#Inicialização do Clock
clock = pygame.time.Clock()
```

Fonte: Criado pelo Autor (2022)

O código acima informa a configuração básica de tela dos jogos e a função `Pygame.time.Clock()` inicializa o uso do `Clock` que é utilizado para atualização de frames por segundo dos nossos jogos, que são passados mais à frente no código.

Todas as imagens e sons que são usadas nos jogos, seguem a estruturação de chamada da seguinte forma:

Figura 3.5 -Exemplo de chamada de arquivo de imagem e som

```
#Informa qual som selecionar e passa para uma variavel
somEscolha=pygame.mixer.Sound(f"{path}\\\sons\\\selecionar.ogg")
#O mesmo para imagem, porem a função usada é diferente
paredeMenu = pygame.image.load_extended(f"{path}\\\imagens\\\menu.png")
```

Fonte: Criado pelo Autor (2022)

Vale destacar que todos os sons usados nos jogos são em formatos OGG (.ogg), pois foi o formato que se comportou da melhor maneira na *Raspberry Pi*.

Para o código saber onde procurar a imagem/som, foi utilizado a biblioteca de sistema operacional (OS) disponível para *Python*. Assim, é possível identificar em qual pasta os arquivos se encontram.

Figura 3.6 - Identificar pasta atual

```
# Identificador da pasta onde o jogo esta rodando em .exe ou .py
if getattr(sys, 'frozen', False):
    path = sys._MEIPASS
else:
    path = os.path.dirname(os.path.abspath(__file__))
```

Fonte: Criado pelo Autor (2022)

Após esses processos, todos os jogos desenvolvidos em *Pygame* necessita de, no mínimo, um *loop* para a tela permanecer aberta até que seja fechada por escolha do usuário ou seja concluído o que foi designado para o jogo. Cada jogo possui seus próprios algoritmos, porém todos irão possuir uma função que atualiza a tela e o seu frame por segundo definido.

Figura 3.7 - Funções do *Pygame*

```
#Atualiza a tela do jogo
pygame.display.update()
#Define a quanto frame por segundo
clock.tick(60)
```

Fonte: Criado pelo Autor (2022)

O *pygame* permite o uso de controle de videogame para interação nos jogos. Sua integração é realizada de forma simples, tendo suporte para diversos tipos de controles já disponíveis no mercado, como também suporte para criação de um caso necessário. Para os jogos, foi pensado em utilizar o controle e o teclado de forma simultânea. Dessa forma, o usuário pode usar tanto o controle quanto o teclado/mouse para a interação. As linhas de código a seguir, mostra que, para evitar conflitos, foi necessário criar uma classe para caso não tenha controle conectado, criar um falso para que seja interpretado que há um controle

Figura 3.8 - Integração com controle

```
class joysticksFalso(object):
    def __init__(self):
        pass
    def get_button(self,numero):
        pass
    joysticks = []
    for i in range(pygame.joystick.get_count()):
        joysticks.append(pygame.joystick.Joystick(i))
    for joystick in joysticks:
        print(joystick.get_name())
        joystick.init()
    if len(joysticks)==0:
        joystick = joysticksFalso()
```

Fonte: Criado pelo Autor (2022)

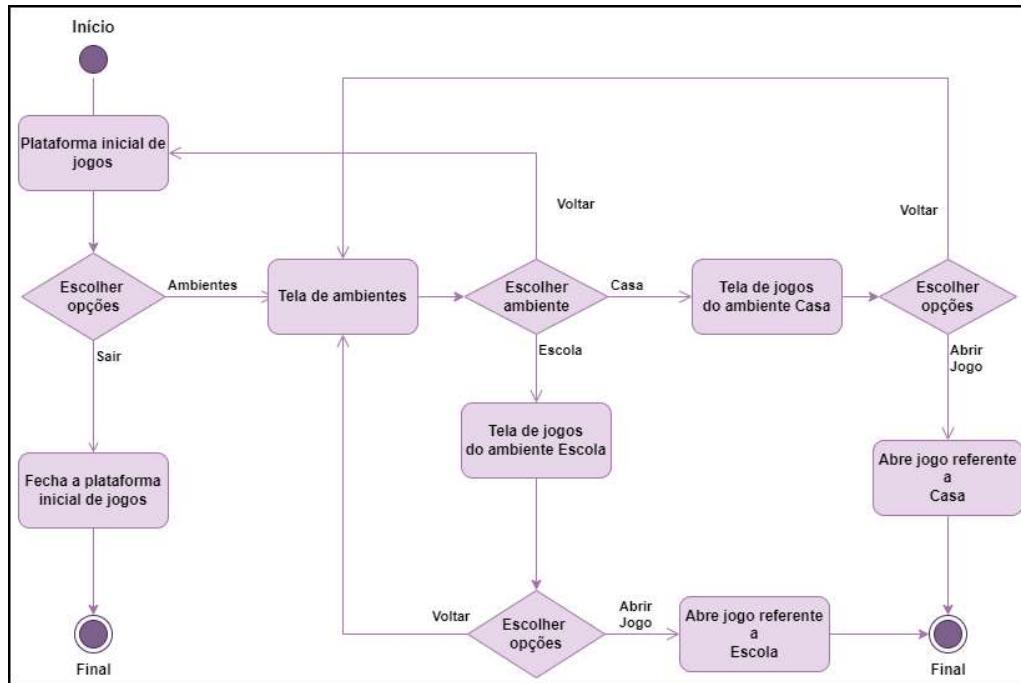
. Assim, a condicional analisa todos os controles que são identificados e o inicializa utilizando a função *Joystick.init()*.

Como cada jogo possui seu próprio comportamento, vamos descrever separadamente seus pontos mais importantes para o funcionamento de cada.

### 3.7.2 Fundamentos para a plataforma de jogos inicial

Para desenvolver um protótipo que funcione como ideia o uso de um console, é interessante analisar como os jogos são disponibilizados hoje em dia para cada console. Antigamente, os consoles utilizavam de fitas, CDs ou DVDs para serem acessados, o que fazia com que cada jogo aparecesse de forma individual para ser jogado para o usuário. Com a tecnologia avançando e o sistema de nuvem se tornando cada vez mais presente, foi pensado em desenvolver plataformas onde os jogos ficasse armazenados e disponíveis para serem acessados sempre que abrissem a plataforma, como se fosse uma biblioteca de jogos. A ideia inicial é de interação é descrita pelo diagrama de sequência a seguir:

Figura 3.9 - Diagrama de sequência da plataforma de jogo inicial



Fonte: Criado pelo Autor (2022)

De início, para estruturação da plataforma inicial de jogos, foi pensado em ter dois ambientes, sendo Escola e Casa. Dessa forma, criou-se uma classe para facilitar sua identificação, tendo como atributos de seu construtor nome e a posição que terá em tela. Após isso, foram instanciados os dois ambientes e adicionados em vetor.

Figura 3.10 - Classe de Ambiente

```

#Classe de ambiente
class Ambiente(object):
    fonte = None
    def __init__(self, nome,x,y):
        self.x = x
        self.y = y
        self.nome = nome

#Instancia de ambientes atribuída a um vetor
ambientes = []
ambientes.append(Ambiente("escola", 250, 200))
ambientes[0].fonte=fonte.render('ESCOLA', True, (255,0,0))
ambientes.append(Ambiente("casa", 550, 200))
ambientes[1].fonte=fonte.render('CASA', True, (255,0,0))
  
```

Fonte: Criado pelo Autor (2022)

Em seguida, foi criada uma classe Jogo, que terá o ambiente que ele faz parte, nome e posição na tela. Também possui um método que irá ser o responsável por abrir os jogos. Caso esteja rodando no Windows e seja aberto por um executável (.exe), irá abrir como um subprocesso do sistema. Caso seja aberto por extensão Python (.py) irá usar a função da biblioteca *rumpy*, que abre o jogo através do próprio compilador *Python*.

Figura 3.11 - Classe Jogo

```
#Classe jogos
class Jogo(object):
    fonte

    #Construtor
    def __init__(self, ambiente, nome, x, y):
        self.x = x
        self.y = y
        self.nome = nome
        self.ambiente = ambiente

    #Função de abrir os jogos
    def Jogar(self):
        if getattr(sys, 'frozen', False):
            modulo = f"{self.ambiente}\\"{self.nome}\\"{self.nome}.exe"
            subprocess.Popen(modulo, cwd=path)
        else:

            modulo = f"{self.ambiente}\\"{self.nome}\\"{self.nome}.py"
            #Personagem.exp = modulo.Jogar(ids, exp)
            rumpy.run_path(path_name=modulo)
```

Fonte: Criado pelo Autor (2022)

Dá mesma forma que o ambiente, instanciamos os jogos e colocamos em um vetor. A diferença é que, os jogos do ambiente escola vão para um vetor diferente dos jogos do ambiente Casa.

Figura 3.12 - Vetor para chamar os jogos escolhidos

```
#jogos do ambiente 1 (escola)
jogos = []
jogos.append(Jogo(ambientes[0].nome, 'jogo1', (largura/2)-40,(altura/2)))
#jogos do ambiente 2 (casa)
jogos1=[]
jogos1.append(Jogo(ambientes[1].nome, 'jogo1', (largura/2)-40,(altura/2)))
jogos1.append(Jogo(ambientes[1].nome, 'jogo2', 100,(altura/2)))
```

Fonte: Criado pelo Autor (2022)

O primeiro *loop* referente a tela inicial, possui algumas chamadas de funções do *Pygame* que são vistos em todos os jogos. Como exemplo, na imagem abaixo está sendo passado qual imagem a ser desenhada e em qual posição da tela. Além da criação do uso das palavras e suas fontes.

Figura 3.13 - Loop de início da plataforma de jogo

```
while not fechar:
    #definição de variável de fonte
    menu=fonte.render('OPÇÕES:', True, (255,0,255))
    ambiente=fonte1.render('Ambientes de jogos', True, (255,100,100))
    sair=fonte1.render('Sair', True, (255,100,100))
    tela.fill((0,0,0))
    #Aplica as imagens na tela
    tela.blit(paredePrincipal,(0,0))
    desenha_brinquedos(brinquedinhos)
    tela.blit(paredeMenu,(0,0))
    tela.blit(menu,((int(largura/15)),int(altura/1.5)))
    tela.blit(ambiente,escolha1xy)
    tela.blit(sair,escolha2xy)
    tela.blit(seta,setaxy)
```

Fonte: Criado pelo Autor (2022)

Já na imagem abaixo, temos um exemplo de uso da função *Pygame.event.peek(evento)*, que seria a forma do *Pygame* identificar qual evento está sendo realizado pelos periféricos, como o pressionar de uma tecla(ou botão) ou o clicar do *mouse*.

Figura 3.14 - Condições com as funções de evento do *Pygame*

```
#Identifica um evento de fechar a tela
if pygame.event.peek(QUIT)==True:
    fechar = True
#Identifica um evento de teclado ou de controle
if pygame.event.peek(KEYDOWN)==True or pygame.event.peek(JOYBUTTONDOWN)==True:
    #Identifica se o botão(seta) para baixo está sendo pressionado
    if pygame.key.get_pressed()[K_DOWN]==True :
        #limpa o buffer de eventos para não haver conflito
        pygame.event.clear()
        #Aciona o som de escolha
        pygame.mixer.Sound.play(somEscolha)
        #Mexe a seta de escolha
        setaxy = (escolha2xy[0]-75,escolha2xy[1]-10)
```

Fonte: Criado pelo Autor (2022)

Essa função é utilizada em todos os jogos de maneiras diferentes, com condições e atribuições próprias de cada jogo.

Assim, para selecionar e ir para tela de escolha de ambiente ou sair do jogo, a seta precisa estar posicionada na posição pré-estabelecida correta. Se estiver, o *loop* da primeira tela é encerrado e partimos para a próxima tela.

Figura 3.15 - Condição do primeiro menu

```
#Identifica se o espaço(teclado) está sendo pressionado
if pygame.key.get_pressed()[K_SPACE]==True:
    pygame.event.clear()
    pygame.mixer.Sound.play(somConfirmar)

    if setaxy == (escolha1xy[0]-75,escolha1xy[1]-10):
        print ("ambiente selecionado")
        azul=pygame.transform.smoothscale(pygame.image.load_extended(f"{path}\imagens\azul.png"),(220,220))
        fechar=True
        amb = True
    elif setaxy == (escolha2xy[0]-75,escolha2xy[1]-10):
        print ("sair selecionado")
        fechar = True
```

Fonte: Criado pelo Autor (2022)

A segunda tela é responsável por mostrar qual ambiente gostaríamos de ir “Escola” ou “Casa”. A base de estruturação da segunda tela é parecida com a da primeira, com a diferença que agora o código verifica qual ambiente está sendo selecionado a partir da posição passada previamente.

Figura 3.16 - Condição de escolha para escolher ambiente

```
if pygame.key.get_pressed()[K_SPACE]==True or joystick.get_button(0):
    pygame.mixer.Sound.play(somConfirmar)
    #Identifica se estamos selecionando o ambiente escola
    if vol == (ambientes[0].x-10,ambientes[0].y-20):
        print ("escola selecionado")
        seta1xy = (jogos[0].x-75,jogos[0].y)
        esc = True

    #Identifica se estamos selecionando o ambiente casa
    elif vol == (ambientes[1].x-10,ambientes[1].y-20):
        print ("casa selecionado")
        seta1xy = (jogos1[0].x-75,jogos1[0].y)
        cas = True
```

Fonte: Criado pelo Autor (2022)

Cada ambiente irá disponibilizar qual jogos referentes aquele ambiente pode ser jogado. O ambiente “Escola” possui apenas um jogo enquanto o ambiente “Casa” possui dois jogos. Assim que selecionado qual ambiente iremos, é iniciado o terceiro *loop* de tela.

O *loop* do ambiente Escola e do ambiente Casa são iguais em estrutura, o diferencial é a imagem de cada. Dessa forma, vale o destaque para a forma que o código define qual jogo está sendo selecionado.

Figura 3.17 - Estrutura de condição para selecionar o jogo

```

if pygame.key.get_pressed()[K_SPACE]==True or joystick.get_button(0):
    pygame.mixer.Sound.play(somConfirmar)
    #checha se estamos no em "voltar", caso sim, voltamos para os ambientes
    if seta1xy==(voltar.x,voltar.y+75):
        pygame.event.clear()
        cas= False
        amb= True
    #checha se estamos no jogo 1, caso sim, o jogo 1 é iniciado
    elif seta1xy==(jogos1[0].x-75,jogos1[0].y):
        print("Jogo 1 selecionado")
        print(jogos1[0].nome)
        pygame.event.clear()
        #Chamada de função da classe Jogo, que abre o jogo selecionado (jogo 1)
        jogos1[0].Jogar()
        pygame.event.clear()
    #checha se estamos no jogo 1, caso sim, o jogo 2 é iniciado
    elif seta1xy==(jogos1[1].x-75,jogos1[1].y):
        print("Jogo 2 selecionado")
        print(jogos1[1].nome)
        pygame.event.clear()
        #Chamada de função da classe Jogo, que abre o jogo selecionado (jogo 2)
        jogos1[1].Jogar()
        pygame.event.clear()

```

Fonte: Criado pelo Autor (2022)

### 3.7.3 Fundamentos para escolha do jogo da Memória

Segundo Hughes (2006), crianças com síndrome de *Down* possuem dificuldade em sua memória de curto-prazo em informações verbais, mas são ótimos em sua memória visão espacial, o que melhora significativamente a habilidade de aprender

usando informações visuais. Dessa forma, para Hughes (2006) o jogo da memória é uma ótima estratégia e estimulante para os seguintes aspectos:

- Utilizar da memória de curto prazo
- Ajudar no desenvolvimento linguístico
- Realizar associações entre imagem e palavra

Assim, o jogo da memória desenvolvido possui doze cartas, com quatro pares de letras do alfabeto, onde a criança precisa encontrar o par de cada letra. A cada descoberta de carta correspondente, um animal ou fruta, juntamente ao seu nome, irá aparecer para relacionar a letra do alfabeto encontrado, buscando aumentar o seu vocabulário. O jogo utiliza um tutor virtual, que ensina a jogar através da voz, textos e gestos, além de fazer elogios a cada acerto ou encorajar a criança a continuar a jogar caso ela erre.

O jogo da memória foi desenvolvido e adaptado usando como base um jogo para a *Raspberry Pi*, feito por Al Sweigart, que utiliza *Pygame* para seu desenvolvimento. Ele é estruturado possuindo diversas funções diferentes para a simulação de um jogo da memória com 12 cartas.

De início, o jogo identifica se a pessoa escolheu animal ou fruta e aplica as letras aleatoriamente em um vetor, que irão ser passados para mais tarde serem aplicadas no quadro. A função responsável por isso está na imagem a seguir

Figura 3.18 - Algoritmo para determinação de letras que serão usadas

```
#Responsável por pegar as letras e entregar sua devida posição.
def cartarandomica(tipo):
    #pega todas as letras disponíveis pra fruta ou animal
    icones = []
    if tipo == 'animal':
        for letra in letrasAn:
            icones.append(letra)
    elif tipo == 'fruta':
        for letra in letrasFr:
            icones.append(letra)

    #Aleatoriza e adiciona apenas o numero necessário de letras.
    random.shuffle(icones)
    num = int((Nhor)*(Nver)/2)
    icones = icones[:num]*2
    random.shuffle(icones)
    quadro = []
    #entrega a posição de cada letra para ser colocado no jogo.
    for x in range(Nhor):
        coluna = []
        for y in range(Nver):
            coluna.append(icones[0])
            del icones[0]
        quadro.append(coluna)
        random.shuffle(quadro)
    return quadro
```

Fonte: Criado pelo Autor (2022)

A função mais importante do jogo é a que inicializa o desenho do quadro. Além disso, ela é responsável por iniciar o tutorial, ou seja, as falas que são usadas no jogo para ensinar a criança a jogar, tanto por escrita quanto por voz. A imagem a seguir, mostra o início da função.

Figura 3.19 - Início da função de iniciar o jogo

```
#Inicia o jogo
def iniciojogo(quadro):
    tela.blit(escola,(0,0))
    #cartas iniciam cobertas
    cartarevelada = cartarevelação(False)
    #desenha as cartas(cobertas) a partir da função desenhaquadro
    desenhaquadro(quadro,cartarevelada)
```

Fonte: Criado pelo Autor (2022)

Como podemos observar, essa função chama uma outra função que transforma todas as cartas em não reveladas (*False*) e em seguida, chama a função de desenhar no quadro. Mais à frente, ela faz uso de mais duas funções importantes

sendo a de cobrir e revelar cartas. As imagens a seguir irão mostrar como as duas funções se comportam.

Figura 3.20 - Função que adiciona o status de revelação

```
#Identifica qual carta está revelada ou não
def cartarevelação(val):
    cartarevelada = []
    for i in range (Nhor):
        cartarevelada.append([val]*Nver)
    return cartarevelada
```

Fonte: Criado pelo Autor (2022)

Figura 3.21 - Função que faz associação de letra e cartas desenhadas

```
#Pega todas as cartas e suas coordenadas e desenha elas no quadro.
def desenhaquadro(quadro, carta):
    todas_as_estrelas.draw(tela)
    tela.blit(pygame.image.load_extended(f"{path}\\\\imagens\\dica.png"),(0,0))
    #Usa o numero de quantas cartas terão na horizontal e vertical e aplica suas coordenadas.
    for cartax in range(Nhor):
        for cartay in range(Nver):
            esquerda, cima = ecdacarta(cartax, cartay)
            #se a carta não foi revelada deixa coberta.
            if not carta[cartax][cartay]:
                pygame.draw.rect(tela,(255,255,255), ((esquerda+quarto),(cima+quarto), tamcarta, tamcarta))
            else:
                #pega uma coordenada, adiciona uma letra e desenha
                letra = getletra(quadro, cartax, cartay)
                desenhaicone(letra, cartax, cartay)
```

Fonte: Criado pelo Autor (2022)

Ou seja, ela aplica como as cartas irão ser demonstradas no quadro e faz toda a preparação de posição usando outras três funções. Onde, basicamente, são reservadas as coordenadas de cada letra, além de serem perfeitamente desenhadas no quadro.

As funções de revelar e cobrir carta são simples, como podem ser observadas na imagem a seguir.

Figura 3.22 - Funções de revelar e cobrir carta

```
#Revela a carta selecionada
def revelarcarta(quadro, cartarevelar):
    desenhocobrircarta(quadro,cartarevelar,0)

#Inicia a função de cobrir a carta pra cada carta.
def cobrircarta(quadro, cartacobrir,carta):
    for coberto in range(0,tamcarta):
        desenhocobrircarta(quadro, cartacobrir,coberto)
```

Fonte: Criado pelo Autor (2022)

Elas chamam uma mesma função que irá receber a ordem de cobrir ou revelar a carta. Que irá receber qual carta é, ou seja, suas posições no vetor de armazenamento de cartas.

Figura 3.23 - Função que faz a animação de revelar e cobrir a carta

```
#Função utilizada para cobrir as cartas no quadro.
def desenhocobrircarta(quadro, cartas, cobrir):
    for carta in cartas:
        #Mostra a carta a partir de sua posição.
        esquerda, cima = ecdacarta(carta[0],carta[1])
        pygame.draw.rect(tela, (0,0,0), ((esquerda+quarto),(cima+quarto), tamcarta, tamcarta))
        letra= getletra(quadro, carta[0], carta[1])
        desenhaicone(letra, carta[0], carta[1])
        #cobra a carta usando o quadro por segundos estabelecido.
        if cobrir > 0:
            pygame.draw.rect(tela,(255,255,255), ((esquerda+quarto),(cima+quarto),cobrir,tamcarta))
            pygame.display.update()
            clock.tick(60)
```

Fonte: Criado pelo Autor (2022)

Conforme descrito no desenvolvimento da plataforma inicial, usamos funções para encontrar as imagens e sons para serem aplicados no jogo, como exemplo:

Figura 3.24 - Chamada de imagens

```
#animais
aguia = pygame.image.load_extended(f"{path}\\\imagens\\animal\\aguia.png")
borboleta = pygame.image.load_extended(f"{path}\\\imagens\\animal\\borboleta.png")
coruja = pygame.image.load_extended(f"{path}\\\imagens\\animal\\coruja.png")
```

Fonte: Criado pelo Autor (2022)

Também foi necessário usar algumas variáveis para definir os desenhos das cartas, tamanho delas etc.

Figura 3.25 - Variáveis importantes

```
#variaveis do jogo
Nhor = 4 # numero de cartas horizontal
Nver = 3 # numero de cartas vertical
tamcarta = 75 #tamanho da carta
quarto = int(tamcarta*0.25) #variaveis de ajustes
meio = int(tamcarta*0.5)#variaveis de ajustes
gap = 50 #gap entre cartas
margemx = int((largura - (Nhor*(tamcarta+gap)))/6)
margemy = int((altura - (Nver*(tamcarta+gap)))/3)
```

Fonte: Criado pelo Autor (2022)

E assim foi possível fazer a inicialização do jogo utilizando as funções apresentadas acima. Onde a função de inicioJogo receberá todas as cartas como falsas, ou seja, que não foram descobertas ainda.

Figura 3.26 - Chamada de funções para inicialização do jogo

```
#chamadas de funções para o inicio do jogo
principalquadro=cartarandomica(tipo)
cartarevelação1 = cartarevelação(False)
selecionado = None
iniciojogo(principalquadro)
```

Fonte: Criado pelo Autor (2022)

Em seguida, entramos em um *loop* que só irá ser fechado quando acabar o jogo ou a tela ser fechada. O jogo utiliza as setas do teclado para movimentação, aonde cada movimento vai automaticamente para a carta que foi direcionado o comando. Como exemplo:

Figura 3.27 - Exemplo de condição de direção

```
#Vai para carta a direita
if pygame.key.get_pressed()[K_RIGHT]==True or joystick.get_button(14):
    pygame.mixer.Sound.play(somEscolha)
    pygame.event.clear()
    if xc<=Nhor*margemx+tamcarta:
        xc+=125
```

Fonte: Criado pelo Autor (2022)

Para o algoritmo de escolha de cartas, é necessário saber a letra da carta selecionada e qual a sua posição, além disso, precisamos saber se já tinha uma carta previamente selecionada ou não, para fazer a comparação. A imagem a seguir descreve esse processo.

Figura 3.28 - Algoritmo de seleção de carta

```
#Passa as coordenadas da carta pega para 2 variáveis
cartax, cartay = pegarcarta(xc, yc)
#Checa se não existe valor e se a carta já foi revelada ou não
if cartax != None and cartay != None :
    if not cartarevelação1[cartax][cartay] and espaço:
        revelarcarta(principalquadro,[cartax,cartay])
        cartarevelação1[cartax][cartay]=True
    #Se é a primeira carta a ser selecionada, vai para outra variável
    if selecionado == None:
        selecionado =(cartax, cartay)
        print(selecionado)
    else:
        #Checa se as letras são iguais ou diferentes
        letra1 = getletra(principalquadro,selecionado[0],selecionado[1])
        letra2 = getletra(principalquadro, cartax, cartay)
        if letra1 != letra2:
            fraseserro()
            pygame.time.wait(1000)
            cobrircarta(principalquadro,[selecionado[0],selecionado[1]],[cartax,cartay],cartarevelação1)
            cartarevelação1[selecionado[0]][selecionado[1]]= False
            cartarevelação1[cartax][cartay] = False
            pygame.event.clear()
        #Se são iguais irá mostrar o animal ou fruta referente a letra
        elif letra1==letra2:
            conto+=1
            pygame.mixer.Sound.play(somStar)
            if tipo == 'animal':
                if letra1 == A:
                    tela.blit(agavia,(665,150))
                    pygame.display.update()
                    fraseselogio()
```

Fonte: Criado pelo Autor (2022)

Como podemos observar, caso as duas cartas tenham a mesma letra, irá aparecer a imagem do animal ou fruta referente a letra acertada. O jogo possui atualmente nove letras para animais e oito letras para frutas. A cada acerto, a criança recebe uma estrela.

O código sempre faz a checagem de caso tenha chegado no final ou não. Então, se a criança conseguir todas as estrelas o jogo termina. A checagem é feita utilizando a função a seguir.

Figura 3.29 - Função de vitória

```
#Checa se todas as cartas foram descobertas
def GANHOU(cartarevelada):
    for i in cartarevelada:
        if False in i:
            return False
    return True
```

Fonte: Criado pelo Autor (2022)

E assim exibe o final do jogo, com a criança recebendo o troféu, e o jogo é finalizado. Para jogar novamente, a criança irá precisar abrir o jogo novamente no ambiente, pois o jogo fecha automaticamente, dando a sensação de tarefa concluída.

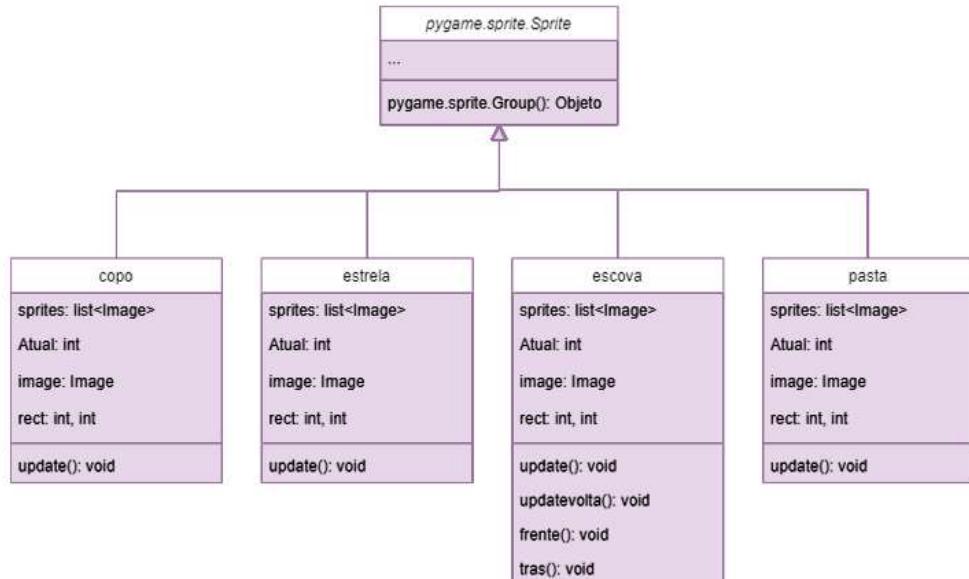
### 3.7.4 Fundamentos para escolha do jogo de Escovar os Dentes

O ato de escovar os dentes pode parecer simples, mas para crianças com atrasos cognitivos, alguns passos podem acabar passando despercebidos ou sendo difíceis de realizar. Os estudos realizados por Oredugba (2007), chegaram à conclusão de que, crianças com Síndrome de *Down*, possuem grandes dificuldades para manter sua rotina de higiene oral, além de boa parte precisar da ajuda de adultos ou cuidadores para realizar a tarefa. Por conta disso, o jogo foi pensado para ajudar crianças com Síndrome de *Down* a visualizar e reproduzir o passo a passo da tarefa de escovar os dentes, para auxiliar os pais na hora de ensiná-las e promover a inclusão social delas.

Dessa forma, o jogo foi estruturado de uma maneira visual, com atos sendo realizados automaticamente a cada avanço que a criança fizer, com um mini game, que mostra a escovação removendo a sujeira nos dentes.

O jogo de escovar os dentes possui 3 etapas, sendo apenas uma jogável, de fato. A forma de construção desse jogo, utiliza mais classes com o uso das *Sprites*<sup>12</sup>. *Sprites* são muito utilizadas para mudanças de uma mesma imagem.

Figura 3.30 - Diagrama de Classes



Fonte: Criado pelo Autor (2022)

As instâncias para as classes são definidas como descrito abaixo

Figura 3.31 - Instâncias de classes e sprites

```

#instancias
todas_as_escovas = pygame.sprite.Group()
escovas = escova()
todas_as_escovas.add(escovas)
todas_as_pastas = pygame.sprite.Group()
pastas = pasta()
todas_as_pastas.add(pastas)
todas_as_estrelas = pygame.sprite.Group()
estrelas = estrela()
todas_as_estrelas.add(estrelas)
todas_os_copos = pygame.sprite.Group()
copos = copo()
todas_os_copos.add(copos)
personagem = personagem()
personagem.criar_personagem()
  
```

Fonte: Criado pelo Autor (2022)

---

<sup>12</sup> **Sprites** - Em computação gráfica, um sprite é um objeto gráfico bi ou tridimensional que se move numa tela sem deixar traços de sua passagem.

A primeira e a terceira parte do jogo são feitas pelo mesmo *loop*. Onde as mudanças de imagem e avanços são dadas conforme o estado da escova, da pasta e do copo que estão sendo utilizados pela criança. A sequência é feita passo a passo e sempre continuada após apertar o “espaço” no teclado. Como podemos acompanhar no código a seguir.

Figura 3.32 - Exemplo de condição de avanço do tutorial

```
#Checa o evento de teclado ou controle
if pygame.event.peek(KEYDOWN)==True or pygame.event.peek(JOYBUTTONDOWN)==True:
    if (pygame.key.get_pressed()[K_SPACE]==True) or joystick.get_button(0):
        if pygame.mixer.get_busy()== False:
            pygame.mixer.Sound.play(somBota)
            #tocamos o fala da terceira fase a cada avanço
            if contar >6 and contar <10:
                somtutorial1(contar-6)
            contar+=1
            #tocamos o fala da primeira fase a cada avanço
            if contar <7:
                somtutorial(contar)
            #Atualizamos a escova e a pasta conforme o avanço
            if contar > 2:
                if escovas.atual == 0 and pastas.atual == 1:
                    #telaboca = True
                    escovas.update()
                elif escovas.atual == 0:
                    pastas.update()
```

Fonte: Criado pelo Autor (2022)

Quando terminamos a primeira parte do jogo, é iniciado o mini game de escovar os dentes. Aqui, a criança irá levar a escova até a sujeira a ser limpa, que será escolhida aleatoriamente, a partir da sua identificação. As sujeiras e sua identificação são postas em um vetor da seguinte forma:

Figura 3.33 - Vetor com as sujeiras e seu posicionamento

```
tudo = []
tudo.append((pygame.image.load_extended(f"{path}\\"imagens\\sujeiraesc.png"), "esquerda cima"))
tudo.append((pygame.image.load_extended(f"{path}\\"imagens\\sujeiraesc.png"), "esquerda baixo"))
tudo.append((pygame.image.load_extended(f"{path}\\"imagens\\sujeira.png"), "meio cima"))
tudo.append((pygame.image.load_extended(f"{path}\\"imagens\\sujeira.png"), "meio baixo"))
tudo.append((pygame.image.load_extended(f"{path}\\"imagens\\sujeiradir.png"), "direita cima"))
tudo.append((pygame.image.load_extended(f"{path}\\"imagens\\sujeiradir.png"), "direita baixo"))
```

Fonte: Criado pelo Autor (2022)

As posições são definidas depois pelas condicionais usando valores já pré-definidos, pois haverá três sujeiras nos dentes de cima e três nos dentes de baixo.

Figura 3.34 - Gerando os valores de posicionamento

```
sujeiras = []
for x in range(3):
    for y in range(2):
        if ((x == 0 or x==2) and (y == 1)):
            sujeiras.append(((x*(100+distancia)+margem),(y*(52+distancia)+margem+50)))
        elif x==1:
            sujeiras.append(((x*(100+distancia)+margem),(y*(75+distancia)+margem+75)))
        else:
            sujeiras.append(((x*(100+distancia)+margem),(y*(50+distancia)+margem+50)))
```

Fonte: Criado pelo Autor (2022)

Por sua vez, os desenhos são adicionados a tela, juntando os vetores “tudo” e “sujeiras”. Como mostra a condicional a seguir

Figura 3.35 - Juntando o vetor de sujeira com o vetor de valores

```
for sujeira in range(len(sujeiras)):
    tela.blit(todo[sujeira][0],sujeiras[sujeira])
```

Fonte: Criado pelo Autor (2022)

Para levar a escova até as sujeiras, é usado as “setas” e o “espaço” do teclado, seguindo o padrão de seleção de eventos usada nos jogos anteriores. Caso seja escolhido a sujeira errada, uma mensagem com voz e escrita dirá que a escolha está errada, mas incentiva a continuar. Se a sujeira selecionada for a correta, iremos para um outro *loop*, onde iremos limpar a sujeira.

Para limpar a sujeira, usamos as “setas” para simular a limpeza. O código que executa tal tarefa é:

Figura 3.36 - Condições de movimento da escova

```

if pygame.event.peek(KEYDOWN)==True or pygame.event.peek(JOYBUTTONDOWN)==True:
    if pygame.key.get_pressed()[K_RIGHT]==True or joystick.get_button(12):
        if escovas.rect[0]<sujeiras[vari][0]+100:
            escovas.frente()
            stack+=1

    if pygame.key.get_pressed()[K_LEFT]==True or joystick.get_button(13):
        if escovas.rect[0]>sujeiras[vari][0]:
            escovas.tras()
            stack+=1
    pygame.event.clear()

```

Fonte: Criado pelo Autor (2022)

A limpeza ocorre após fazer 10 movimentos (não importa a ordem dos movimentos). Assim que terminada, a criança recebe uma frase de elogio e o código irá substituir a identificação da sujeira selecionada e irá checar se todas as sujeiras já foram removidas, caso contrário voltamos a tela de limpar, saindo do *loop* de mini game.

Figura 3.37 - Condição para finalizar a limpeza de uma sujeira

```

#Se chegar a 10, a sujeira é substituída por um identificador sem valor
#Dessa forma, podemos randomizar a escolha sem problemas fazendo o devido tratamento
if stack>=10:
    tudo[vari] = (pygame.image.load_extended(f"{path}\\\imagens\\\nada.png"),None)
    pygame.mixer.Sound.play(somStar)
    count+=1
    tela.blit(estrela1,(837,260))
    tela.blit(parabens,(795,190))
    fraseselogio(count)
    estrelas.update()
    #Se ainda há sujeira, escolha uma nova sujeira a ser limpada
    if count<6:
        ran = random.choice(tudo)
        escovas.updatevolta()
    escovas.rect[0]= sujeiras[vari][0]+50
    escovas.rect[1]= sujeiras[vari][1]

```

Fonte: Criado pelo Autor (2022)

Indo para terceira parte do jogo, a criança irá enxaguar a boca e assim finalizar a tarefa. O funcionamento do código segue a mesma linha da primeira parte, com a diferença que, quando chegar ao final da tarefa, a criança irá receber o troféu e ser parabenizada por isso.

Figura 3.38 - Final do jogo exibindo o troféu

```
def final():
    parabens=fonte1.render('Parabéns, Aqui está sua recompensa!', True, (0,0,0))
    tela.blit(eu,(680,480))
    tela.blit(coisinho,(0,675))
    tela.blit(parabens,(15,695))
    tela.blit(trofeu,(680,200))
    if pygame.event.peek(KEYDOWN)==True or pygame.event.peek(JOYBUTTONDOWN)==True:
        if (pygame.key.get_pressed()[K_SPACE]==True) or joystick.get_button(0):
            return exp
```

Fonte: Criado pelo Autor (2022)

### 3.7.5 Fundamentos para escolha do jogo de arrumar o quarto.

A organização é uma tarefa importante para qualquer criança, que deve ser aprendida e aprimorada, trazendo responsabilidade, autonomia e melhora na sua experiência em habilidades sociais, através da comunicação e cooperação. (Louise Dunn, 2004). Pensando nisso, foi desenvolvido um jogo onde o objetivo é a criança organizar o seu quarto, colocando os brinquedos espalhados no armário de brinquedo.

Os estudos realizados por Laws (2002) mostram que, crianças com Síndrome de *Down*, tem boa relação com a identificação de cores. Complementando, Wilkinson (2008) percebeu que, ao utilizar as cores em símbolos como uma forma de comunicação, facilitaria no término da tarefa de maneira rápida e correta. Dessa forma, o método de organização escolhido foi relacionar os brinquedos a partir de cores primárias. Com isso, a criança irá aprender a diferenciar cores, organizar seus brinquedos e aprender o nome de cada brinquedo. Assim como os outros jogos, um tutor virtual auxilia a entender e seguir no jogo.

Para esse jogo, foi decidido a utilização do *mouse*, pois Feng (2008), em suas pesquisas, concluiu que pessoas com Síndrome de *Down* conseguem utilizar-se do *mouse* para interagir com o computador e que é uma boa forma de estimular a habilidade motora fina, que é uma das dificuldades em crianças com SD. Isso

acarreta algumas mudanças significativas para a estruturação e comportamento do código.

De início, foi criada uma classe para instanciar todos os brinquedos que serão utilizados no jogo. Cada brinquedo tem o seu nome, posição e a cor de referência.

Figura 3.39 - Classe de objeto imagem

```
#Estabelece o nome, a posição e a cor referente ao brinquedo
class imagens(object):
    def __init__(self,nome, position, imagem, cor):
        self.imagem = imagem
        self.position = position
        self.cor = cor
        self.nome = nome
        self.rect = imagem.get_rect()
        self.rect.topleft = self.position
        self.moving = False
        self.start = position
```

Fonte: Criado pelo Autor (2022)

Em seguida, o jogo foi desenvolvido dentro da classe Lógica, que possui funções bem importantes para a sequência do jogo. Como a função de criar as imagens dos brinquedos, que instancia todos os brinquedos usando a classe imagens. A imagem abaixo mostra a distribuição de valores para 3 brinquedos (no jogo são 9).

Figura 3.40 - Criar os brinquedos

```
def cria_imagem (self):
    posi = [(380,585),(440,680),(520,475),(590,650),(190,535),(620,475),(700,555),(265,620),(90,575)]
    random.shuffle (posi)
    self.imagem.append(imagens("bola",posi[0],pygame.image.load_extended(f"{path}\imagens\bola.png"), "verde"))
    self.imagem.append(imagens("dinossauro",posi[1],pygame.image.load_extended(f"{path}\imagens\dino.png"), "azul"))
    self.imagem.append(imagens("robô",posi[2],pygame.image.load_extended(f"{path}\imagens\robo.png"), "vermelho"))
```

Fonte: Criado pelo Autor (2022)

A imagem acima mostra que as posições são pré-definidas, em seguida colocadas em um vetor e é aplicada uma função para mexer aleatoriamente elas, após isso, as posições são passadas para os brinquedos.

A próxima função que merece destaque é a de desenhar os brinquedos na tela. Ela utiliza todos os brinquedos que foram instanciados e os desenham na tela a

partir da posição, utilizando uma condicional. Como o jogo possui duas telas, onde uma é a tela do quarto inteiro e a segunda é a tela de interação, que mostra mais de perto o armário e os brinquedos, a função possui uma condição que funciona de forma diferentes para desenhar os brinquedos.

Figura 3.41 - Função de desenhar brinquedos na tela

```
#Essa função desenha os brinquedos na tela.
def draw(self, screen, brinquedos, count):
    #Se estivermos na tela aproximada, usamos essa condição
    if brinquedos == True: ...
    #Caso seja a tela inicial, usamos essa condição
    else:
```

Fonte: Criado pelo Autor (2022)

Se a tela estiver aproximada, os desenhos irão utilizar seu posicionamento e tamanho original (passado na função `cria_imagem()`), caso contrário, eles serão diminuídos e terão posicionamento diferente.

Figura 3.42 - Condição de desenho para tela aproximada

```
if brinquedos == True:
    screen.fill((0,0,0))
    screen.blit(parede2,(0,0))
    screen.blit(modelo, (50,135))
    screen.blit(placa, (50,0))
    for i in range(0, len(self.imagem)):
        screen.blit(self.imagem[i].imagem, (self.imagem[i].rect.topleft[0],self.imagem[i].rect.topleft[1]))
```

Fonte: Criado pelo Autor (2022)

Como dito no início desse tópico, o jogo utiliza do `mouse` para interagir, assim foram criadas três funções que descrevem o comportamento dele para 3 situações: botão pressionado, botão solto e movimento do `mouse`.

Com o botão pressionado, a função que descreve seu comportamento é a seguinte

Figura 3.43 - Função de identificação do *mouse* pressionado

```
#Botão pressionado
def mouse_button_down(self):
    #Pega a posição do mouse
    mouse_pos = mouse.get_pos()
    #Ele irá checar se o posicionamento do mouse corresponde com de algum brinquedo
    for i in range(0,len(self.imagem)):
        if self.imagem[i].rect.collidepoint(mouse_pos):
            #Condição para caso a imagem ainda não esteja no armário correspondente
            if not self.collide(self.imagem[i].cor,self.imagem[i].rect):
                #Passa a posição atual para a variável start
                self.imagem[i].start = (self.imagem[i].rect[0],self.imagem[i].rect[1])
                #permite a imagem mexer
                self.imagem[i].moving = True
```

Fonte: Criado pelo Autor (2022)

Ou seja, sempre que o botão for pressionado, ele irá pegar sua posição em coordenadas e irá chegar se há uma colisão entre a posição do *mouse* e de algum brinquedo. Caso haja essa colisão, o brinquedo recebe o estado de “mexendo” e sua posição inicial é reservado em outra variável para, caso o brinquedo seja colocado em local errado, ele volta para sua posição inicial.

Por sua vez, quando o botão do *mouse* é solto, ele reage de maneira parecida, porém é aqui que ele irá identificar se o brinquedo foi posto no lugar certo ou não.

Figura 3.44 - Função de identificação do *mouse* solto

```
#Botão solto
def mouse_button_up(self, screen):
    #Apenas checa se o brinquedo solto está no armário ou não
    #Caso não volta pra posição inicial
    for i in range(0,len(self.imagem)):
        if not self.collide(self.imagem[i].cor,self.imagem[i].rect):
            (self.imagem[i].rect[0],self.imagem[i].rect[1]) = self.imagem[i].start
            self.imagem[i].moving = False
```

Fonte: Criado pelo Autor (2022)

E assim chegamos na função onde descreve o funcionamento do *mouse* após seu botão estar pressionado e carregando um brinquedo junto com ele. A função move o brinquedo que está com o estado “mexendo” conforme o quanto o *mouse* se desloca da posição inicial, como visto na imagem abaixo.

Figura 3.45 - Função de identificação do mouse sendo arrastado

```
#Movimento do mouse
def mouse_drag(self, screen):
    #Pega as coordenadas do mouse
    mouse_rel = mouse.get_rel()

    for i in range(0,len(self.imagem)):
        #Checa qual imagem foi pega e move ela conforme o mouse mexer
        if self.imagem[i].moving == True:
            self.imagem[i].rect.move_ip(mouse_rel)
            #Se colidir com o armário de cor correta ele para de mover e solta automaticamente
            if self.collide(self.imagem[i].cor,self.imagem[i].rect) and self.imagem[i].cor == "verde":
                self.count.append([self.imagem[i].nome, self.imagem[i].cor,self.flagV])
                self.imagem[i].moving = False
                self.imagem[i].rect.topleft = 70+(30+self.verde[0])*self.flagV,15+self.verde[1]
                self.flagV = self.flagV+1
                #Se tiver 3 brinquedos no armário correto, ganha elogio e uma estrela
                if self.flagV ==3:
                    pygame.mixer.Sound.play(somStar)
                    if len(self.count)<=8:
                        self.Elogios(screen)

            else:
                pygame.mixer.Sound.play(somConfirmar)
```

Fonte: Criado pelo Autor (2022)

Dessa forma, a função continua fazendo a checagem de colisão. Caso o brinquedo colidir com o armário de mesma cor que ele, o brinquedo vai para a posição pré-determinada como primeiro e assim por diante até ter três. Concluído os três brinquedos de mesma cor no armário, o código exibe uma estrela nova e uma mensagem elogiando.

Quando todos os brinquedos são colocados corretamente, é acionada a função que exibe o troféu e realiza a volta para a tela inicial, onde é exibido os brinquedos de forma organizada e o troféu em cima da escrivaninha.

Figura 3.46 - Função de vitória que exibe troféu ao final

```
#Todos os brinquedos no Lugar correto
def ganhou(self, count, screen):
    self.frasesPos(self.countFrasePos,screen)
    if self.countFrasePos == 1:
        screen.blit(pygame.image.load_extended(f"{path}\\\\imagens\\trofeu.png"),(572,100))
    for event in pygame.event.get():
        if event.type == pygame.MOUSEBUTTONDOWN and pygame.mixer.get_busy()== False:
            logica.countFrasePos=logica.countFrasePos+1
            pygame.mixer.Sound.play(somStar)
            logica.sonsPos(logica.countFrasePos)
            self.count.append('')
```

Fonte: Criado pelo Autor (2022)

Com isso, o jogo é finalizado assim que o usuário fechar para jogar novamente.

### 3.8 Planejamento de testes

Para atingir os objetivos especificados propostos para esse projeto, cada jogo seguiu o planejamento de teste detalhado nos próximos tópicos, seguindo os princípios que são utilizados em desenvolvimento de jogos em geral, adaptando para nossa realidade de projeto, para garantir a integridade e compatibilidade das ideias.

#### 3.8.1 Testes durante o desenvolvimento ou teste Alfa

Durante todo o desenvolvimento dos jogos, são realizados testes nas funcionalidades implementadas, a fim de buscar problemas ou erros no meio do caminho de desenvolvimento. A importância em fazer os testes, está em garantir a qualidade dos jogos para as próximas etapas de testes, pois erros que poderiam ser críticos podem ser fixados com antecedência e ainda assegurar que alguns *bugs*<sup>13</sup>sejam encontrados antes de virar algo maior e trazer problemas que desperdice tempo para arrumá-los.

#### 3.8.2 Teste de funcionalidade

Após o término do desenvolvimento do jogo, aonde ele se torna jogável, é realizado testes que percorrem por todo o jogo tentando encontrar erros ou *bugs* que interferem na experiência dos jogadores. Essa fase é responsável por determinar se o jogo está indo de acordo com a especificação, encontrar as inconsistências nas imagens, sons, que podem interferir na sua jogabilidade etc.

---

<sup>13</sup> Bug Um bug de software é um erro, falha ou falha no design, desenvolvimento ou operação de software de computador que faz com que ele produza um resultado incorreto ou inesperado ou se comporte de maneira não intencional.

### 3.8.3 Teste de jogabilidade

Com os testes anteriores finalizados, o jogo foi testado em outras plataformas, como *notebook*, computadores, e utilizando outros periféricos, para determinar se os recursos estavam de acordo com o propósito definido. O Objetivo dos testes é identificar possíveis *bugs*, demonstrar possíveis melhorias, e apontar onde temos dificuldades em executar as tarefas dentro dos jogos e se é necessário fazer alterações para facilitar o desempenho e entendimento das crianças com os periféricos que estão serão utilizados (como *mouse*, teclado, controle etc.).

## 4 Resultados e discussões

A partir do que foi apresentado nos tópicos anteriores, o Jogo é composto de três jogos independentes que foram desenvolvidos separadamente e depois adicionados no jogo principal. Para acessá-los e testá-los, foi utilizado um *hardware* bem simples, que fosse de baixo custo já que é o foco do projeto, a *Raspberry PI 3* modelo B. Que possui um processador quad-core de 1.2GHz, 1 GB(*GigaBytes*) de memória RAM, 4 portas USB 2.0 e HDMI, além de um microSD de 32GB para armazenamento. Os periféricos para navegação e interação usados foram um teclado e um *mouse* de computador simples, um controle de videogame (*DualShock® 4*) e para reprodução da imagem um monitor de 24° Polegadas.

Nesse capítulo, irá ser demonstrado toda a estruturação da jogabilidade, com passo a passo de cada etapa e as principais telas dos jogos. Para organizar os resultados obtidos, os tópicos serão divididos baseando-se nas fases de testes dos jogos.

### 4.1 Resultados após fases teste Alfa e teste de funcionalidade

Como parte dos resultados obtidos, serão demonstrados como os jogos foram estruturados durante o seu desenvolvimento, com comentários sobre cada etapa.

Também os conceitos implementados e o porquê de algumas tomadas de decisões para tornar os jogos mais amigáveis para o uso e prontos para eventuais testes de pós-produção.

Para iniciar os testes, é conectado na *Raspberry PI* o teclado e o *mouse* de computador via USB e um cabo HDMI para o monitor e em seguida a *Raspberry PI* é ligada na tomada com sua fonte de 5V. O jogo é aberto automaticamente assim que ligado e dessa forma pode-se navegar no menu inicial e escolher os jogos disponíveis.

#### 4.1.1 Plataforma de jogo inicial

A plataforma de jogo inicial foi desenvolvida para ser simples, intuitiva e amigável para a criança ser guiada, sem dificuldades, a escolher o jogo de sua preferência. A ideia do projeto é que fosse possível a inclusão de novos jogos na plataforma conforme eles forem desenvolvidos.

O protótipo foi desenvolvido com a tela de início, tendo um menu com 2 escolhas:

- **Ambientes** – A escolha leva o usuário a próxima página, onde mostra os ambientes disponíveis.
- **Sair** – O usuário irá fechar o jogo.

Na tela de “Ambientes” temos 3 opções a serem escolhidas, sendo elas:

- **Ambiente Casa** – O ambiente Casa mostra os jogos com a temática “casa”, onde a ideia é de serem jogos que auxiliem os pais a ensinar as crianças com SD a fazer tarefas do cotidiano, aprimorando a sua inclusão social.
- **Ambiente Escola** – O Ambiente Escola possui jogos com a temática “escola”, ou seja, voltadas a ensinar as crianças com SD coisas

relacionadas a linguagem, trabalhar sua memória cognitiva, interação, concentração etc.

- **Voltar** – Essa opção fará o usuário voltar para o menu inicial do jogo.

Ao escolher uma das opções de Ambiente, a próxima tela a ser mostrada é onde a criança escolhe o jogo a ser jogado. Para o protótipo foi desenvolvido um jogo para o Ambiente Escola o **jogo da memória** e dois jogos para o Ambiente Casa: o jogo de **escovar os dentes** e **arrumar o quarto**.

Figura 4.1 - Tela inicial e dos ambientes.



Fonte: Plataforma dos jogos, Autor (2022)

Os testes para a plataforma de jogos iniciais foram bem simples. Como a ideia é simular a área de escolha de jogos, cada jogo fica em uma determinada posição. Os três jogos possíveis de serem abertos funcionaram da maneira esperada. Toda vez que um jogo é fechado, voltamos automaticamente para o menu

de escolha de jogo e as opções de “voltar” transforma o processo de navegar entre telas mais dinâmico.

Com os testes na plataforma inicial de jogo concluídas, irá ser apresentado os testes dos jogos disponíveis.

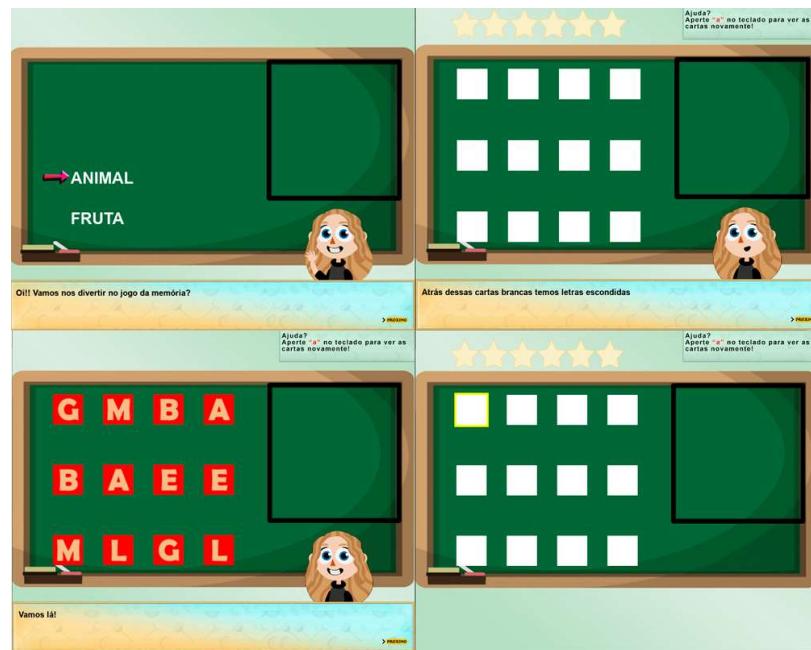
#### 4.1.2 Jogo da memória

O jogo da memória foi o primeiro a ser desenvolvido. Decidiu-se relacioná-lo ao ambiente Escola, pois leva-se a ideia de que na escola é onde a criança desenvolve seu raciocínio lógico, aprende novas palavras e aprimora a memória. Durante a fase de desenvolvimento e teste Alfa, notou-se que era importante levar em consideração o tempo individual de cada criança com Síndrome de *Down*. Por conta disso, chegou-se à conclusão de que a melhor forma para mostrar o conteúdo das cartas, no início do jogo para a memorização, seria deixar a critério da criança ou instrutor, não estabelecendo um limite para as cartas virarem. Foi optado também em deixar as letras e cores de fundo padronizados para não haver confusão na hora de visualizar cada carta, utilizando-se da cor vermelho para chamar a atenção que a carta foi virada.

Assim, o jogo da memória foi estruturado da seguinte forma:

- No início do jogo, a criança escolhe se quer aprender sobre frutas ou animais.
- Após a escolha, as cartas aparecem descobertas e a criança tem o tempo que achar necessário para memorizar as cartas. Quando se sentir segura para avançar, as cartas são cobertas.
- As cartas possuem letras do alfabeto, onde o objetivo é encontrar o par de cada letra entre as cartas.

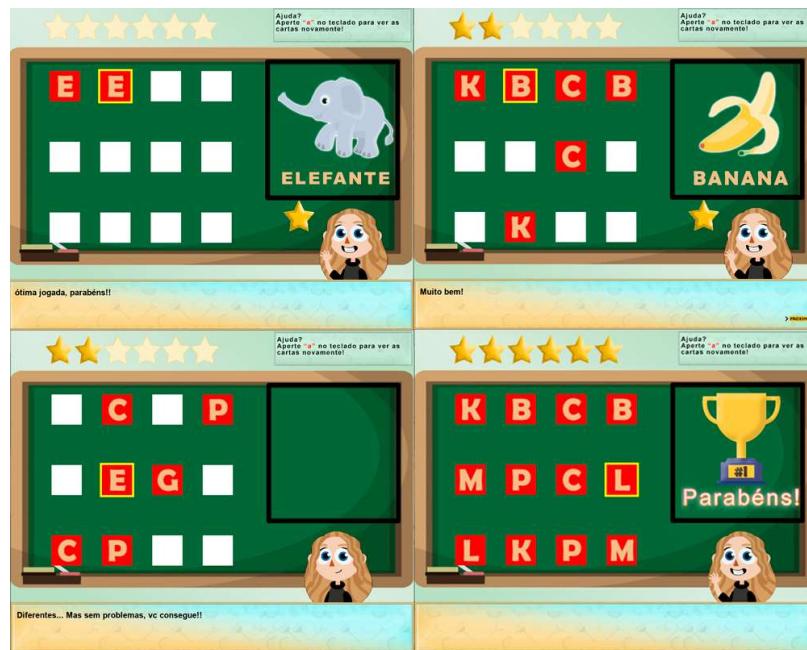
Figura 4.2 - Telas de escolha e tutorial.



Fonte: Jogo da memória, autor (2022)

- A cada par encontrado, a criança ganha uma estrela e o animal ou fruta com a letra inicial correspondendo a letra que foi acertada, aparece ao lado.
- Caso a criança erre, o tutor motiva a continuar tentando com fala e mensagem de texto, sem haver punição.
- No final, quando todas as estrelas forem recebidas, a criança recebe um troféu de recompensa.
- Para jogar novamente, é só fechar e abrir o jogo na plataforma inicial como feito inicialmente.

Figura 4.3 - Telas de acertos, erros e final.



Fonte: Jogo da memória, autor (2022)

Para interação com o jogo da memória, é utilizado apenas o teclado de computador. O usuário utiliza as “setas” do teclado para navegar pelas cartas e usa a tecla “espaço” para avanças falas ou selecionar as cartas. Os testes realizados saíram como esperado, não havendo nenhum *bug* ou inconsistência relacionado ao som, imagem e funcionalidade dos comandos. Como o jogo da memória possui suporte para o uso do controle de videogame, ele foi conectado usando um cabo USB na *Raspberry PI*. Para navegar utilizando o controle de *DualShock® 4*, é usado as “setas” que estão à esquerda do controle e o botão “X” para selecionar as cartas. Com o controle os testes foram positivos, não havendo nenhum erro caso algum botão seja apertado errado ou fora de hora.

#### 4.1.3 Jogo de escovar os dentes

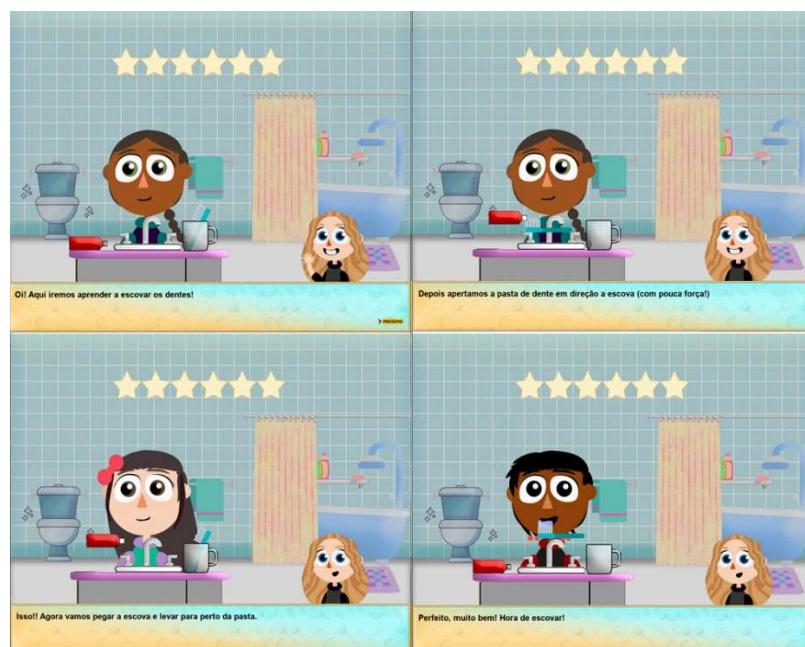
Já o jogo de escovar os dentes, foi o primeiro destinado ao ambiente Casa, pois remete ao ensinamento familiar de higiene e cuidados, que são aprendidos em casa e os desenvolvem para a inclusão social. Durante os testes Alfa, pode-se observar que a tarefa poderia ser muito longa, entediante e difícil para a criança com

Síndrome de *Down* aprender. Para evitar isso e tornar o jogo mais interessante, ele foi dividido em três etapas, onde apenas em uma delas a criança, de fato, interage como um jogo. Isso traz uma dinâmica e mudança de ritmo que pode trazer uma satisfação e curiosidade ao concluir cada etapa.

A estrutura para esse jogo foi pensada e desenvolvida desta forma:

- No início do jogo, o tutor irá ensinar o passo a passo para se escovar os dentes.
- Seguindo dessa forma: Pegar a escova -> Pegar a pasta de dente -> Abrir a pasta de dente-> posicionar a escovar embaixo da pasta -> apertar a pasta- > levar a escova até a boca.
- Com a escova na boca, o mini game começa, onde o jogo informa qual sujeira a criança precisa limpar, seguindo uma sequência aleatória.

Figura 4.4 - Telas de início e tutorial inicial.



Fonte: Jogo de escovar os dentes, autor (2022)

- Para limpar a sujeira, a criança leva a escova até a sujeira, seleciona o local e faz movimentos para “direita e “esquerda”, simulando uma escovação. O progresso é mostrado a partir de uma barrinha ao lado da boca.

- Com a barrinha cheia, a criança recebe uma estrela e o jogo seleciona outra sujeira a ser removida.
- Após a remoção de todas as sujeiras, o mini game acaba e ela volta para o passo a passo.
- Jogar água na escova -> encher um pouco uma xícara com água -> fazer o bochecho -> cuspir a água na pia.
- Concluído todas as etapas, a criança recebe um troféu como recompensa final.

Figura 4.5 - Mini game e tela pós finalizar escovação.



Fonte: Jogo de escovar os dentes, autor (2022)

Como notado nas imagens, foi tomada a decisão de que, como o jogo de escovar os dentes possui dois personagens na tela, o personagem que irá escovar os dentes varia entre seis estilos diferentes. Essa escolha partiu da ideia de inclusão social para as crianças com Síndrome de *Down* e representação. Eles mudam a cada vez que o jogo for aberto.

Assim como no jogo da memória, os testes realizados foram positivos, não havendo apenas inconsistência na movimentação da escova caso as “setas” sejam apertadas rapidamente uma após a outra, mas não comprometendo a jogabilidade. Esse jogo

possui suporte também para o controle de videogame, onde as “setas” do controle movimentam as escovas e o “X” é usado para selecionar as sujeiras e avançar falas.

#### 4.1.4 Jogo de arrumar o quarto

O terceiro e último jogo desenvolvido foi o de arrumar o quarto, que está ligado ao ambiente Casa. Em seu desenvolvimento, chegou-se à conclusão que, por se tratar de um jogo que simula uma tarefa que exige da habilidade motora, foi pensado em utilizar o *mouse* do computador para jogá-lo, pois é uma maneira interessante de treinar a coordenação e a habilidade motora da criança, sem trazer um esforço muito grande e complicado. A escolha dos brinquedos passa pela importância da inclusão social, trazendo diversos tipos de brinquedos diferentes. O jogo está estruturado da seguinte forma:

Figura 4.6 - Telas de início mostrando o quarto e aproximando do armário.



Fonte: Jogo de arrumar o quarto, autor (2022)

- De início, o instrutor virtual observa o quarto bagunçado e convida a criança para organizar.

- Para isso, a criança deve levar o *mouse* até o armário e clicar com botão esquerdo para começar.
- Após clicar, aproximamos do armário e dos brinquedos espalhados.
- O tutor explica as cores que são relacionadas com os brinquedos.
- No final da explicação, a criança poderá escolher qualquer brinquedo a sua escolha, clicando com o botão esquerdo do *mouse* no brinquedo.
- Após o clique o brinquedo e seu nome aparecem no canto superior direito para visualização e aprendizado.
- A criança então, arrastará o brinquedo até o local com a cor correspondente a do brinquedo escolhido.
- Quando a criança organizar 3 brinquedos da mesma cor, ela ganhará uma estrela e um troféu. Assim voltaremos para a tela de quarto.
- O quarto estará organizado e o troféu exposto na bancada da criança.

Figura 4.7 - Telas de acerto de cada etapa mais o final onde tudo está organizado.



Fonte: Jogo de arrumar o quarto, autor (2022)

Para o jogo de arrumar o quarto, foi utilizado apenas o *mouse* para interação com o jogo. Os testes mostraram que o jogo possui inconsistência apenas caso dois brinquedos estejam próximos demais um do outro no momento que forem

selecionados, pois o jogo irá interpretar que foram os dois brinquedos e não escolhendo apenas um. Porém, atualmente, não há risco de haver brinquedos próximos o suficiente para desencadear essa inconsistência. Esse jogo não possui suporte para controle, sendo usado apenas o *mouse* de computador.

Dos jogos desenvolvidos, o de arrumar o quarto exige mais da sua cognição motora. Pois utilizar o *mouse* é uma tarefa que envolve movimento e não apenas o clicar de uma tecla ou botão. Dessa forma, algumas dificuldades podem aparecer, mas torna o desenvolvimento mais interessante de ser observado, caso seja realizado da maneira correta pela criança.

#### 4.2 Mudanças realizadas após os testes de funcionalidade

Após o desenvolvimento dos jogos e a realização dos testes alfas e os de jogabilidade, percebeu-se que apesar da importância dos tutoriais, que são necessários e indispensáveis, podem acabar sendo longos, o que poderia acarretar uma perda de foco momentânea. Dessa forma, para aumentar a isonomia da criança com os jogos, cada jogo recebeu a possibilidade de repetir os tutoriais, para caso a criança tenha dificuldade na compreensão da tarefa, ela seja capaz de rever as instruções quantas vezes forem necessárias, mesmo durante a tarefa.

Além disso, para acrescentar variações e ajudar em seu desenvolvimento cognitivo, com a utilização do *Pygame*, que possui integração com periféricos para jogos, os jogos de memória e de escovar os dentes, possuem suporte para utilização de controles feitos para jogos. O interessante dessa adição está em desenvolver outras funções cognitivas e uma forma diferente para a criança se sentir motivada a repetir certas tarefas, mas usando formas diferentes para jogar.

#### 4.3 Expansão de utilização para outros sistemas operacionais

Com o sucesso dos resultados obtidos dos jogos na *Raspberry PI*, foi discutido a possibilidade de os jogos serem disponibilizados em outros sistemas operacionais, de forma que não seja necessário instalação de IDE, compiladores e

bibliotecas para ter acesso ao jogo, pois traria uma expansão em sua utilização, caso o público-alvo já tenha uma plataforma computacional, assim ajudando com a premissa de motivação de algo de baixo custo.

Assim, foi utilizado uma ferramenta chamada *PyInstaller*<sup>14</sup> onde é possível transformar todos os arquivos do jogo, mais o código desenvolvido e as bibliotecas utilizadas em seu desenvolvimento, em um único executável(.exe). Esse executável, quando acionado, irá abrir uma pasta temporária onde todos os dados necessários estarão e excluirá a pasta assim que o jogo for fechado. Cada jogo então se torna um executável, tendo assim quatro arquivos, que no total possui 58 MB(MegaBytes) de tamanho.

Os testes para o executável, foram realizados usando um computador com sistema operacional Windows, com processador Intel i7-4790, com 3.6 GHz (GigaHertz) e memória RAM DDR3 com 16 GB(GigaBytes). Os resultados foram positivos e mostram que os jogos funcionam de maneira correta, porém com um tempo de carregamento para abrir os jogos com demora de aproximadamente 2~5 segundos para serem executados e, cada jogo, abre como um novo processo de execução (como na imagem abaixo), diferentemente dos jogos rodados na *Raspberry PI* que são executados tudo em um mesmo processo.

Figura 4.8 - Jogos inicializados a partir de executável



Fonte: Autor (2022)

---

<sup>14</sup> ferramenta para gerar arquivos executáveis em Python para Windows e Linux. Com ela você gera arquivos binários e que não dependem de outras bibliotecas para serem executados.

## 5 Considerações finais

Esse trabalho, surge da necessidade de desenvolver jogos de baixo custo para crianças com dificuldades cognitivas e promover sua autonomia para uma melhor inclusão social mais saudável por meio da tecnologia. Muitas ONGs que tem o intuito de auxiliar a inclusão social dessas crianças, possuem recursos limitados e o uso da tecnologia acaba sendo quase nulo, onde a alfabetização e ensinamentos, ainda são aplicados de forma manual para essas crianças. Isso acontece por falta de investimento ou até mesmo de conhecimento para utilizar tecnologias disponíveis.

Primeiramente, foi realizado uma pesquisa bibliográfica e teórica para levantar as dificuldades cognitivas em crianças com Síndrome de *Down* e como os jogos eletrônicos sendo utilizados como uma tecnologia assistiva pode auxiliar o desenvolvimento dessas crianças. Dessa forma, criar jogos para crianças com dificuldades cognitivas está ligado, não só a apresentação da tecnologia, mas também a inclusão e promoção dessas crianças em nossa sociedade.

Além disso, com o foco de produzir projetos que sejam relevantes a sociedade e que visa o baixo custo, vimos que a *Raspberry PI* é uma importante aliada para atingir os objetivos que tenham esses ideais. Por ser portátil, leve e de fácil manutenção para quem queira desenvolver aplicações mais complexas, porém que não demandam uso processional alto.

Dessa forma, os jogos desenvolvidos para este trabalho, alcançaram o objetivo proposto que era de serem simples, dinâmicos, de fácil compreensão, não possuir uma jogabilidade maçante ou que tire o foco das crianças que forem jogá-los. Todas as funcionalidades estão de acordo com o que foi estudado, com os jogos tendo uma interação positiva com a *Raspberry PI*, que por sua vez, interage perfeitamente com os *hardwares* usados. Assim, é possível, de fato, desenvolver jogos em computadores de baixo custo, cumprindo o que foi proposto.

Relevante destacar que, com a comprovação realizada da possibilidade de desenvolver jogos eletrônicos que sejam inclusivos e desenvolvam as crianças com dificuldades cognitivas, abre portas para a realização de testes de pós-produção

com especialistas da área, acompanhados do nosso público-alvo, a fim de enriquecer mais o seu uso e efetividade.

### 5.1 Dificuldades

Durante o desenvolvimento do projeto, foi encontrado algumas dificuldades que envolvem diversas áreas. Como jogos eletrônicos são frutos de muitos estudos multidisciplinares, trabalhos e aperfeiçoamentos, envolvem profissões do mais amplo espectro. Além disso, a área da pedagogia possui várias formas de abordagens quando se trata ao ensino de crianças, porém, para crianças com dificuldades cognitivas, o material é um pouco mais escasso e de difícil acesso. Combinar essas duas áreas é um desafio de profundidade enorme. As principais dificuldades encontradas foram:

- Encontrar métodos pedagógicos comprovadamente eficientes para adaptá-los aos jogos eletrônicos.
- Realizar tarefas que envolvem design gráfico, como design dos objetos, personagens, tela de fundo, e sonorização para a adaptação de sons, enquanto é realizado o desenvolvimento dos jogos, acabou trazendo algumas limitações de aprimoramento na interface gráfica com o usuário.
- Proporcionar um equilíbrio entre a aplicação da parte teórica, no desenvolvimento do jogo, para não se tornar cansativo e maçante.

### 5.2 Trabalhos futuros

Projetos de baixo custo no seu desenvolvimento e na sua produção, para indivíduos com deficiências, são um campo com diversas oportunidades. Diversas ONGs e até pais de crianças com dificuldade cognitiva, que não dispõe o capital necessário para comprar computadores potentes, pagar internet e afins, trazendo dificuldades na utilização dos benefícios da tecnologia. Por conta disso, é notório a importância de um estudo e desenvolvimento de projetos mais robustos e

abrangentes. Pensando nisso, algumas melhorias e ideias para o aprimoramento do trabalho que valem destacar:

- Desenvolver um controle ergonômico e especial próprio para potencializar o uso dos jogos, mantendo o suporte para *mouse* e teclado.
- Realizar testes de usabilidade com especialistas e o público-alvo.
- Adicionar níveis de dificuldades aos jogos e disponibilizar ajuda caso necessário.
- Trazer mais acessibilidade para todos os jogos, podendo configurá-los para atender cada criança.
- Desenvolver jogos para crianças que lidam com outras dificuldades cognitivas (como no Transtorno do Espectro Autista).
- Tornar a plataforma de jogos inicial mais dinâmica, para receber mais jogos.
- Criação de uma aplicação que possibilite o armazenamento de jogos em nuvem, para empresas que tenham interesse em desenvolver jogos para crianças com dificuldades cognitivas publicá-los e, dessa forma, utilizar a plataforma de jogos iniciais para ter diversos jogos acessíveis em um único lugar.
- Possibilitar alguns jogos serem multijogadores, para a sociabilização da criança.
- Melhorar os designs para serem mais amigáveis;
- Pesquisar e desenvolver jogos com a possibilidade de uso de realidade aumentada.

## REFERÊNCIAS

ADEBISI, Rufus Olanrewaju; LIMAN, Nalado Abubakar; LONGPOE, Patricia Kwalzoom. **Using Assistive Technology in Teaching Children with Learning Disabilities in the 21st Century.** Journal of Education and Practice, v. 6, n. 24, p. 14-20, 2015.

BERSCH, Rta. **Introdução à tecnologia assistiva.** Porto Alegre: CEDI, v. 21, 2008.

BISSOTO, M. L. **Desenvolvimento cognitivo e o processo de aprendizagem do portador de Síndrome de Down: revendo concepções e perspectivas educacionais.** Ciências & Cognição, v. 4, 11.

BOWER, Anna; HAYES, Alan. **Short-term memory deficits and Down's syndrome: A comparative study.** Down syndrome research and practice, v. 2, n. 2, p. 47-50, 1994.

BEDIOU, Benoit et al. **Meta-analysis of action video game impact on perceptual, attentional, and cognitive skills.** Psychological bulletin, v. 144, n. 1, p. 77, 2018.

COSTA, Hiwet Mariam; PURSER, Harry Robert McSweeney; PASSOLUNGHI, Maria Chiara. **Improving working memory abilities in individuals with Down syndrome: a treatment case study.** Frontiers in psychology, v. 6, p. 1331, 2015.

DE GLORIA, Alessandro; BELLOTTI, Francesco; BERTA, Riccardo. **Serious Games for education and training.** International Journal of Serious Games, v. 1, n. 1, 2014.

DEUBEL, Patricia. **Game on!** The Journal, v. 33, n. 6, 2006.

BADDELEY, Alan. **Working memory: looking back and looking forward.** Nature reviews neuroscience, v. 4, n. 10, p. 829-839, 2003.

ERDEM, Raziye. **Students with special educational needs and assistive technologies: A literature review.** Turkish Online Journal of Educational Technology-TOJET, v. 16, n. 1, p. 128-146, 2017.

FARAGHER, Rhonda et al. **Children with Down Syndrome Learning Mathematics: Can They Do It? Yes, They Can!** Australian primary mathematics classroom, v. 13, n. 4, p. 10-15, 2008.

GALVÃO FILHO, Teófilo Alves. **A Tecnologia Assistiva: de que se trata.** Conexões: educação, comunicação, inclusão e interculturalidade, v. 1, p. 207-235, 2009.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa.** 4. ed. São Paulo: Atlas, 2008.

GRIFFITHS, Mark D. The educational benefits of videogames. **Education and health**, v. 20, n. 3, p. 47-51, 2002.

GROS, Begoña. Digital games in education: The design of games-based learning environments. **Journal of research on technology in education**, v. 40, n. 1, p. 23-38, 2007.

Heins, Matthew C., **Video Games In Education.** Education and Human Development Master's Theses. 625, 2007.

JUNIOR, Josimar Alves De Almeida et al.. **Meu abc Down: aplicativo para alfabetização e letramento de crianças com Síndrome de Down.** Anais IV CONEDU... Campina Grande: Realize Editora, 2017.

KARINE RAMOS, Daniela. **Jogos cognitivos eletrônicos: contribuições à aprendizagem no contexto escolar.** Ciênc. cogn., RIO de JANEIRO, V. 18, n. 1, p. 19-32, abr. 2013.

LIMA, S. C.; LEITE, C.; LEITE, R. B.; ALCHIERE, J. C.; SILVA, R. H.; ALBUQUERQUE, F. S. **SÍNDROME DE DOWN: ESTUDO EXPLORATÓRIO DA MEMÓRIA NO CONTEXTO DE ESCOLARIDADE.** Ciências & Cognição, v. 14, n. 2, 21 jul. 2009.

MCFADDEN, Amanda T. **The experiences of teachers teaching children with Down syndrome in the early years of schooling.** Tese de Doutorado. Queensland University of Technology, 2014.

MARTIN, Gary E. et al. **Language characteristics of individuals with Down syndrome.** Topics in language disorders, v. 29, n. 2, p. 112, 2009.

PRENSKY, Marc. **Digital game-based learning.** Computers in Entertainment (CIE), v. 1, n. 1, p. 21-21, 2003.

PRESTES, Alyane Caramori; DUARTE, Sanny Carla; DE LIMA, Siumara Aparecida. **A utilização de jogos eletrônicos na estimulação psicomotora e cognitiva de crianças com Síndrome de Down.** Faculdade Sant'Ana em Revista, v. 3, n. 1, p. 70-85, 2019

RODRIGUES, Patrícia Rocha; ALVES, Lynn Rosalina Gama. **TECNOLOGIA ASSISTIVA – UMA REVISÃO DO TEMA.** HOLOS, [S.I.], v. 6, p. 170-180, jan. 2014.

SAVI, Rafael; ULBRICHT, Vania Ribas. **Jogos digitais educacionais: benefícios e desafios.** RENOTE-Revista Novas Tecnologias na Educação, v. 6, n. 1, 2008.

SILVA, Maria de Fátima Minetto Caldeira; KLEINHANS, Andréia Cristina dos Santos. **Cognitive processes and brain plasticity in Down syndrome.** Revista Brasileira de Educação Especial, v. 12, n. 1, p. 123-138, 2006.

SPENCE, Ian; FENG, Jing. **Video games and spatial cognition.** Review of General Psychology, v. 14, n. 2, p. 92-104, 2010.

THEOFYLAKTOS ANASTASIADIS; GEORGIOS LAMPROPOULOS; KERSTIN SIAKAS. **Digital Game-based Learning and Serious Games in Education.** International Journal of Advances in Scientific Research and Engineering (IJASRE), v. 4, n. 12, p. 139-144, 5 Dec. 2018.

THIOLLENT, Michel. **Metodologia da pesquisa - ação.** 2. ed. São Paulo: Cortez, 1986.

TORRES-CARRIÓN, Pablo V. et al. **Improving cognitive visual-motor abilities in individuals with Down syndrome.** Sensors, v. 19, n. 18, p. 3984, 2019.

DUNN, Louise. **Validation of the CHORES: a measure of school-aged children's participation in household tasks.** Scandinavian Journal of Occupational Therapy, 11:4, 179-190, 2004.