

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO  
UNIDADE ACADÊMICA DE GARANHUNS

NAIRA LUCIA TORRES PEREIRA

**A CUSTOMIZAÇÃO DE ATIVIDADES DO GCOMPRIS  
UTILIZANDO O DESIGN PARTICIPATIVO COM CRIANÇAS:  
UM ESTUDO DE CASO**

Garanhuns  
2019

NAIRA LUCIA TORRES PEREIRA

**A CUSTOMIZAÇÃO DE ATIVIDADES DO GCOMPRIS  
UTILIZANDO O DESIGN PARTICIPATIVO COM  
CRIANÇAS: UM ESTUDO DE CASO**

Trabalho de Conclusão de Curso submetido à Universidade Federal Rural de Pernambuco - Unidade Acadêmica de Garanhuns, como requisito parcial para obtenção do grau de Bacharel em Ciência de Computação, sob a orientação do Drº Mariel José Pimentel de Andrade.

Garanhuns  
2019

Dados Internacionais de Catalogação na Publicação  
Universidade Federal Rural de Pernambuco  
Sistema Integrado de Bibliotecas  
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

---

- P436c Pereira, Naira Lucia Torres  
A Customização de Atividades do GCompris Utilizando o Design Participativo com Crianças: um estudo de caso / Naira Lucia Torres Pereira. - 2019.  
62 f. : il.
- Orientador: Mariel Jose Pimentel de Andrade.  
Inclui referências e apêndice(s).
- Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco,  
Bacharelado em Ciência da Computação, Garanhuns, 2019.
1. Design Participativo. 2. GCompris. 3. Design com Crianças. 4. Design Thinking. I. Andrade, Mariel Jose Pimentel de, orient. II. Título

CDD 004

---

NAIRA LUCIA TORRES PEREIRA

A CUSTOMIZAÇÃO DE ATIVIDADES DO GCOMPRIS UTILIZANDO DESIGN  
PARTICIPATIVO COM CRIANÇAS: UM ESTUDO DE CASO

Trabalho de Conclusão de Curso submetido  
à Universidade Federal Rural de Pernambuco  
- Unidade Acadêmica de Garanhuns, como  
requisito parcial para obtenção do grau de  
Bacharel em Ciência de Computação, sob a  
orientação do Drº Mariel José Pimentel de  
Andrade.

Garanhuns, 20 de dezembro de 2019.

---

Orientador: Drº Mariel José Pimentel de  
Andrade  
Universidade Federal Rural de Pernambuco -  
UFRPE

---

Drº Anderson Fernandes de Alencar  
Universidade Federal Rural de Pernambuco -  
UFRPE

---

Drº Igor Medeiros Vanderlei  
Universidade Federal Rural de Pernambuco -  
UFRPE

*Dedico este trabalho à minha avó Ananete,  
que orou e não mediu esforços para que eu chegasse até esta etapa.*

# Agradecimentos

Ao Senhor, por todos os bens que tem feito na minha vida, por proporcionar bênçãos sem medidas sob mim, e por Sua Graça em me guiar ao longo desta graduação.

Ao meu esposo Thiago, pelas palavras de apoio nos momentos mais difíceis, por ter me ajudado a enxergar o que era importante, pelo colo e todos os conselhos dados ao longo do curso, obrigada por se fazer presente em cada momento.

À minha família, minha avó Ananete, minha mãe e irmãos, por todo o apoio emocional e orações, pela compreensão e carinho, não teria chegado até aqui sem a ajuda de vocês.

Aos Professores por compartilhar o conhecimento e experiências enriquecendo o nosso aprendizado, pelas oportunidades proporcionadas e pelo tempo dedicado à nós. Em especial ao meu orientador Drº Mariel José Pimentel de Andrade pelo apoio, compreensão e dedicação à realização deste trabalho.

Aos professores e coordenadores do LMTS pela oportunidade e o acolhimento ao longo deste período, e aos meus colegas do LMTS pelos momentos vividos.

Às minhas amigas, Amanda, Vanessa, Raquel, Aline e Isa, presentes que BCC me deu, vocês são incríveis e vão continuar presentes em minha vida.

E, por fim, à todos os colegas de curso e amigos que direta ou indiretamente contribuíram para a conclusão desta etapa.

# Resumo

Diversos sistemas de software são desenvolvidos atualmente a partir da visão dos desenvolvedores que, nem sempre, compreendem a visão do seu público-alvo, influenciando o modo como os usuários interagem com o sistema. Com o crescimento do desenvolvimento de sistemas de software focados na educação, surge a demanda de compreender o universo dos usuários, suas visões e expectativas; em especial, as crianças. Este trabalho tem como principal objetivo avaliar o uso de Design *Thinking* com crianças, numa perspectiva do Design Participativo, para customização e desenvolvimento de uma atividade no software GCompris. Para isto, realizou-se um estudo de caso com um grupo de crianças, na perspectiva de método qualitativo de observação participante, na busca pela compreensão de quais dificuldades são encontradas neste tipo de processo. A aplicação do Design Participativo permitiu enxergar a necessidade do envolvimento de potenciais usuários no processo de criação, pois levou em consideração suas opiniões e questionamentos sobre o produto que está sendo desenvolvido. O uso das técnicas de Design *Thinking* contribuíram para a percepção destas necessidades, visto que orientam de modo eficiente esta construção.

**Palavras-chave:** Design participativo. GCompris. Design com crianças. Design Thinking

# Abstract

Several software systems currently developed for developers, who don't quite understand their target audience perspectives and reality, have a direct influence on the way users interact with these systems. As the development of educational software grows, the need to comprehend the users' universe, their perspectives and expectations, rises as well, especially for children. This study aims to evaluate the use of Design Thinking with children, from a participative design perspective to the customization and development of activity on the software GCompris. Therefore, a case study was conducted with a group of kids, on the perspective of participant observation qualitative method, aiming to comprehend what difficulties appear in this type of process. The application of Participative Design allowed us to see the need for the involvement of potential users in the creation process, as it took into consideration their opinions and questions about the product being developed. The use of Design *Thinking* techniques contributed to the perception of these needs, as they efficiently guide this construction.

**Keywords:** Participatory Design. GCompris. Design with Children. Design Thinking

# Lista de ilustrações

Figura 1 – Etapas do Design <i>Thinking</i> usadas neste trabalho . . . . .	32
Figura 2 – Versão original atividade <i>Programming Maze</i> . . . . .	33
Figura 3 – Problemas encontrados no cenário original . . . . .	34
Figura 4 – Sugestões propostas como soluções . . . . .	35
Figura 5 – Personagens sugeridos para atividade “ <i>Programming Maze</i> ” . . . . .	37
Figura 6 – Protótipo de cenário da atividade “ <i>Programming Loop</i> ” . . . . .	38
Figura 7 – Primeira versão de design da Atividade “ <i>Programming Maze</i> ” . . . . .	39
Figura 8 – Versão final do design da atividade “ <i>Programming Maze</i> ” . . . . .	40
Figura 9 – Primeira Versão da Atividade “ <i>Programming Loop</i> ” . . . . .	41
Figura 10 – Versão final do design da atividade “ <i>Programming Loop</i> ” . . . . .	42
Figura 11 – Estrutura da atividade “ <i>Programming Loop</i> ” . . . . .	43
Figura 12 – Versão original das atividades . . . . .	57

# Lista de quadros

2.1	Categorias GCompris . . . . .	19
4.1	Realização das atividades . . . . .	32
C.1	Percepção dos participantes - Atividade original . . . . .	60
C.2	Percepção dos participantes - Atividade “ <i>Programming Maze</i> ” primeira versão	61
C.3	Percepção dos participantes - Atividade “ <i>Programming Maze</i> ” versão final .	61
C.4	Percepção dos participantes - Atividade “ <i>Programming Loop</i> ” primeira versão	62
C.5	Percepção dos participantes - Atividade “ <i>Programming Loop</i> ” versão final .	62

# Lista de códigos

A.1	Código do padrão GCTest . . . . .	53
A.2	Código para String . . . . .	53
A.3	Código para uso de anchors . . . . .	54

# Lista de abreviaturas e siglas

GPL	General Public License
QML	Qt Meta Linguagem ou Qt Modeling Language
CSTA	Computer Science Teachers Association
JS	JavaScript
KDE	K Desktop Environment
LMTS	Laboratório Multidisciplinar de Tecnologias Sociais

# Sumário

1	INTRODUÇÃO . . . . .	15
1.1	Problema de Pesquisa . . . . .	16
1.2	Objetivos . . . . .	16
1.3	Organização do Trabalho . . . . .	16
2	REVISÃO DE LITERATURA . . . . .	17
2.1	Software Livre . . . . .	17
2.2	GCompris . . . . .	19
2.3	Pensamento Computacional . . . . .	21
2.4	Desenvolvimento de Artefatos Digitais para Crianças . . . . .	23
2.5	Design participativo . . . . .	25
2.6	Design <i>Thinking</i> . . . . .	26
3	PERCURSO METODOLÓGICO . . . . .	29
3.1	Tipo de Pesquisa . . . . .	29
3.2	Sujeitos da Pesquisa . . . . .	29
3.3	Procedimento de Coleta dos Dados . . . . .	30
4	RESULTADOS . . . . .	31
4.1	Customização Usando Design Participativo . . . . .	31
4.1.1	Etapa 1: Empatia . . . . .	32
4.1.2	Etapa 2: Definição . . . . .	34
4.1.3	Etapa 3: Ideias . . . . .	35
4.1.4	Etapa 4: Protótipo . . . . .	37
4.1.5	Etapa 5: Teste . . . . .	38
4.1.6	Implementação da atividade <i>Programming Loop</i> . . . . .	42
4.2	Utilização de Design Participativo com Crianças na Customização da Suíte GCompris . . . . .	44
5	CONSIDERAÇÕES FINAIS . . . . .	46
5.1	Conclusões . . . . .	46
5.2	Limitações do Trabalho . . . . .	47
5.3	Trabalhos Futuros . . . . .	47
	REFERÊNCIAS . . . . .	48

	<b>APÊNDICES</b>	<b>51</b>
	<b>APÊNDICE A – CUSTOMIZAÇÃO DE AMBIENTE DE DESENVOLVIMENTO DO GCOMPRIS . . . . .</b>	<b>52</b>
	<b>APÊNDICE B – DESCRIÇÃO DAS ATIVIDADES DE PROGRAMAÇÃO . . . . .</b>	<b>56</b>
B.1	A atividade “ <i>Programming Maze</i> ” . . . . .	56
B.2	A atividade “ <i>Programming Loop</i> ” . . . . .	56
	<b>APÊNDICE C – ENTREVISTA DE GRUPO FOCAL . . . . .</b>	<b>58</b>

# 1 Introdução

O incentivo ao uso de oftwares educacionais cresce a cada dia na intenção de proporcionar aos estudantes qualidade de ensino e facilidade no aprendizado, além de contribuir para que os mesmos construam novos conhecimentos. Há também o incentivo que esses softwares educacionais sejam livres. De acordo com a Free Software Foundation ao se produzir softwares livres há o respeito à liberdade e ao senso de comunidade dos usuários. Ou seja, estes usuários podem executar, copiar, distribuir, estudar e melhorar o software, contribuindo assim para a sua melhoria e divulgação em larga escala de soluções eficientes e de baixo custo (FSF, 2019).

O GCompris é um exemplo de software livre que possui atividades de cunho educacional com a finalidade de contribuir para o processo de aprendizagem tornando-o mais lúdico. O GCompris não está vinculado a orientações curriculares de um país específico, apesar de estar disponível em mais de 50 idiomas e ter como principal público crianças na faixa de 2 a 10 anos; fica a critério do professor utilizar as atividades de acordo com os objetivos educacionais estabelecidos pelos mesmos (GCOMPRIS, 2016). Além de possuir atividades relacionadas ao currículo escolar tais como ciências, matemática, letramento, há também atividades que propõem auxiliar o aluno na literacia digital estimulando o aprendizado sobre o uso de periféricos como o mouse e teclado.

No entanto, com a integração dos conceitos sobre pensamento computacional ao currículo do ensino básico ganhando espaço, faz-se necessário a atualização de atividades deste cunho no software. Para Wing (2006) o pensamento computacional é um método para a solução de problemas baseado nos fundamentos e técnicas da Ciência da Computação, e sua principal proposta é aplicar as habilidades utilizadas para criação de programas computacionais como metodologia para solução de problemas gerais. Através do uso do pensamento computacional no ensino fundamental é possível não só despertar o interesse pela área da computação, mas também contribuir para a relação de outras áreas de estudo com a computação. A fim de proporcionar uma interface mais atrativa e compreensível para este público, é necessário entender a perspectiva dos usuários, daí parte a necessidade de envolver crianças na criação de artefatos digitais destinados ao público infantil. É necessário fazer a criança ser ouvida para entender a sua perspectiva do que é interessante, auxiliando no desenvolvimento de produtos compreensíveis e usáveis.

Martins et al. (2017) afirma que o design centrado no usuário é um conceito que baseia-se na ideia de que todo e qualquer processo de design precisa ter o usuário como origem e como finalidade na sua concepção e planejamento. Possibilitar a expressão do usuário é um dos principais meios para se alcançar o engajamento deles e o sentimento de

pertencimento ao grupo de trabalho. Para isto é possível usar o design participativo para coletar, analisar e projetar as informações do processo de desenvolvimento de sistemas focando na necessidade do usuário final. Baseado nesta metodologia temos o uso do Design *Thinking*. Brown (2010) diz que o Design *Thinking* não é apenas uma proposta centrada no ser humano, sendo profundamente humana pela própria natureza, pois se baseia na capacidade do ser humano em ser intuitivo, reconhecer padrões, desenvolver ideias que tenham um significado emocional, ultrapassando as barreiras do funcional.

## 1.1 Problema de Pesquisa

Devido ao crescimento do uso de softwares educacionais, há a necessidade de compreender a perspectiva de potenciais usuários destes softwares, na intenção de desenvolver produtos mais atrativos. Por consequência se faz necessário questionar-se quais as potencialidades e dificuldades do uso do Design Thinking com crianças, numa perspectiva do Design Participativo, para customização e desenvolvimento de uma atividade do GCompris sobre pensamento computacional?

## 1.2 Objetivos

Este trabalho tem como objetivo avaliar o uso do Design *Thinking* com crianças, numa perspectiva do Design Participativo, para customização e desenvolvimento de uma atividade do GCompris sobre pensamento computacional. Para alcançar o objetivo geral deste trabalho elencamos os seguintes objetivos específicos:

- Customizar e desenvolver uma atividade para o GCompris utilizando o Design *Thinking* na perspectiva do Design Participativo.
- Avaliar a utilização de Design Participativo com crianças na customização da suíte GCompris.

## 1.3 Organização do Trabalho

Este trabalho está organizado em cinco seções distribuídas da seguinte forma. Na seção 1 está apresentado a problemática e os objetivos deste trabalho. Na seção 2 apresentamos a revisão de literatura realizada acerca dos temas tratados neste trabalho e que fundamentaram nossa pesquisa. Na terceira seção, apresentamos a metodologia usada para o procedimento de coleta de dados e para o procedimento de avaliação dos dados, o tipo de pesquisa e os sujeitos da pesquisa. A seção 4 descreve os resultados do trabalho com base nos objetivos definidos e como se deu o processo de aplicação deste trabalho. E, na seção 5, as conclusões deste trabalho, limitações e sugestões de trabalhos futuros.

## 2 Revisão de Literatura

Nesta seção são apresentados os principais conceitos utilizados no desenvolvimento desta pesquisa. Na seção 2.1 apresentaremos o conceito software livre e as características de um software livre. Na seção 2.2 apresentaremos o GCompris e seu funcionamento. Na seção 2.3 apresentaremos o conceito de pensamento computacional e sua contribuição para crianças. Na seção 2.4 apresentamos a importância do desenvolvimento de artefatos digitais para crianças. Na seção 2.5 o Design Participativo e como o mesmo contribui para o processo de desenvolvimento de sistemas. E por fim na seção 2.6 apresentamos o Design *Thinking*, uma das ferramentas utilizadas neste processo.

### 2.1 Software Livre

Nos últimos anos o movimento de publicação de software livre tem ganhado cada vez mais notoriedade. Este modo de produção do software tem resultado em produtos de qualidade e tem ganhado muito espaço em alguns setores do mercado de software. Hexsel afirma que um código de um software livre torna-se um bem público e à disposição da sociedade. Este tipo de software geralmente é desenvolvido por comunidades compostas por membros que estão distribuídos geograficamente e que se comunicam por alguma ferramenta que reúne a comunidade participante (como listas de discussão, e-mail e etc) com a finalidade de contribuir para a evolução do software em todos os aspectos (HEXSEL, 2002).

Conforme a descrição do Software Livre ou *Free Software* de Richard Stallman, adotada pela Free Software Foundation (FSF, 2019), devemos entender aquele software que respeita a liberdade e senso de comunidade dos usuários. O que significa que os usuários possuem a liberdade de executar, copiar, distribuir, estudar e melhorar o software. Assim, Software Livre é uma questão de liberdade e não de preço, não se referindo ao software como grátis. Um programa pode ser considerado um software livre se os usuários possuem quatro liberdades essenciais:

- A liberdade de executar o programa como você desejar, para qualquer propósito (liberdade 0).
- A liberdade de estudar como o programa funciona, e adaptá-lo às suas necessidades (liberdade 1). Para tanto, acesso ao código-fonte é um pré-requisito.
- A liberdade de redistribuir cópias de modo que você possa ajudar outros (liberdade 2).

- A liberdade de distribuir cópias de suas versões modificadas a outros (liberdade 3). Assim o acesso ao código fonte é um pré-requisito.

Um programa livre deve oferecer as quatro liberdades a qualquer usuário que obtenha uma cópia do software, desde que o usuário atenda às condições de licença livre que cobre o software. Para que a liberdade 1 e 3 ocorram é necessário que o código fonte do programa esteja disponível. Conseqüentemente, o acesso ao código-fonte é uma condição necessária para o software livre. A liberdade de utilizar um programa significa a liberdade para qualquer tipo de pessoa, seja física ou jurídica, de utilizar o software para qualquer tipo de trabalho ou atividade, seja ela paga ou não.

A liberdade de redistribuir cópias deve incluir formas binárias ou executáveis do programa, assim como o código-fonte tanto para as versões originais quanto para as modificadas.

Todavia, certas regras sobre a maneira de distribuir software são aceitáveis quando elas não entram em conflito com as liberdades centrais. O *copyleft*, por exemplo, é um método geral para tornar um programa livre e exigir que todas as versões modificadas e extensões do programa sejam livres também. Segundo a Free Software Foundation, qualquer um que distribui o software, com ou sem modificações, tem que passar adiante a liberdade de copiar e modificar novamente o software. As licenças de muitos softwares são desenvolvidas para cercear a liberdade de uso, compartilhamento e mudanças. A Licença Pública Geral (GNU) ao contrário, pretende garantir a liberdade de compartilhar e alterar softwares de livre distribuição - tornando-os de livre distribuição também para quaisquer usuários. A Licença Pública Geral aplica-se à maioria dos softwares da Free Software Foundation e a qualquer autor que esteja de acordo de utilizá-la.

Segundo Hexsel , algumas vantagens da utilização de software livre (HEXSEL, 2002):

- custo social baixo
- não se fica refém de tecnologia proprietária
- independência de fornecedor único
- desembolso inicial próximo de zero
- não obsolescência do hardware
- robustez e segurança
- possibilidade de adequar aplicativos e redistribuir versão alterada
- suporte abundante e gratuito

- sistemas e aplicativos geralmente muito confiáveis

Segundo Melo (2002, p. 64) “os melhores programas são aqueles em que o aprendiz usa seu raciocínio e pesquisa informações para resolver os desafios”. O uso de softwares contribui para auxiliar os professores na sua tarefa de orientar a construção do conhecimento e adquirir uma nova maneira de ensinar cada vez mais criativa, dinâmica, auxiliando novas descobertas e investigações. Vale ressaltar que o computador atua como facilitador para o aprendizado das atividades educativas contribuindo para o aprendizado e para superação das limitações dos alunos, porém não como elemento principal deste aprendizado.

O uso de software livre pode contribuir significativamente para a propagação e o uso em larga escala de soluções eficientes e de baixo custo. Atualmente existe uma enorme quantidade de softwares que estão disponibilizados gratuitamente e podem ser aplicados em uso educacional, um destes é o GCompris uma suíte educacional de qualidade que possui uma grande quantidade de atividades de cunho educacional, e que está em constante evolução.

## 2.2 GCompris

O GCompris é um software que consiste em um conjunto de atividades de cunho educacional com a finalidade de tornar o processo de aprendizagem mais lúdico e produtivo. Teve seu início no ano 2000 como fruto do trabalho do francês Bruno Coudoin motivado por fornecer aplicativos educacionais nativos para Linux. O software ganhou o Free Software Awards que ocorreu na França, na cidade de Soisson em Maio de 2003, na categoria de “programas educativos” (FREDERICO, 2004).

O GCompris é um software destinado à crianças de idade entre 2 e 10 anos, cuja característica principal das atividades é o caráter lúdico, mas sem deixar de lado o caráter educacional (GCOMPRIS, 2016). Atualmente o GCompris possui mais de 100 atividades que estão divididas em 7 categorias, classificadas como:

Categoria	Atividades
Descoberta do Computador	teclado, mouse, touchscreen, ...
Leitura	letras, palavras, prática de leitura, digitação de texto, ...
Aritmética	operações com números, memorização de tabelas, enumeração. tabelas de entrada dupla, ...
Ciências	controle do canal, ciclo da água, energia renovável, ...
Geografia	países, regiões, cultura, ...
Jogos	xadrez, memória, ligue 4, forca, jogo da velha, ...
Outros	cores, formas, Braille, aprenda a dizer as horas, ...

Quadro 2.1: Categorias GCompris

Fonte: GCompris 2019

O GCompris é distribuído sob licença GNU/GPL, sendo um pacote GNU oficial. Este software está disponível para diversas plataformas GNU/Linux, Windows, MacOS e outras versões do Unix. Sua versão mais atual, até a escrita deste trabalho, é a 0.97 atualizada em 01 de Dezembro de 2019. Para utilização do GCompris é necessário um computador com os requisitos mínimos disponíveis: processador Pentium 3500 Mhz, memória RAM 512 MB, não é necessário placa aceleradora 3D (GCOMPRIS, 2016). As instruções para instalação nos diversos sistemas operacionais estão disponíveis no site do GCompris na página de downloads <sup>1</sup>.

Para GNU/Linux o GCompris oferece pacotes independentes para os sistemas de 64bits ou 32bits. As versões deverão funcionar com qualquer distribuição sendo necessário ter, pelo menos, o ‘kernel’ de Linux 3.10 e do GStreamer 1.0 e o site do GCompris traz todas as instruções necessárias para a instalação e configuração do software. O software conta ainda com um pacote autônomo que pode ser instalado no Raspberry Pi, este pacote é uma versão beta e foi testado apenas no Raspberry Pi 3. Estas distribuições facilitam na propagação e uso deste software em escolas e laboratórios de ensino, considerando a redução de custos com licenças de softwares.

Vale ressaltar que o GCompris é um software livre, e portanto, o mesmo pode ser adquirido, modificado, estudado e redistribuído por qualquer dos seus usuários sob os termos de licença da GNU/GPL.

O projeto GCompris é armazenado e desenvolvido pela comunidade KDE (KDE, 2019a) que, através do esforço de milhares de contribuintes, objetiva oferecer uma experiência inovadora em computação e software livre. Os colaboradores da comunidade KDE podem se envolver nas atividades como: relatores de problemas do software, triagem de bugs, desenvolvedores, tradutores, design de interface visual e humano, colaboradores da documentação, suporte ao usuário, promotores do software, web design, colaboradores no processo de transformação para acessibilidade, e doadores de recursos para manutenção do software.

A comunicação destes colaboradores se dá por uma variedade de locais diferentes da comunidade do KDE a depender do tópico da discussão, *bugs* e novos recursos devem ser discutidos em bugs.kde.org; para rastrear e discutir o trabalho usa-se Phabricator (KDE, 2019b), sistema de gerenciamento de tarefas e revisão de patches do KDE; discussões curtas e direcionadas a objetivos podem ser feitas no *Internet relay*, *Matrix* ou *Telegram*; para iniciar uma discussão a longo prazo também estão disponíveis outros tipos de canais onde é possível acompanhar desenvolvimento do software, base de usuários, suporte da comunidade e fóruns. Além do mais, é importante para os novos integrantes assinarem as comunidades do kde-community, uma lista de discussão para informações não técnicas e relevantes para a comunidade; e o kde-devel, para discussões de todos os problemas de

<sup>1</sup> <https://gcompris.net/downloads-ptBR.html>

desenvolvimento relacionados ao KDE.

Os desenvolvedores colaboradores do projeto devem optar por uma distribuição GNU/Linux de sua escolha para maior facilidade no processo de colaboração. Os mesmos armazenam seus arquivos em repositórios online oficial do KDE ou no repositório do GitHub, onde estes são avaliados para serem aceitos. A versão mais recente do GCompris é baseada na tecnologia *Qt Quick*, a biblioteca padrão para fazer aplicativos QML. Com a versão Gtk+ não era possível entregar aos usuários um sistema executável em Desktops e Tablets justificando portanto a reescrita do software em uma nova estrutura. O desenvolvimento do software é feito usando uma mistura de QML, JavaScript e C++ (GCOMPRIS, 2017).

Assim, diante do exposto, podemos perceber que o GCompris possui uma série de características que o torna uma ferramenta educacional que contempla diversas áreas do conhecimento, além de, por ser livre, não gerar custos ao ser adotado nas redes de ensino e possuir uma comunidade de desenvolvimento ativa. No entanto, foi observado que, na versão do GCompris utilizada nesta pesquisa, foi implementada apenas uma atividade relacionada ao desenvolvimento do pensamento computacional. Tal conceito será discutido na próxima seção.

## 2.3 Pensamento Computacional

O pensamento computacional é um método para solução de problemas baseado nos fundamentos e técnicas da Ciência da Computação (WING, 2006). Sua proposta é aplicar habilidades utilizadas para criação de programas computacionais como metodologia para resolução de problemas gerais. Outras áreas do conhecimento também podem se beneficiar de parte das competências da Computação. Para Wing (2006) essas competências são agrupadas nas seguintes características:

- *Conceituar ao invés de programar* - Resolver um problema aplicando o pensamento computacional significa reduzir um problema relativamente grande em problemas menores e mais simples de resolver. Exigindo a capacidade de pensar de forma abstrata e em diferentes níveis.
- *É uma habilidade fundamental e não utilitária* - O pensamento computacional é algo que permite a resolução de problemas diversos usando os computadores, um recurso presente na sociedade atual, e por isso deve ser desenvolvido por todos os estudantes.
- *É a maneira na qual as pessoas pensam, e não os computadores* - A resolução de problemas através do pensamento computacional é um tratamento específico do problema de forma que ele possa ser resolvido por computadores, e não uma redução do raciocínio para simular o processamento do computador.

- *Complementa e combina a Matemática e a Engenharia* - a definição de Wing considera o aporte da Matemática e da Engenharia para a computação e reconhece as particularidades trazidas pelo enfoque computacional;
- *Gera ideias e não artefatos* - O pensamento computacional não deve ter necessariamente como resultado final a produção de software e hardware e reconhece que os conceitos fundamentais da Computação estarão presentes para resolver problemas em vários contextos do cotidiano;
- *É para todos, em qualquer lugar* - O pensamento computacional deve ser útil para todas as pessoas em diversas aplicações.

Introduzir o pensamento computacional no ensino básico pode instigar o interesse pela área da Computação, mostrando sua importância na resolução de problemas do mundo atual. Ainda que nem todos os alunos ingressam na área da computação posteriormente, mas essa estratégia educacional cria e intensifica relações interdisciplinares com outras áreas. Na visão de Papert (1985), comunicar-se com o computador é um processo natural do ser humano. É ele quem deve programar o computador e não ao contrário. Ou seja, nesta perspectiva o ser humano assume o domínio e controle sobre a máquina. Programar significa comunicar-se com o computador em uma linguagem que ambos, homem e máquina, entendam.

Em seu estudo, Nunes (2011) afirma que o raciocínio computacional é intuitivo no ser humano e se manifesta já na infância. Portanto, a criança naturalmente raciocina de forma computacional. Entretanto, a não abordagem na educação básica desta habilidade, gera como consequência a perda do raciocínio computacional ao longo do crescimento e da formação do indivíduo. A organização Computer Science Teachers Association (CSTA, 2019) propõe que a introdução de conceitos de Ciência da Computação na educação básica é fundamental para manter o raciocínio computacional das crianças, pelo seu caráter transversal à todas as ciências, para formar cidadãos neste importante ramo da ciência, para dominar suas aplicações, para viver num mundo cada vez mais globalizado e para tornar o País mais rico e mais competitivo na área de Tecnologia da Informação. Tendo em vista que a sociedade está cada vez mais tecnológica deve haver uma conscientização para a inclusão das habilidades tecnológicas nos currículos escolares (HOLANDA; KOH; REPENNING, 2017).

Schumacher Barcelos (2012), através do estudo de Basawapatna, Koh e Repenning (2011) define pensamento computacional baseando-se na identificação de padrões do funcionamento de agentes de software em jogos digitais. Os agentes são programados por estudantes em atividades didáticas, usando uma linguagem de programação visual e baseada na construção de blocos. Trata-se de uma proposta mais específica, porém seu objetivo é identificar a possível transferência, pelos estudantes, do conhecimento aos

padrões para o domínio da simulação de fenômenos físicos e biológicos. Os livros que ensinam programação estruturada em sua maioria abordam primeiramente o conceito de algoritmos, uma sequência finita de instruções bem definidas (FORBELLONE, 1993), e em seguida o conceito de estruturas de decisão e estruturas de repetição, estruturas usadas para executar, repetidamente, uma instrução ou bloco de instrução por determinada quantidade de vezes (FILGUEIRAS, 2015).

Essa metodologia também é abordada no ensino do pensamento computacional para crianças, facilitando a compreensão da programação pela mesma e auxiliando na apropriação do conhecimento e do exercício do raciocínio lógico através processo de repetição de atividades e desafios. Segundo Papert (1985), uma característica relevante da programação de computadores é que a criança dificilmente vai acertar na primeira vez que programar, então, ela vai persistindo e resolvendo os problemas até chegar ao resultado que pretende. Assim, os softwares utilizados devem possuir características desafiantes, mas, ao mesmo tempo, lúdicas e estimulantes para que gere engajamento das crianças. Para tanto, é importante levar em consideração as recomendações de design para o desenvolvimento de artefatos digitais para e com crianças, assunto este que será discutido na próxima seção.

## 2.4 Desenvolvimento de Artefatos Digitais para Crianças

A participação do usuário final no processo de criação do artefato desde o início, permite que o produto seja moldado às suas necessidades e expectativas. O design centrado no usuário é um conceito que se baseia na ideia de que todo e qualquer processo de design precisa ter o usuário como origem e como finalidade na sua concepção e planejamento (MARTINS et al., 2017). Para Norman (2006) o Design Centrado no Usuário é aquele capaz de reunir produtos compreensíveis e facilmente utilizáveis através da simplificação de tarefas. Fernandes (2018) ressalta que o projeto que é desenvolvido para crianças deve ser diferente do projeto para adultos, pois crianças são diferentes, elas pensam diferentes, e possuem vidas diferentes e isso interfere diretamente no ato de projetar, devido às diferentes estratégias e formas de se relacionar com o produto.

Segundo Idler (2014 apud MARTINS et al., 2017) existem cinco pontos onde as crianças são diferentes dos adultos:

1. Desenvolvimento físico: A coordenação olho-mão e a coordenação motora são diferentes dos adultos, afetando a interação com os equipamentos;
2. Desenvolvimento Cognitivo: as habilidades cognitivas da criança estão em desenvolvimento na criança, o que afeta os métodos de coleta e os resultados de pesquisas;
3. Desenvolvimento social: crianças podem ser muito individualistas, mas a maturidade altera sua percepção do outro;

4. Concentração: crianças podem ser naturalmente estimuladas mas podem desconcentrar-se com a mesma facilidade;
5. Experiencia: crianças vivenciaram menos do que adultos, e não estão treinadas a aplicar alguns conhecimentos à novas situações, por isso a necessidade de oferecer informações suficientes, para não gerar incerteza e desencorajamento à atividade proposta.

Na construção de produtos envolvendo a participação de crianças é necessário levar em consideração suas características próprias e incluir estas no processo de uma forma geral. Idler (2014) também propõe cinco características que devem ser levadas em consideração ao colocar a criança como centro do desenvolvimento:

1. Entretenimento: a maioria das crianças usam as mídias para entretenimentos, e portanto esperam um alto índice de entretenimento nos produtos oferecidos à ela;
2. Apelo visual: as crianças usam as mídias por motivos diferentes dos adultos, o que faz-se necessário influência externa, essa motivação pode vir de um amigo ou parente, mas não basta apenas a influência, após a utilização inicial o apelo visual continua importante, porém é necessário que o artefato tenha uma resposta positiva para incentivar a continuação do uso.
3. Usabilidade: Se o aplicativo não for intuitivo ou não funcionar da maneira que esperam, dificilmente elas se darão o trabalho de ler as instruções ou se adaptarem ao uso da nova interface;
4. Conteúdo apropriado à idade: o conteúdo deve combinar com os interesses naturais e modelos mentais para a idade dos usuários, tomando cautela com conteúdos muito simples pois serão considerados tediosos, e conteúdos muito difíceis pois são considerados inviáveis;
5. Aprendizado incentivado: mídias digitais oferecem possibilidades de ensinar ludicamente, conteúdo novo e atrativo torna o aplicativo divertido e cativante.

Como Melo, Baranauskas e Soares (2008) afirma em seu trabalho, o desenvolvimento de sistemas de informação voltados ao público infantil envolve: a clarificação dos significados que as crianças dão para o mundo e para os artefatos tecnológicos com os quais interagem em seu dia-a-dia; a representação do entendimento do designer para estes significados, com vistas a orientar o processo de design do novo artefato em desenvolvimento; a proposição e avaliação iterativa de soluções de design com o envolvimento das próprias crianças. A participação da criança na criação de um produto permite que a mesma se expresse em relação ao serviço que será oferecido à esta posteriormente.

## 2.5 Design participativo

O processo de desenvolvimento de sistemas de informação deve focar as necessidades do usuários finais, garantindo uma satisfação efetiva por meio do alto grau de usabilidade, acessibilidade e qualidade do mesmo. É possível usar o design participativo para coletar, analisar e projetar informações neste processo.

Muller, Haslwanter e Dayton (1997) afirmam que o design participativo apresenta-se como método de investigação do designer, oferecendo técnicas para ajudá-lo a propor soluções que atendam às necessidades do usuário final, com a participação ativa de diferentes partes interessadas no sistema da informação, incluindo os próprios usuários. O design participativo também abrange questões relacionadas à maneira de como os usuários pensam e agem, enfatizam a importância de cada participante ao decorrer de todo o processo de desenvolvimento, e considera que o usuário do sistema sabe quais funções são úteis e quais são as prioridades para sua prática profissional.

O design participativo tem relações diretas com outras áreas relacionadas como o design centrado no usuário, design de interação, design cooperativo, design de interface, arquitetura da informação etc. As principais diferenças é que nessas áreas levam-se em consideração os dados relacionados aos usuários, focando na utilização do produto pelo mesmo. Entretanto os usuários não fazem parte do processo de desenvolvimento do sistema (CAMARGO; FAZANI, 2014). No design participativo se torna central a participação do usuário no processo de produção da tecnologia. As práticas participativas são destinadas para serem aplicadas durante o ciclo de vida do software, considerando os participantes envolvidos nestas atividades. A participação dos interessados pode auxiliar na usabilidade do sistema pois abrir espaços para as habilidades, experiências e interesses dos trabalhadores no design de sistemas pode fazer com que estes sistemas sejam mais úteis e melhor integrados às práticas dos usuários dentro das organizações (NERIS, 2010).

Atualmente, as crianças são potenciais consumidoras de tecnologias digitais, principalmente nas áreas de entretenimento e educação. Melo, Baranauskas e Soares (2008) mostram em seu estudo que o desenvolvimento de sistemas de informação voltados ao público infantil envolve: a clarificação dos significados que as crianças dão para o mundo e para os artefatos tecnológicos com os quais interagem em seu dia-a-dia; a representação do entendimento do designer para estes significados, com vistas a orientar o processo de design do novo artefato em desenvolvimento; a proposição e avaliação iterativa de soluções de design com o envolvimento das próprias crianças. Druin (2002) mostra que existem muitos outros papéis que as crianças não só podem mas devem desempenhar no design de novas tecnologias: usuário, testador, informante e parceiro de design. No papel de usuária, a criança contribui usando dispositivos tecnológicos sob a observação do pesquisador, que procura entender o impacto das tecnologias para as crianças, para que elas sejam futuramente adaptadas para adequarem-se melhor. No papel de testadora, a criança testa

protótipos em construção, que serão ajustados de acordo com seu feedback, antes de serem lançados no mercado. Como informante, a criança tem um papel em vários estágios do processo de design, em momentos determinados pelos projetistas. Esse papel pode incluir teste de protótipos, rascunhos de ideias para soluções tecnológicas, opiniões e sugestões. Finalmente, como parceira de design, a criança contribui para o processo em nível de igual importância aos demais *stakeholders* (interessados no projeto), ainda que de maneira adequada às particularidades de sua idade.

Participando diretamente no processo de design, a criança pode expressar suas idéias clarificando o que faz sentido a ela, de que gosta e o que a interessa. Compartilhando seu universo e entendendo-o, podemos, como designers, criar produtos que respeitem suas expectativas como usuários e seu entendimento como autores. Além do mais, as crianças podem se beneficiar da aprendizagem centrada no design a partir de suas experiências no processo. Possibilitar a expressão do usuário é um dos principais meios para se alcançar o engajamento deles e o sentimento de pertencimento ao grupo de trabalho. Spinuzzi (2005) sintetiza três estágios básicos para aplicar o design participativo:

- *Exploração inicial do trabalho* - estágio em que os desenvolvedores se reúnem com os usuários para uma primeira aproximação e familiarização com o seu modo de trabalho, usando atividades de observação e entrevista para melhor identificação das necessidades do usuário.
- *Processo de descoberta* - fase em que os desenvolvedores e usuários empregam técnicas para entender o problema e priorizar as etapas de trabalho. Atividades como storyboard, jogos organizacionais, entre outros, podem ser exploradas nesta fase.
- *Prototipação* - etapa em que os desenvolvedores e usuários de forma iterativa criam artefatos a fim de entender os requisitos para solução do problema. Podem ser exploradas diferentes técnicas de prototipação, sejam digitais ou em papel.

Com a abordagem de design focado na participação da criança como autor, proporcionamos a esta a oportunidade de interagir com outras crianças exercendo seu senso crítico, conhecendo novas tecnologias, e explorando novas habilidades, da mesma maneira proporcionamos ao designer uma melhor compreensão do mundo infantil e uma perspectiva de visão sobre este. Para sistematização das atividades realizadas no Design Participativo iremos utilizar o Design *Thinking*, que será melhor discutido na próxima seção.

## 2.6 Design *Thinking*

O Design *Thinking* é um processo para resolução de problemas, desenvolvido colaborativamente e centrado no humano, segundo Martin (2010), o Design *Thinking* promove

o equilíbrio entre o pensamento analítico e intuitivo, o que permite às organizações gerar inovações para aumentar eficiência e competitividade. Além disso, pode ser considerado uma ferramenta para o desenvolvimento de criatividade e inovação. Brown (2010) aborda o Design *Thinking* como a junção das habilidades aprendidas pelos designers ao longo do tempo, na busca pela correspondência entre as necessidades humanas com os recursos técnicos disponíveis. O Design *Thinking* integra o que é viável (busca por projeto sustentável), desejável (faz sentido para as pessoas) e praticável (é possível de realizar). A partir disto podemos iniciar o Design *Thinking* em sua aplicação que, de acordo com Brown (2010), possui como base:

- Inspiração: o problema ou oportunidade que motiva a busca de soluções;
- Ideação: é compreendido como o processo de gerar, desenvolver e testar ideias;
- Implementação: o caminho que conduz a ideia a partir da fase de projeto até chegar à vida das pessoas.

Ainda segundo o autor os projetos podem seguir um loop através desses espaços dependendo do que a equipe define.

A técnica de observação é uma das ferramentas utilizadas para a divisão do problema em partes para melhor compreensão da equipe. Alinhada à observação é necessário a empatia, Oliveira (2014) afirma que ao utilizar esta técnica para compreensão dos problemas, os dados qualitativos sobrepõem-se aos dados quantitativos, pois o maior interesse está em entender os sentimentos e as emoções que foram despertados no grupo. A colaboração é um dos tripés do Design *Thinking* e se caracteriza pelo desenvolvimento de produtos “com” os clientes e não “para” os clientes. As pessoas passam de consumidores para participantes ativos do desenvolvimento e criação do produto. O Design *Thinking* também visa obter um protótipo da ideia para que ela tome forma e seja testada em seus pontos fortes e fracos, para isso usamos a experimentação. As abordagens do Design *Thinking* obriga as equipes a estarem constantemente visualizando e testando novas ideias por meio da experimentação, a fim de colocar novas soluções no mercado de modo inteligente.

Para Lockwood (2009), Design *Thinking* é, essencialmente, um processo de inovação centrado no ser humano que enfatiza observação, colaboração, rápido aprendizado, visualização de ideias, construção rápida de protótipos de conceitos e análise de negócios dos concorrentes, para influenciar a inovação e a estratégia de negócio. Brown (2010) complementa a visão de Lockwood (2009) quando diz que o Design *Thinking* não é apenas uma proposta centrada no ser humano, sendo profundamente humana pela própria natureza, pois se baseia na capacidade do ser humano em ser intuitivo, reconhecer padrões, desenvolver ideias que tenham um significado emocional, ultrapassando as barreiras do funcional.

Segundo Jones (1978 apud OLIVEIRA, 2014), o processo do design é dividido em etapas com pensamentos diferentes: divergência, transformação e convergência. Divergência trata-se da ampliação dos limites e da obtenção de um espaço de investigação para ampliar o conhecimento sobre o problema, a partir disso surge a capacidade de análise e, conseqüentemente, a habilidade de observar, focando no descobrimento das ideias. A transformação é composta de *insights* que contribuem para converter o design em uma tarefa, dividindo-se em métodos de investigação de ideias e auxiliado na elaboração dos conceitos para exploração do problema. Na convergência desenvolve-se a capacidade de síntese a partir das informações levantadas na análise. Na divergência está o pensamento racional e o uso de conhecimentos explícitos para obter informações do contexto.

Das técnicas usadas no Design *Thinking* o *brainstorming* é uma das mais comuns, consiste em estimular a geração de um grande número de ideias em um curto espaço de tempo (VIANNA, 2012). Brown (2010) considera o *brainstorming* como a principal técnica que o design *thinker* possui para o desenvolvimento de produtos criativos. As recomendações de Vianna (2012) para o sucesso no uso de *brainstorming* é prezar pela qualidade da quantidade de ideias, evitar julgar as ideias, aceitar ideias ousadas, combinar adaptar, aprimorar e transformar as ideias que foram propostas.

Oliveira (2014) afirma que é necessário que os sistemas educacionais promovam os valores sociais e as atitudes, além das experiências construtivas. Na prática de acordo com o Instituto EducaDigital (2014) a aplicação do Design *Thinking* no ensinar e no aprender se dá com as seguintes fases:

- Descoberta: reconhecer que se tem um desafio e como se pode abordá-lo;
- Interpretação: perceber que se aprendeu alguma coisa e como se pode interpretá-la;
- Ideação: enxergar uma oportunidade e pensar como pode criá-la;
- Experimentação: identificar a sua ideia e como esta pode ser concretizada;
- Evolução: experimentar algo novo e pensar como é possível aprimorá-lo.

Como afirma Oliveira (2014) Design *Thinking* é um modelo de pensamento. Significa ter a possibilidade de acreditar que se pode fazer a diferença, desenvolvendo um processo intencional para se chegar ao novo, com soluções criativas e criar impacto positivo.

## 3 Percurso Metodológico

De acordo com os objetivos e fundamentação teórica desta pesquisa, apresentados no capítulo de introdução, optamos por adotar uma perspectiva qualitativa, através de um estudo de caso e entendida como uma pesquisa aplicada de caráter descritiva e exploratória. Cada uma dessas opções será justificada nos tópicos que se seguem ao longo deste capítulo.

### 3.1 Tipo de Pesquisa

De acordo com Bogdan e Biklen (2006, p. 4), a pesquisa qualitativa possui cinco características principais: 1) Naturalista; 2) Os dados são descritivos; 3) Há uma preocupação com o processo; 4) A abordagem é indutiva e; 5) Há uma busca pelo significado. Esta pesquisa pode ser classificada como naturalista, pois foi no ambiente natural (escolar) o local de onde se obteve os dados. Além disso, o investigador foi o ator principal para a recolha desses dados. Os dados apresentados nesta pesquisa são de natureza descritiva, ou seja, os dados recolhidos tomaram a forma de palavras ou imagem ao invés de números e incluem transcrições de entrevista, notas de campo e imagens. Houve também uma preocupação maior com o processo do que com o resultado final e os dados foram analisados de forma indutiva, não procurando confirmar ou infirmar hipóteses prévias. Além disso, houve uma preocupação com a perspectiva do participante e os significados que os mesmos dão a ação. Com base no exposto e seguindo características apresentadas por Bogdan e Biklen (2006, p. 4), podemos classificar esta pesquisa como possuidora de uma abordagem qualitativa.

Quanto aos procedimentos esta pesquisa pode ser classificada como estudo de caso. Para Merriam (1998, p. xiii) um estudo de caso é "uma descrição e análise intensiva e holística de um fenômeno limitado, como um programa, uma instituição, uma pessoa, um processo ou uma unidade social". Apesar do estudo de caso não ter uma definição única e amplamente aceita, é possível identificar algumas características que diferenciam o estudo de caso de outros métodos. Tal identificação foi feita por Coutinho (2014, p. 336) ao sintetizar o estudo de caso qualitativo como uma investigação empírica, que se baseia no raciocínio indutivo, não experimental, sendo dependente fortemente do trabalho de campo e baseado em fontes de dados múltiplas e variadas.

### 3.2 Sujeitos da Pesquisa

A pesquisa foi desenvolvida com um grupo de 15 alunos, com idades entre 11 e 12 anos, estudantes da rede particular de ensino da cidade de Garanhuns, sendo um

grupo diversificado de meninos e meninas. Alguns dos integrantes já tiveram contato com sites e repositórios que contêm atividades de ensino de programação para crianças. Toda pesquisa foi realizada no Laboratório de Pensamento Computacional do Colégio Diocesano de Garanhuns. A participação dos alunos foi previamente autorizada pelos pais ou responsáveis. A instituição que acolheu o desenvolvimento da pesquisa foi informada, por documento de abertura de projeto, sobre como se daria a realização do mesmo e estava de acordo com todos os procedimentos adotados. As identidades dos sujeitos da pesquisa, bem como das crianças e profissionais foram preservadas na pesquisa.

### 3.3 Procedimento de Coleta dos Dados

Como procedimento para construção dos dados optou-se pela Observação Participante que é um método qualitativo com raízes na pesquisa etnográfica tradicional, cujo objetivo é ajudar os pesquisadores a conhecer diferentes perspectivas das populações investigadas (MACK, 2005, p. 13). Nesse tipo de pesquisa o que é observado através dos olhos do pesquisador deve ser registado em algum meio que possa ser analisado posteriormente. Assim, os principais instrumentos de coleta de dados na observação participante são os diários de bordo e as notas de campo (DENSCOMBE, 2014; GIBBS, 2009).

Como instrumentos de pesquisa, foram utilizadas as notas de campo, os artefatos produzidos pelos participantes da pesquisa e uma entrevista de grupo focal. De acordo com Gibbs (2009, p. 46), as notas de campo são anotações realizadas no ambiente de pesquisa e que podem ser produzidas enquanto ainda se está em campo ou imediatamente após sair dele, registrando palavras, frases ou ações. Além disso, as notas de campo não são planejadas nem estruturadas.

Para complementar as notas de campo, também foi utilizada uma entrevista de grupo focal. Caplan (1990 apud DIAS, 2000), define os grupos focais como pequenos grupos de pessoas reunidos para avaliar conceitos ou identificar problemas. A vantagem desta técnica é que os participantes podem interagir em torno de uma ou mais problemáticas, como argumenta Dias (2000, p. 4) “a sinergia entre os participantes leva a resultados que ultrapassam a soma das partes individuais”. Por fim, também serão considerados as produções de artefatos elaborados pelos participantes, tais como imagens, textos e rascunhos. Na próxima seção serão apresentados os resultados obtidos através do percurso metodológico descrito anteriormente.

## 4 Resultados

Esta seção detalha os resultados obtidos na realização deste trabalho. Na seção 4.1 veremos os resultados obtidos através da aplicação das técnicas escolhidas e da metodologia definida para coleta de dados. Na seção 4.2 a exploração da utilização do design participativo e a análise dos dados incluindo a perspectiva dos usuários sobre o artefato gerado.

### 4.1 Customização Usando Design Participativo

Conforme apresentado na seção 2.4 deste trabalho, o design participativo é uma prática usada para identificar e coletar as necessidades dos usuários finais do produto, garantindo assim a usabilidade e qualidade do mesmo. Quando envolvemos a participação direta de crianças na co-autoria de produtos para o seu uso, damos a estas a oportunidade de serem ouvidas e de terem suas expectativas atendidas em relação ao produto desenvolvido. Além do mais, envolver a criança no processo de construção de design possibilita que a ferramenta seja melhor pensada, garantindo portanto uma melhor compreensão da mesma, envolvendo o que faz sentido à ela e o que a motiva.

Como vimos, Design *Thinking* é uma forma de resolver problemas de modo criativo e centralizando as pessoas, focando na geração de ideias inovadoras e que coloquem os usuários no centro da solução - o grupo de pessoas a que a resolução se destina. Além disso, o conceito também pode ser aplicado para desenvolver novos produtos e serviços, de acordo com as necessidades do público-alvo.

Dividido em várias fases, que não necessariamente devem ser executadas sequencialmente, o Design *Thinking* apresenta uma série de técnicas e ferramentas colaborativas utilizadas para a coleta e classificação de informações, permitindo aos usuários desenvolver a criatividade, colaboração e trabalho em equipe. Para viabilizar a construção do nosso processo de design participativo com o público infantil abordamos cinco etapas do Design *Thinking*, conforme apresentado na Figura 1:



Figura 1 – Etapas do Design *Thinking* usadas neste trabalho

Fonte: (MICHAELS, 2017) adaptado pelo autor

A pesquisa foi aplicada em 4 encontros seguidos com intervalos de uma semana, em cada encontro foram realizadas as etapas citadas acima distribuídas da seguinte forma:

Encontro	Atividade do GCompris	Etapas Desenvolvidas
Encontro 1	Programming Maze	1, 2, 3 e 4.
Encontro 2	Programming Maze Programming Loop	5 e 3. 1, 2, 3, e 4.
Encontro 3	Programming Maze Programming Loop	5. 5 e 3.
Encontro 4	Programming Loop	5

Quadro 4.1: Realização das atividades

Fonte: Dados da pesquisa (2019)

Para melhor compreensão das atividades de programação referenciadas nas seções a seguir consultar Apêndice B

#### 4.1.1 Etapa 1: Empatia

A fase de empatia deve despertar o interesse e envolvimento da parte dos colaboradores com o produto, identificando o que implica direta ou indiretamente na experiência do usuário, e suas principais necessidades gerando um ponto de vista a partir das mesmas.

Para esta fase a técnica de pesquisa exploratória foi aplicada para auxiliar a equipe no entendimento do contexto a ser trabalhado e fornecer insumos para a definição dos atores e ambientes que atendessem as expectativas dos usuários (PROVALORE, 2019).

Para imergir a equipe de estudantes participantes com a atividade, o GCompris foi apresentado destacando toda sua importância como um software livre internacional que proporciona às crianças de diferentes partes do mundo a oportunidade de testarem e reforçarem seus conhecimentos sobre diversos assuntos, garantindo que os participantes compreendessem o impacto positivo da colaboração deles no aprendizado destas crianças, incluindo os próprios participantes do projeto, e propiciando a familiarização dos membros da equipe com as realidades de uso do produto que seria explorado ao longo do projeto.

Inicialmente os alunos tiveram o seu primeiro contato com a suíte GCompris que foi apresentada à estes como um todo, onde eles usaram algumas atividades pertencentes ao mesmo. O intuito de proporcionar este contato com o GCompris em sua totalidade é para que os alunos envolvidos no projeto tivessem conhecimento do estilo de atividades do GCompris e de certo modo guiar a perspectiva deles para que as sugestões não destoassem do restante do produto. Então eles usaram a atividade “*Programming Maze*” e a atividade “*Programming Loop*” em sua versão original, Figura 2, para observar e identificar padrões do software, levando em consideração o seu conhecimento prévio de outras ferramentas que possuem o mesmo objetivo, o ensino de programação para crianças.



Figura 2 – Versão original atividade *Programming Maze*

Fonte: Dados da pesquisa (2019)

Ao abrirem as atividades houve uma participação coletiva para apontar o que poderia ser melhorado ou o que não estava claro. Pontos como: cenário, botões, distribuição de espaço na tela, e outros foram ressaltados. Para melhor identificação das necessidades encontradas, uma lista, como mostra a Figura 3, foi criada coletivamente e visivelmente para todo o grupo, onde cada participante poderia contribuir com algum ponto negativo encontrado no uso da atividade, ou alguma dificuldade visual ou de usabilidade.

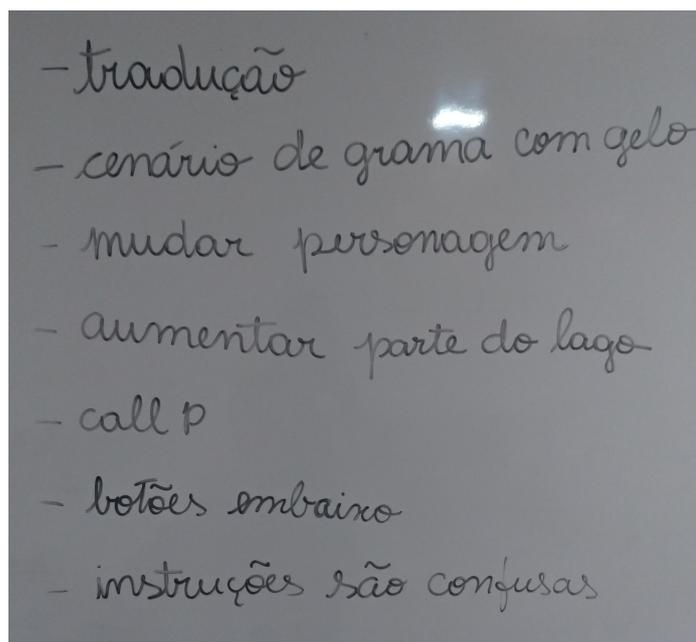


Figura 3 – Problemas encontrados no cenário original

Fonte: Dados da pesquisa (2019)

#### 4.1.2 Etapa 2: Definição

Na etapa de definição é iniciado o “pensar sobre” as necessidades encontradas na etapa anterior, identificando as principais necessidades e encontrando soluções para estas. É importante assegurar que nenhuma questão relevante seja negligenciada e que as soluções geradas não se distanciem do foco da demanda, determinando as primeiras ideias de solução a serem prototipadas. Para isto usamos a técnica de critérios norteadores, que são diretrizes balizadoras para o projeto, evidenciando aspectos que não devem ser perdidos de vista ao longo de todas as etapas do desenvolvimento das soluções (PROVALORE, 2019). Servem como base para a determinação dos limites do projeto e do seu verdadeiro propósito. Os critérios norteadores devem estar sempre presentes durante o desenvolvimento de um projeto porque parametrizam e orientam as soluções, evidenciando sua adequação ao escopo que deve ser respeitado.

Já na primeira fase é possível notar o envolvimento dos estudantes com o projeto e a disponibilidade de encontrar soluções para os problemas apontados assim a partida para a segunda fase se deu de modo natural. Da lista de necessidades criada na etapa anterior, as principais foram identificadas e colocadas em ordem de prioridade a serem discutidas. Para cada ponto negativo abordado pelos participantes, o orientador da tarefa sugeriu que uma solução fosse criada para aquele problema e que possíveis medidas poderiam ser tomadas para melhoria do ponto em questão, conferir Figura 4.

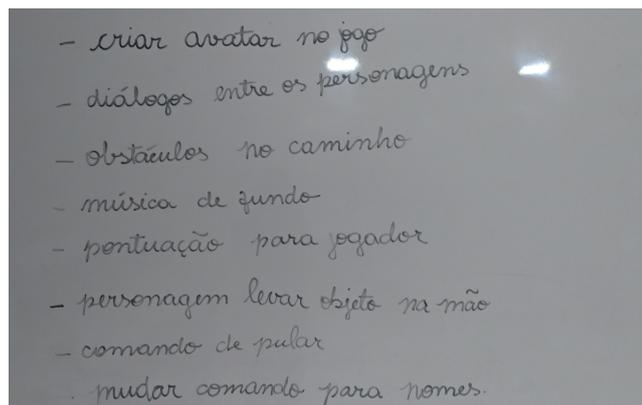


Figura 4 – Sugestões propostas como soluções

Fonte: Dados da pesquisa (2019)

É interessante ressaltar que ao falarem as justificativas para cada erro que estavam sendo apontados, os participantes se colocavam no lugar de crianças mais novas, argumentando que alguns elementos e sua representatividade não passavam clareza da sua finalidade, e que isto poderia influenciar no aprendizado e desempenho dos usuários dificultando este processo. Nesta segunda fase é reforçado com o que os estudantes devem colaborar, deixando claro que o nosso principal papel ali é contribuir para a melhoria do design e que mudanças no funcionamento do jogo não necessariamente serão realizadas.

#### 4.1.3 Etapa 3: Ideias

Na terceira etapa abordamos a ideação, onde o objetivo é estimular o surgimento de ideias de design para o artefato buscando a diversificação dos caminhos que podem ser seguidos. A etapa de ideação deve se dar de forma total e absolutamente colaborativa, incentivando novas criações de soluções a partir das ideias propostas. Portanto usamos a técnica de *brainstorming*, que consiste em estimular a geração de um grande número de ideias em um curto espaço de tempo (PROVALORE, 2019).

Quanto maior a quantidade de ideias geradas pela equipe, maior é a chance de produzir uma solução funcional. Críticas não devem atrapalhar o processo criativo e a geração de ideias ousadas, o foco deve estar em aprimorar e produzir ideias, adiando a avaliação para um momento posterior. E, em seguida, foi aplicada a técnica de workshop de cocriação, organizado na forma de uma série de atividades em grupo com o objetivo de estimular a criatividade e a colaboração, fomentando a criação de soluções para os problemas identificados numa sessão criativa de trabalho onde os participantes são convidados a interagir na geração de ideias de forma colaborativa (PROVALORE, 2019).

A partir da lista de soluções apresentadas os participantes se dividiram em equipes com três estudantes para a fase de ideação. Nesta terceira fase eles receberam papéis e lápis de colorir para colocarem suas ideias de melhorias, assim eles discutiram e reuniram ideias

entre os membros de cada equipe, porém cada um produziu sua colaboração em folhas individuais. Diferentes visões e sugestões surgiram nesta interação entre eles: um propunha a ideia ao outro buscando contribuir para solução dos defeitos que haviam encontrado, ao mesmo tempo que se colocavam como usuários do jogo. Soluções de melhoria de jogabilidade também foram propostas como, por exemplo, a criação de avatar para os participantes, a conquista de recompensas ao concluir cada nível do jogo sugerindo que estas pudessem ser trocadas por itens de interesse do jogador.

Dentre as sugestões de design foi ressaltado a importância de criação de uma história do jogo que pudesse proporcionar mais afinidade com os usuários e envolvê-los mais no cumprimento dos desafios, definindo melhor o cenário onde o jogo passaria. Sugestão de diálogos entre o personagem principal e o alvo, melhoria dos botões de instrução deixando mais claro a função de cada um, sugestão de retirada de botões padrão de navegação do GCompris, alegando que os mesmos traziam uma certa confusão e poluição do espaço da atividade recorrente, visto que a mesma já possui bastante informação para o seu prosseguimento. Sugestão de música de fundo para envolver o usuário na atividade, de áudio explicativo para crianças que ainda não dominam completamente o processo de leitura. Sugestões de instruções detalhadas sobre o conceito do jogo e sobre como funcionavam os recursos de instrução do mesmo. Reorganização dos elementos visuais do jogo na tela.

Ao concluírem seus esboços no papel os colaboradores apresentaram o que tinham produzido ao restante da turma e foi proposto uma filtragem nas ideias apresentadas do que se encaixaria na questão do design e do que seriam sugestões de melhoria de outro teor. Após a apresentação dos caminhos que haviam sido propostos, as ideias foram organizadas e agrupadas em categorias se alinhando com as necessidades que haviam sido apresentadas na primeira fase da proposta.

Primeiramente definimos qual seria a história do nosso jogo para definirmos o nosso cenário, dentre as propostas estavam cenário de fazendinha em que os personagens principais poderiam ser animais que iriam em busca do seu alvo que estaria representado como alimento do respectivo animal ou os seus filhotes, ou por um fazendeiro que estaria em busca de algum tipo de ferramenta ou da fazendeira; proposta de cenário no espaço sideral onde o personagem poderia ser um astronauta em busca de um planeta ou bandeira ou em busca de um foguete; ou poderia ainda se passar em um cenário de floresta, onde os animais poderiam ir em busca de alimento ou de suas presas. Após votação de ideias optou-se pelo cenário de fazenda para atividade de “*Programming Maze*”, e para a atividade “*Programming Loop*” o cenário de espaço sideral foi o escolhido. Em cima deste tema iniciamos a fase de prototipação.

Para manter um padrão entre as atividades de programação que estavam sendo desenvolvidas dentro do GCompris, os elementos funcionais de ambas as atividades ficaram posicionados de modo semelhante.

#### 4.1.4 Etapa 4: Protótipo

Na fase prototipação o objetivo é ter um esboço definido do design materializando a ideia definida na fase anterior sobre as hipóteses de soluções apontadas como satisfatórias para sanar ou mitigar os problemas dos usuários.

Usamos então a técnica de protótipo em papel, que são representações de interfaces gráficas com diferentes níveis de fidelidade (PROVALORE, 2019). Com a prototipação em papel é possível visualizar o fluxo de informações e a navegação no sistema, explorando as possibilidades de comunicação do produto e tangibilizando a apresentação da ideia aos usuários. O resultado deste protótipo foi feito em papel, executado à mão como um rascunho da solução funcional do produto.

A partir do momento em que tínhamos um caminho definido, os participantes se concentraram em trabalhar os elementos necessários para compor o jogo e fazer a substituição dos atuais elementos. Ainda divididos em equipes de três participantes os alunos começaram a desenhar os possíveis atores principais e secundários (alvos) do jogo, Figura 5, e os outros elementos que compunham a interface gráfica, gerando assim os protótipos em papel do jogo.



Figura 5 – Personagens sugeridos para atividade “*Programming Maze*”

Fonte: Dados da pesquisa (2019)

Também foram desenhadas sugestões de distribuição dos elementos do jogo na tela. A seguir, na Figura 6, um exemplo da proposta de posicionamento dos elementos para as atividades.

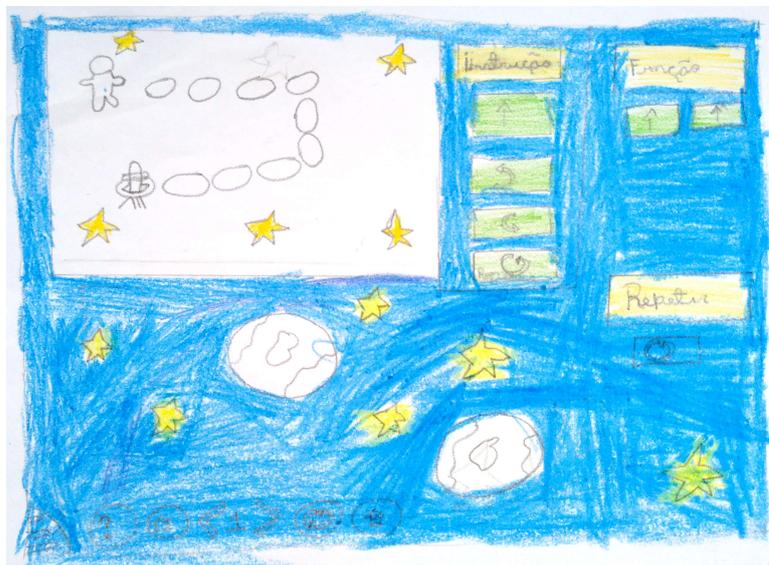


Figura 6 – Protótipo de cenário da atividade “*Programming Loop*”

Fonte: Dados da pesquisa (2019)

Muitas ideias foram geradas e refinadas para aplicar a melhor solução para as necessidades daquela atividade. Assim ao final da prototipação em papel algumas sugestões foram votadas e escolhidas pelos participantes como definitivas e outras ficaram em aberto para que na etapa seguinte, de testes, fosse observado sua percepção e aceitação da proposta de design.

#### 4.1.5 Etapa 5: Teste

E, por fim, na fase de teste avaliamos e validamos o que funcionou ou não funcionou no design sugerido, buscando identificar e aproveitar os diferentes feedbacks. A técnica de *shadowing* (técnica de sombra), foi aplicada, onde o usuário do sistema é observado, de maneira pouco intrusiva, ao longo de um período de tempo que incluía sua interação com o produto analisado, sem que o orientador da atividade interfira no contexto (PROVALORE, 2019).

Nesta fase é necessário entender como a pessoa se relaciona com o produto, quais emoções, expectativas e hábitos foram expressadas nesta interação. Assim identificamos oportunidades e necessidades latentes que muitas vezes não seriam verbalizadas ou explicitadas numa entrevista ou sessão generativa. A partir dessa interação, os pontos identificados como defeituosos ou não satisfatórios devem ser colocados em questão para serem novamente analisados e novas soluções serem propostas.

Finalizada as quatro fases de conhecimento e planejamento da atividade “*Programming Maze*” e “*Programming Loop*” da suíte GCompris, as mudanças de design foram então implementadas no jogo para apresentação, levando portanto, uma primeira versão do jogo para aplicação da quinta fase do projeto que consiste em testes com a técnica de *shadowing* (PROVALORE, 2019), observando o uso do jogo e identificando possíveis melhorias e dificuldades de compreensão dos usuários, e que foi aplicada no segundo encontro com os colaboradores.

Para a atividade de “*Programming Maze*” a primeira versão com as alterações do jogo foi apresentada, como mostra a Figura 7.



Figura 7 – Primeira versão de design da Atividade “*Programming Maze*”

Fonte: Dados da pesquisa (2019)

Ao usarem esta versão, os participantes ainda mostraram insatisfação com alguns elementos presentes na tela, como por exemplo a área laranja, que remete a um canteiro, por estar atrapalhando o entendimento da trilha e preenchendo muito o espaço, tornando confuso e dando a entender que se trata de um elemento funcional do jogo e não apenas de um elemento visual. Os elementos escolhidos para trilha e alvo tiveram aprovação de alguns participantes e a desaprovação de outros, que consideraram que o caminho por onde o personagem trilharia poderia ser composto por outros elementos no lugar de um arbusto e que o alvo deste personagem poderia ser a busca por algum alimento.

Ao observar o uso do jogo pelos participantes, pude notar que os botões padrão de navegação no GCompris não apenas confundiam o usuário como atrapalhavam a visão do mesmo em alguns níveis onde a trilha era mais extensa. Daí partiu a necessidade de, por padrão, deixar estes botões de navegação no modo oculto, esta é uma medida para

o problema dos usuários que se sentem tentados a passar para as fases seguintes sem completar o desafio atual. Assim novas alterações foram feitas e a versão seguinte foi apresentada no terceiro encontro, conferir Figura 8.



Figura 8 – Versão final do design da atividade “*Programming Maze*”

Fonte: Dados da pesquisa (2019)

Esta versão teve melhor aceitação da área de navegação do personagem principal. Os participantes concordaram entre si que o campo de navegação do personagem estava legível e que a retirada de alguns elementos visuais do jogo facilitariam para a compreensão do desafio a ser resolvido pois diminuía a chance de distração com os elementos. Foram, portanto, finalizadas as alterações de design nesta atividade de “*Programming Maze*”.

Para a atividade “*Programming Loop*” o cenário a seguir, Figura 9, foi criado com base nas sugestões propostas pelos participantes. Quando apresentado foi usada a técnica de *shadowing* para identificação de possíveis melhorias no uso da atividade.

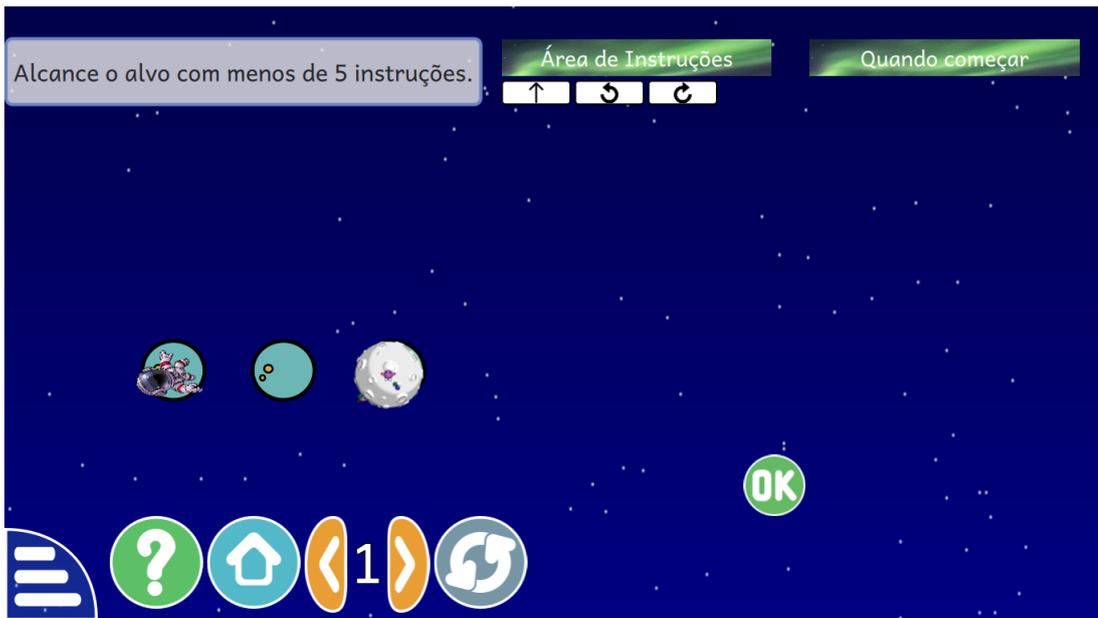


Figura 9 – Primeira Versão da Atividade “*Programming Loop*”

Fonte: Dados da pesquisa (2019)

Para esta versão do jogo foram apresentadas uma quantidade menor de sugestões de mudanças. Como a troca do ator principal de astronauta para um foguete e o questionamento do que seriam os detalhes coloridos na lua, que neste cenário representava o alvo a ser alcançado. Outra sugestão foi a mudança dos planetas para uma outra cor, pois os planetas azuis não estavam em destaque no cenário de céu estrelado, e finalizando, a sugestão de melhoria dos cabeçalhos onde localizam-se as áreas de instruções. Com base nas alterações propostas o cenário foi modificado e apresentado para avaliação como mostra a seguir na Figura 10 :



Figura 10 – Versão final do design da atividade “*Programming Loop*”

Fonte: Dados da pesquisa (2019)

Esta proposta de design para a atividade “*Programming Loop*” foi aceita pelos participantes e dada como satisfatória da perspectiva dos mesmos.

#### 4.1.6 Implementação da atividade *Programming Loop*

Para criação da atividade “*Programming Loop*”, foi usado o recurso de reuso e customização de código. Enxergando da expectativa de mecanismo de produtividade, o reuso está relacionado à porções de projetos antigos para geração de novos ao invés de escrever tudo do zero (FOX et al., 2015).

As atividades desenvolvidas para o GCompris possuem a seguinte estrutura:

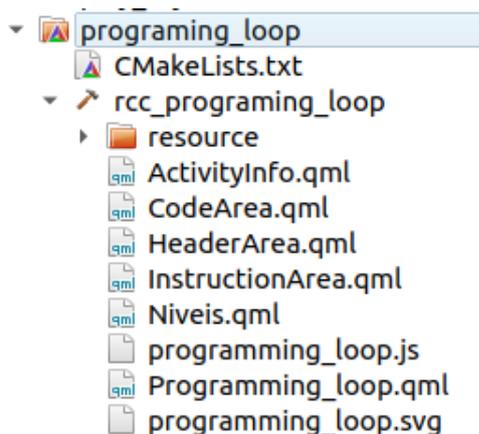


Figura 11 – Estrutura da atividade “*Programming Loop*”

Fonte: Dados da pesquisa (2019)

es

- *Resource* - contém arquivos de imagens que serão usados na atividade recorrente e os arquivos em qml que definem o comportamento de cada instrução *move-forward*, *turn-left*, *turn-right*, *loop*.
- Arquivos ‘qml’ correspondem à interface do sistema e à algumas funções específicas
  - *ActivityInfo.qml* - direitos autorais da atividade, que devem ser atualizados pelo autor após criação da mesma com descrição da atividade, definição à qual categoria pertence, instruções de ajuda.
  - *CodeArea.qml* - representação gráfica da área onde as instruções, escolhidas pelo usuário para cumprimento do desafio, serão inseridas visualmente. Função para acrescentar instruções. Função para reset dos valores da área de instruções.
  - *HeaderArea.qml* - representação gráfica do cabeçalho ao qual as instruções escolhidas pertence. Na atividade de “*Programming Maze*” é possível escolher entre inserir instruções no ‘programa principal’ ou inserir instruções no ‘procedimento’. E na atividade de “*Programming Loop*” o usuário pode inserir instrução no ‘programa principal’ ou inserir no comando ‘loop’. Função para identificar quando a ‘header area’ foi escolhida.
  - *InstructionArea.qml* - contém elementos visuais da instrução. Função de verificação de qual instrução foi selecionada. Função de verificação de onde a instrução será inserida.
  - *Niveis.qml* - Arquivo com mapeamento das trilhas de cada nível do jogo.
  - *Programming\_loop.qml* - elementos visuais da atividade (personagem, trilha, cabeçalhos, botões de navegação, instruções).

- *programming\_loop.js* - Declaração de possíveis instruções e onde encontrar o modo de comportamento de cada uma delas. Descrição dos tutoriais de uso do jogo. Funções de *start*, *stop*, iniciar nível, próximo nível, nível anterior . Funções de criação das instruções, função de destruição das instruções, função de rotação, função para executar algoritmo criado pelo usuário, função para executar próxima instrução, função para verificar se o personagem está apontando para o limite do caminho, função para identificar se alcançou o alvo, função para resetar posições do personagem alvo.

## 4.2 Utilização de Design Participativo com Crianças na Customização da Suíte GCompris

O design participativo aplicado no ambiente educacional, para produção de artefatos que serão usados por crianças, visa levar em consideração a opinião destes no processo do desenvolvimento ressaltando o que é importante para eles e colocando as suas sugestões no artefato produzido o que facilita a utilização do produto pelo seu público alvo.

Em geral, as crianças se mostram bem francas ao expressar suas opiniões quando se deparam com um ambiente que não funciona ou que não é atrativo para elas. O que nos traz a vantagem, enquanto desenvolvedores, de termos uma opinião sincera sobre o que é ou não interessante e como podemos transformar isto em melhorias para o nosso sistema. Outro fator positivo a ser ressaltado é a empatia vinda da criança, em diversas situações o grupo facilmente se colocou no lugar de usuários mais novos, o que facilitava a identificação de possíveis bloqueios no uso da atividade.

Ao observar o seguimento do processo, identificamos que há um desejo de construir algo realmente positivo no artefato gerado, e de encontrar soluções para o que está sendo proposto. A observação do grupo também colabora para a validação da interface produzida, pois facilita a identificação de elementos que não estão sendo compreendidos ou que não estão usáveis, assim como reforça a necessidade de alguns recursos, como o de tradução para linguagem do país da criança já que se trata de um público infantil. E também identificar através da observação expressões de surpresa ao descobrirem o funcionamento de algum elemento da interface e ao se depararem com o resultado do trabalho com elementos criados por eles.

Das dificuldades encontradas durante o processo é importante destacar algumas. A técnica de prototipação em papel, exigiu um pouco mais de alguns participantes, nesta fase é possível notar resistência de alguns ao expressarem suas ideias no papel em forma de desenho pela falta de afinidade, o que os deixavam pouco confortáveis com a observação do processo em si.

Um limite encontrado no processo do projeto foi o de ajudar os participantes a diferenciarem elementos de design do jogo de elementos de funcionalidades ou jogabilidade. Constantemente surgiam ideias de melhorias de jogabilidade na intenção de produzir um jogo mais atrativo e com mais recursos, o que fugia da proposta de aplicação do projeto; isso reforça que a produção de design participativo com crianças precisa ser guiada em todas as fases de modo claro, deixando-os à vontade para fazer suas considerações e guiando-os para obtenção dos elementos desejados.

Embora alguns participantes tivessem contato prévio com conceitos de programação é bastante comum a confusão de alguns conceitos que estavam sendo abordados no jogo, por exemplo, confundir a repetição da chamada do procedimento criado com o conceito de “loop” que repete um algoritmo por um número de vezes definido previamente; a compreensão de lateralidade em relação à posição do personagem; e a sugestão de comandos que fogem dos conceitos ensinados na atividade.

Na entrevista de grupo focal, observamos que a maior participação vem do grupo feminino, visto que os meninos têm uma maior facilidade em se envolver quando se trata de produzir elementos para inclusão no trabalho. O poder de percepção do que pode ser melhorado é feito por todos, porém a expressão desta melhoria é guiada pelas meninas. O uso da técnica de entrevista de grupo focal facilita a compreensão da visão dos participantes e a validação do trabalho apresentado. A entrevista foi guiada em modo de conversação e dividida para avaliação das diferentes sugestões de design que foram criadas. Trechos da entrevista de grupo focal podem ser conferidos no Apêndice C.

## 5 Considerações Finais

Neste capítulo apresentaremos as conclusões do nosso estudo, que vão ao encontro dos resultados apresentados no capítulo anterior. Em seguida apontaremos algumas limitações encontradas neste trabalho e, por fim, algumas sugestões de novas pesquisas de estudos nesta área.

### 5.1 Conclusões

Diversos sistemas de software são desenvolvidos atualmente a partir da visão dos desenvolvedores que, nem sempre, compreendem o contexto onde estão os seus usuários. Isto pode influenciar tanto no modo como os usuários finais do sistema irão interagir com ele, como na usabilidade deste sistema. Quando se trata de software desenvolvido para o público infantil a não compreensão das necessidades do usuário influenciam no uso e na percepção do mesmo sobre o software. Ao envolvermos as crianças no processo de construção do design do software observamos que as mesmas contribuem significativamente, pois, estão usando dispositivos tecnológicos sob a observação do pesquisador, que procura entender o funcionamento das tecnologias para as crianças, para que elas sejam adaptadas e que correspondam melhor às expectativas das mesmas.

A pesquisa objetivou contribuir para o GCompris com o desenvolvimento de uma atividade no nicho do ensino do pensamento computacional e com o desenvolvimento do design de uma atividade já existente da perspectiva de potenciais usuários, e auxiliar os novos colaboradores do GCompris proporcionando uma melhor visão de como o ambiente de desenvolvimento do GCompris pode ser configurado.

O trabalho com os participantes foi desenvolvido a partir destes pontos visando ressaltar a importância do sistema e aplicar as opiniões dos participantes na customização e construção do artefato proposto. A aplicação do design participativo nos artefatos desenvolvidos para o público infantil permite enxergar a necessidade do envolvimento de potenciais usuários no processo de criação, pois ao invés de desenvolvemos sistemas de software colocando a nossa visão do que é belo e usável independente do público alvo, estamos entendendo a perspectiva do usuário final e o seu posicionamento diante do software proposto, e para esta compreensão se faz necessário colocá-lo como participante do processo de criação, levando em consideração suas opiniões e questionamentos.

As técnicas usadas na aplicação deste trabalho contribuem para apresentação e envolvimento do público com o sistema proposto, para a identificação de problemas no sistema avaliado, para o encontro de soluções para estes problemas e para o entendimento

das necessidades do público e do que precisa ser corrigido e adaptado no mesmo, assim como a validação do software proposto.

## 5.2 Limitações do Trabalho

A primeira limitação está na diferença de idade, ainda que pequena, entre os participantes e o público alvo do GCompris, pois esta pode ter influenciado na construção do artefato gerado nesta pesquisa.

A segunda limitação está na dificuldade de envolvimento de alguns participantes durante a pesquisa, seja por medo ou vergonha de exporem sua opinião. Algumas técnicas incentivando o envolvimento dos participantes poderiam ter sido aplicadas, na intenção de obter melhores resultados e eliminar o desconforto dos participantes nos momentos de limitação dos mesmos.

A terceira limitação está no tempo de desenvolvimento da pesquisa, pois este influenciou diretamente no resultado final se considerarmos que a aplicação de algumas técnicas poderiam ter sido repetidas mais vezes nas etapas de construção do design, melhorando ainda alguns aspectos e elementos que foram inseridos no design das atividades; e, também, proporcionando um tempo hábil para a aprovação da proposta de design das atividades pela equipe do GCompris.

A quarta limitação está relacionada a não validação do design criado nesta pesquisa por outro grupo de potenciais usuários, fazendo um comparativo entre as versões iniciais e finais de cada atividade, e analisando este resultado para obtenção de melhorias.

## 5.3 Trabalhos Futuros

A partir dos resultados obtidos na realização deste trabalho, é possível identificar a contribuição que os participantes deram à usabilidade do sistema e vislumbrar possíveis trabalhos futuros.

Podemos aplicar as técnicas usadas neste trabalho com outro público alvo, na intenção de obter uma perspectiva diferente da obtida, e validar o uso destas técnicas para construção de design de outros sistemas de mesmo objetivo.

Avaliar a aplicação de outras técnicas para construção de design, buscando alcançar a contribuição do público alvo de sistemas de software na avaliação e validação da usabilidade dos mesmos.

Avaliar o sistema proposto com um grupo diferente de usuários na intenção de validar o que foi sugerido e modificado pelos participantes do processo apresentado neste trabalho.

## Referências

- BASAWAPATNA, A.; KOH, K. H.; REPENNING, A. Recognizing computational thinking patterns. *SIGCSE 2011* — Proceedings of SIGCSE 2011, New York: ACM, 2011.
- BOGDAN, R. C.; BIKLEN, S. K. Qualitative research for education: an introduction to theory and methods. Boston: Pearson Education Group, 2006.
- BROWN, T. Designthinking: uma metodologia ponderosa para decretar o 3m das velhas ideias. Rio de Janeiro: Elsevier, 2010.
- CAMARGO, L. S. d. A.; FAZANI, A. J. Explorando o Design Participativo como Prática de Desenvolvimento de Sistemas de Informação. *InCID: Revista de Ciência da Informação e Documentação*, Universidade de Sao Paulo Sistema Integrado de Bibliotecas - SIBiUSP, v. 5, n. 1, p. 138, mar 2014. ISSN 2178-2075.
- CAPLAN, S. Using focus group methodology for ergonomic design. *Ergonomics*, v. 33, n. 5, p. 527–33, 1990.
- COUTINHO, C. P. Metodologia de investigação em ciências sociais e humanas: Teoria e prática. *Coimbra: Almedina*, 2014.
- CSTA. *K-12 Computer Science Standards*. 2019. Disponível em: <<https://www.csteachers.org/?>> Acesso em: 01 Dez 2019.
- DENSCOMBE, M. The good research guide: for small-scale research projects. Berkshire: Open University Press, 2014.
- DIAS, C. A. Grupo focal: técnica de coleta de dados em pesquisas qualitativas. *Informação Sociedade: Estudos*, 10, 2000.
- DRUIN, A. The role of children in the design of new technology, behaviour information technology. 2002.
- FERNANDES, L. C. D. *Projeto de livro com crianças e para crianças uma abordagem do Design Participativo*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, 2018.
- FILGUEIRAS, F. *JAVA – Estruturas de Repetição*. 2015. Disponível em: <<https://tableless.com.br/java-estruturas-de-repeticao/>>. Acesso em: 10 Dez 2019.
- FORBELLONE, A. L. V. Lógica de programação – a construção de algoritmos e estruturas de dados. *São Paulo. MAKRON*, 1993.
- FOX, A. et al. *Construindo software como serviço (SaaS) : uma abordagem ágil usando computacao em nuvem*. [S.l.: s.n.], 2015. ISBN 9780984881277.
- FREDERICO. *Premios*. 2004. Disponível em: <<https://web.archive.org/web/20081011235610/http://gcompris.net/Premios>>. Acesso em: 30 nov. 2019.
- FSF. *What is free software?* 2019. Disponível em: <<https://www.gnu.org/philosophy/free-sw.en.html>>. Acesso em: 27 nov. 2019.

- GCOMPRIS. *Manual*. 2016. Disponível em: <<https://gcompris.net/wiki/Manual>>. Acesso em: 15 out. 2019.
- GCOMPRIS. *New contributor*. 2017. Disponível em: <[https://gcompris.net/wiki/New\\_contributor](https://gcompris.net/wiki/New_contributor)>. Acesso em: 7 out. 2019.
- GCOMPRIS. *Reviewing an activity*. 2018. Disponível em: <[https://gcompris.net/wiki/Reviewing\\_an\\_activity](https://gcompris.net/wiki/Reviewing_an_activity)>. Acesso em: 15 nov. 2019.
- GCOMPRIS. *Qt Quick development process*. 2019. Disponível em: <[https://gcompris.net/wiki/Qt\\_Quick\\_development\\_process](https://gcompris.net/wiki/Qt_Quick_development_process)>. Acesso em: 5 set 2019.
- GIBBS, G. Análise de dados qualitativos. tradução roberto cataldo costa. Porto Alegre: Artmed, 2009.
- GIET, T. *GCompris*. 2019. Disponível em: <[https://gcompris.net/index-pt\\_BR.html](https://gcompris.net/index-pt_BR.html)>. Acesso em: 15 nov. 2019.
- HEXSEL, R. A. Software Livre - Propostas de Ações de Governo para incentivar o uso de Software Livre. *Source*, 2002.
- HOLANDA, A. C. A.; KOH, K. H.; REPENNING, A. Possibilidades do e-learning em projetos interdisciplinares no ifac. In *Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação (WCBIE)*, p. 409–413, 2017.
- IDLER, S. *UXKids*. 2014. Disponível em: <<http://uxkids.com/blog/>>. Acesso em: 15 Dez 2019.
- INSTITUTO EDUCADIGITAL. *Design thinking para Educadores*. 2014. Disponível em: <<http://issuu.com/dtparaeducadores>>. Acesso em: 15 Dez 2019.
- JONES, C. Métodos de diseño. Barcelona: Gustavo Gilli, 1978.
- KDE. *Get Involved/development*. 2019. Disponível em: <[https://community.kde.org/Get\\_Involved/development](https://community.kde.org/Get_Involved/development)>. Acesso em: 7 out. 2019.
- KDE. *Welcome*. 2019. Disponível em: <<https://phabricator.kde.org/>>. Acesso em: 7 out. 2019.
- LOCKWOOD, T. Designthinking: Integrating innovation, customer experience, and brand value. New York: Allworth Press, 2009.
- MACK, N. e. a. Participant observation. in: *Qualitative research methods: A data collector's field guide*. North Carolina: Family Health International, 2005.
- MARTIN, R. Design thinking: achieving insights via the “knowledge funnel”. *Emerald Group Publishing Limited*, v. 38, n. 2, p. 37–41, 2010.
- MARTINS, D. G. et al. Desenvolvimento de Interfaces Digitais Infantis: Estudo Preliminar sobre Design Centrado na Criança. *Human Factors in Design*, Universidade do Estado de Santa Catarina, v. 6, n. 12, p. 46–57, dec 2017. ISSN 2316-7963.
- MELO, A. M.; BARANAUSKAS, M. C. C.; SOARES, S. C. d. M. Design com crianças: da prática a um modelo de processo. v. 16, n. 01, 2008. ISSN 1414-5685.

- MELO, M. M. M. *Novas Tecnologias na educação: reflexões sobre a prática*. Maceio - AL: EDUFAL, 2002.
- MERRIAM, S. B. Case studies as qualitative research. in: *Qualitative research and case study applications in education*. Rev. and e ed. San Francisco, CA: Jossey-Bass Publishers, p. 29–43, 1998.
- MICHAELS, P. *5 Componentes para o Design Thinking de Stanford d. Escola*. 2017. Disponível em: <<https://medium.com/@philmichaels/5-components-to-design-thinking-by-stanford-d-school-48dd111bbbe5>>. Acesso em: 13 out 2019.
- MULLER, M. J.; HASLWANTER, J. H.; DAYTON, T. *Participatory Practices in the Software Lifecycle*. [S.l.]: Elsevier, 1997.
- NERIS, V. P. A. *Estudo e proposta de um framework para o design de interfaces de usuário ajustáveis*. Tese (Doutorado) — Unicamp, 2010.
- NORMAN, D. A. *O design do dia-a-dia*. Rio de Janeiro: Rocco, 2006.
- NUNES, D. J. “ciência da computação na educação básica”. 2011.
- OLIVEIRA, A. C. A. de. A contribuição do design thinking na educação. *E-Tech*, v. 2, n. Especial Educação, p. 105–121, 2014. ISSN 1983-1838.
- PAPERT, S. *Logo: computadores e educação*. São Paulo: Brasiliense, 1985.
- PROVALORE. *As 25 técnicas e ferramentas utilizadas pelo Design Thinking*. 2019. Disponível em: <<https://www.provalore.com.br/as-25-tecnicas-e-ferramentas-utilizadas-pelo-design-thinking/>>. Acesso em: 5 nov. 2019.
- Schumacher Barcelos, T. *Projecto LATIn View project Smart Ecosystem for Learning and Inclusion-ERANet17/ICT-0076 SELI View project*. [S.l.], 2012. Disponível em: <<https://www.researchgate.net/publication/256439343>>.
- SPINUZZI, C. *The methodology of participatory design*. technical communication. 2005.
- VIANNA, M. e. a. *Designthinking: inovação em negócios*. Rio de Janeiro: MJV Press, 2012.
- WING, J. M. Computational thinking. *Communications of the ACM. Journal of Biomedical Informatics*, n. 3, p. 33–35, 2006.

# Apêndices

# APÊNDICE A – Customização de Ambiente de Desenvolvimento do GCompris

O processo para customização do ambiente para desenvolvimento no GCompris está espalhado em diferentes referências, uma das contribuições deste apêndice é justamente sintetizar essas informações e facilitar a compreensão da customização do ambiente de desenvolvimento. É importante destacar que não se trata de um tutorial e que não é objetivo desta seção explicar comandos de sistemas operacionais ou aplicações.

Para criação de uma nova atividade no GCompris é importante criar um protótipo da mesma, definindo os seus objetivos e estruturando o que será desenvolvido. Este protótipo ajudará a guiar o processo de desenvolvimento e a identificar antecipadamente as possíveis dificuldades do projeto. Depois de esclarecer a ideia, é necessário fazer a proposta da atividade descrevendo o objetivo geral e como esse objetivo vai ser alcançado, além do protótipo da atividade e enviar esta proposta ao e-mail [gcompris-devel@kde.org](mailto:gcompris-devel@kde.org), incluindo também uma *timeline* do desenvolvimento estimando o tempo de uma tarefa. Esta etapa é importante para que a equipe do GCompris possa avaliar e validar a ideia, garantindo que a mesma ainda não tenha sido proposta ou implementada anteriormente por outro colaborador, e sugerindo melhorias na mesma, ou auxiliando em possíveis dúvidas (GCOMPRIS, 2017).

É aconselhado para quem quer desenvolver uma nova atividade no GCompris, o estudo da tecnologia Qt Quick. O desenvolvimento deve ser feito, obrigatoriamente, numa distribuição GNU/Linux de sua preferência. Após a instalação da distribuição do GNU/Linux, é necessário baixar e instalar a versão mais recente do Qt, em seguida instalar o CMake.

O código do GCompris pode ser obtido no sistema de controle de versão do KDE Git ou no GitHub. O colaborador deve fazer um *fork* do projeto e criar uma *branch* para o desenvolvimento da sua nova atividade. É possível criar uma nova atividade automaticamente usando o script `createit.sh` disponível, através dos comandos:

```
» cd src/activities
» ./createit.sh 'my-new-activity'
```

Ou manualmente criando um diretório para sua atividade em `src/activities`. Nele, se cria um *ActivityInfo.qml*, *CMakeLists.txt* e seu ponto de entrada do qml *my-new-activity.qml* (GCOMPRIS, 2019).

Após a criação da atividade os direitos autorais devem ser atualizados no arquivo de

"*ActivityInfo.qml*", após a compilação do GCompris é necessário a verificação da existência da nova atividade.

Uma boa prática para o novo colaborador é observar as atividades existentes no GCompris e verificar o código delas quando tiver um problema. Quando implementadas, as atividades devem ter uma boa aparência e serem utilizáveis em diferentes dispositivos: telefones, tablets e desktops. O teste deve ser feito para os três dispositivos pois a resolução e o valor de dpi podem diferir bastante.

O formato 'svg' é preferível para as imagens e ícones das atividades, caso as imagens não estejam disponíveis no [openclipart.org](http://openclipart.org) usamos um script, em python, disponível no site, para conversão de imagens 'png' para 'svg' (GCOMPRIS, 2018).

As atividades do GCompris também usam uma fonte padrão, definido por GCText para garantir a homogeneidade do texto exibido no aplicativo, como no exemplo do código abaixo:

#### Código A.1 – Código do padrão GCText

```
1 GCText {
2   id: text
3   text: Text
4   fontSize: 14
5   font.weight: Font.DemiBold
6   color: white
7 }
```

Fonte: (GCOMPRIS, 2018)

As atividades devem adaptar seu conteúdo adequadamente quando a janela é redimensionada e devem suportar a rotação da tela.

No arquivo QML apenas pequenos elementos de JavaScript devem ser mantidos, toda a lógica do jogo deve estar em seu arquivo JavaScript, facilitando para outros colaboradores a leitura e entendimento da lógica da atividade. Os arquivos QML devem ser vistos como a descrição da interface gráfica e o JavaScript como a lógica do jogo.

Para facilitar suporte às ferramentas de tradução das atividades do GCompris, cada vez que aparece uma *String* a mesma deve conter `qsTr()`:

#### Código A.2 – Código para String

```
1 GCText {
2   text: ApplicationInfo.isMobile ?
3     qsTr("Tap both hands at the same time, " +
4       "to make the ball go in a straight line.") :
```

```
5     qstr("Press left and right arrow keys at the same time, " +  
6         "to make the ball go in a straight line.")  
7 }
```

Fonte: (GCOMPRIS, 2018)

Para melhor posicionamento dos itens visuais do seu projeto é necessário definir relacionamento entre estes usando *anchors*, isto facilita o processo de organização de diferentes itens na tela. Como no exemplo abaixo:

### Código A.3 – Código para uso de anchors

```
1 GCTest {  
2     id: instructions  
3     anchors {  
4         left: parent.left  
5         right: parent.right  
6     }  
7 }
```

Fonte: (GCOMPRIS, 2018)

Quando a atividade estiver pronta para ser submetida para revisão é necessário fazer uma refatoração do código, garantindo assim que o *commit* não contenha variáveis não usadas ou arquivos temporários, e que todas as mensagens de advertências de compilação estejam resolvidas.

Para compilar o GCompris para Ubuntu você precisa ter um sistema operacional ubuntu instalado e o *ubuntu sdk* (GCOMPRIS, 2018). Em seguida clone o código fonte em uma pasta GCompris-qt. Logo após você deve criar o ambiente *chroot* e fazer login nele.

```
»sudo click chroot -a armhf -f ubuntu-sdk-14.10 create  
»sudo click chroot -a armhf -f ubuntu-sdk-14.10 run
```

Dentro do *chroot* você precisa instalar manualmente o módulo *QTQuickControls* e executar o comando de compilação:

```
»apt-get install qml-module-qtquick-controls:armhf  
»cmake -DWITH_UBUNTU=on -DUBUNTU_CLICK=1 -DQt5_DIR=/usr/lib/arm-  
linux-gnueabi/hf/cmake/Qt5/ ../GCompris-qt/ && export QT_SELECT=qt5-arm-linux-  
gnueabi/hf && make && make DESTDIR=/tmp/click/ install
```

E então, fora do *chroot*, você precisa executar o comando *click* para criar o pacote:

```
»click build /tmp/click/
```

Por padrão, nenhuma tradução está presente no repositório *git*. É possível obtê-los no KDE *svn* <sup>1</sup> e acompanhar o status da migração no portal do GCompris.

<sup>1</sup> <https://websvn.kde.org/trunk/110n-kf5/>

O processo de compilar o aplicativo para outras plataformas está também disponível no site do GCompris na área de processo de desenvolvimento (GCOMPRIS, 2018)

# APÊNDICE B – Descrição das Atividades de Programação

Esta seção visa esclarecer o fluxo das atividades “*Programming Maze*” e “*Programming Loop*” para melhor compreensão do trabalho.

## B.1 A atividade “*Programming Maze*”

A atividade é dividida em duas partes, nas cinco primeiras fases o usuário exercita o conceito de sequência de algoritmos, abordado pelo ensino do pensamento computacional para crianças como o conceito chave para início do aprendizado. A partir da sexta fase o conceito de procedimento é abordado, o usuário deve criar o procedimento e fazer a chamada deste no programa principal usando a instrução “CALL P”. Entre estas fases temos tutoriais de uso que orientam o funcionamento da atividade.

Para acompanhar o processo de execução do algoritmo criado pelo usuário, a interação da atividade foi programada graficamente para que a instrução executada naquele momento esteja em destaque, auxiliando assim a identificação de possíveis erros no algoritmo criado por este. Também pensando no melhor aproveitamento das fases e na intenção de instigar os usuários a criarem um algoritmo correto, esta atividade conta com um limite de instruções que podem ser usadas por fase. Quando o usuário atinge o objetivo da fase este recebe uma mensagem positiva indicando o bom desempenho e em seguida a fase seguinte é aberta, o contrário acontece quando o mesmo não atinge o objetivo, uma mensagem negativa é mostrada e a fase atual continua na tela com as mesmas instruções escolhidas anteriormente para que este faça as correções a partir de onde parou.

## B.2 A atividade “*Programming Loop*”

Similar a atividade descrita anteriormente, esta atividade também é dividida em duas partes, onde nas cinco primeiras fases o conceito de sequência de algoritmos são treinados novamente, e a partir da sexta fase o usuário treina o conceito de laços de repetição, usando o comando de loop e definindo a quantidade de vezes que o mesmo será executado. O processo de execução do algoritmo se dá do mesmo modo que na atividade descrita anteriormente, destacando a instrução em execução naquele momento. Esta atividade também conta com o recurso de limite de instruções que podem ser usadas por fase, e com a identificação de objetivo atingido ou objetivo não atingido.

A seguir a versão original de ambas atividades apresentada aos participantes desta pesquisa.



Figura 12 – Versão original das atividades

Fonte: GCompris (2019)

## APÊNDICE C – Entrevista de Grupo Focal

A seguir trechos da entrevista de grupo focal realizada neste trabalho. Os nomes atribuídos à cada afirmação abaixo são fictícios para preservar a identidade dos participantes.

Pergunta	Resposta
Qual impressão vocês tiveram quando viram o jogo pela primeira vez?	<ul style="list-style-type: none"> <li>• “Gostei da ideia {...} mas quando vi o jogo era difícil compreender o que tinha ali, e realmente precisava de uma organização do que tinha ali. (Helena)</li> <li>• “As cores não tem um padrão, elas não combinam” (Alice)</li> <li>• “A escolha do personagem poderia ser melhor, quando eu vi não entendi muito bem o que era, era meio estranho, poderiam ter colocado um bonequinho olhando de outra posição” ... “O cenário era estranho, tinha que pensar melhor nas coisas (elementos) que estavam ali” (Laura)</li> </ul>
Continuação na próxima página	

Pergunta	Resposta
O que vocês tem a dizer sobre os elementos que foram escolhidos para o jogo e como eles foram posicionados?	<ul style="list-style-type: none"> <li>• Quando vi pensei: qual a lógica de um pinguim no gelo e uma grama ao redor? (Manuela)</li> <li>• Esses botões (botões de navegação) atrapalham de ver o jogo, podia ter uma pastinha que armazenassem esse botões e eles só aparecerem quando clicar. (Alice)</li> <li>• Eles poderiam passar de fase sem ser nas setinhas (botões de navegação), poderiam completar a fase e passar automaticamente (Arthur)</li> <li>• O local onde o jogo ta ocorrendo é só da tela, podia ocupar mais espaço pra chamar mais atenção (Manuela)</li> </ul>
Quais dificuldades vocês encontraram para usar o jogo?	<ul style="list-style-type: none"> <li>• Não tem tradução aí, tá tudo em inglês e quem não entende quase nada de inglês tem dificuldade. (Laura)</li> <li>• Como é para crianças tinha que ter um 'tradutorzinho' do lado para português para ajudar (Isabela)</li> </ul>
Continuação na próxima página	

Pergunta	Resposta
Quais dificuldades vocês encontraram para usar o jogo?	<ul style="list-style-type: none"> <li>• Não tem tradução aí, tá tudo em inglês e quem não entende quase nada de inglês tem dificuldade. (Laura)</li> <li>• Como é para crianças tinha que ter um ‘tradutorzinho’ do lado para português para ajudar (Isabela)</li> <li>• Quando a gente foi jogar no primeiro dia, a gente confundia os lados de direita e esquerda, . . . , a pessoa tinha que se colocar no ângulo que o pinguim tava pra poder saber o lado de virar” (Manuela)</li> <li>• O modo como mostrava a execução das instruções tava desorganizado quando era do lado um do outro, . . . , se fosse um embaixo do outro era mais fácil de entender” (Helena)</li> <li>• Poderia ter umas setinhas indicando como é que se joga, para as primeiras fases e depois a criança iria fazendo sozinha” (Manuela)</li> <li>• Se tivesse um passo a passo a criança teria maior compreensão sobre o que ela tem que fazer no jogo” (Danilo)</li> </ul>

Fonte: Dados da pesquisa (2019)

Quadro C.1: Percepção dos participantes - Atividade original

Pergunta	Resposta
O que melhorou da versão anterior para esta versão?	<ul style="list-style-type: none"> <li>• A posição das instruções que dá pra enxergar direito o caminho do personagem. (Arthur)</li> <li>• Ficou claro a ordem de executar as setas, uma abaixo da outra é muito melhor. (Helena)</li> <li>• A tradução para português. (Erick)</li> </ul>
O que pode ser melhorado nesta versão?	<ul style="list-style-type: none"> <li>• Esses canteiros laranja atrapalha de ver o que é importante. (Alice)</li> <li>• O nome das instruções para saber o que é direita e o que é esquerda. (Helena)</li> <li>• Os arbustos não ficaram bonitos (Alice)</li> </ul>

Fonte: Dados da pesquisa (2019)

Quadro C.2: Percepção dos participantes - Atividade “*Programming Maze*” primeira versão

Pergunta	Resposta
Quais são as considerações finais?	<ul style="list-style-type: none"> <li>• Melhorou a página das instruções sem aquele fundo marrom” (Arthur)</li> <li>• Eu acho que está melhor, porque não tem aquele monte de coisa (botões de navegação) aqui, a tela (espaço de movimento do personagem) está maior, dá pra saber melhor o que tem que se fazer. (Erick)</li> <li>• O milho e a grama deixaram mais organizado do que quando era a lama e os arbustos, tá mais limpo. (Alice)</li> <li>• Quando preenche com mais blocos ficou bom, porque dá um certo destaque nos campos que tem que percorrer, se aumentar esses blocos atrapalha” (Arthur)</li> <li>• Mudaria essa posição do botão de ‘ok’, fica solto nas fases que não tem o ‘procedimento’ (Helena)</li> </ul>

Fonte: Dados da pesquisa (2019)

Quadro C.3: Percepção dos participantes - Atividade “*Programming Maze*” versão final

Pergunta	Resposta
O que pode ser melhorado nesta versão?	<ul style="list-style-type: none"> <li>• Esse astronauta não ficou bom não, . . . , parece q ele vai cair, poderia ser um foguete. (Erick)</li> <li>• Eu não gostei dessa lua, ela ficou confusa com esses coloridinhos. (Alice)</li> <li>• Podia ter mais cor, porque só essas bolinhas azuis, ... 'ta' muito sem graça. (Manuela)</li> <li>• Poderia ser uns planetas mais coloridinhos. (Arthur)</li> <li>• As cores nessa parte da área de instruções estão estranhas. (Helena)</li> </ul>

Fonte: Dados da pesquisa (2019)

Quadro C.4: Percepção dos participantes - Atividade “*Programming Loop*” primeira versão

Pergunta	Resposta
Quais são as considerações finais?	<ul style="list-style-type: none"> <li>• Antes esses planetas eram azuis, aí se confundiam com o fundo, agora tá destacando. (Erick)</li> <li>• O foguete ficou melhor porque dá a ideia que viaja entre os planetas até a lua. (Danilo)</li> <li>• A tela de instruções tá mais suave, tá combinando com o restante do cenário. (Manuela)</li> <li>• Os objetos estão distribuídos de um jeito legal (Arthur)</li> </ul>

Fonte: Dados da pesquisa (2019)

Quadro C.5: Percepção dos participantes - Atividade “*Programming Loop*” versão final