

UNIVERSIDADE FUMEC
Faculdade de Ciências Empresariais
Mestrado em Sistemas de Informação e Gestão do Conhecimento

**AMBIENTE DE PROGRAMAÇÃO SCRATCH
NA PERSPECTIVA DA ENGENHARIA SEMIÓTICA**

Rodrigo de Matos Vargas

Belo Horizonte
2016

Rodrigo de Matos Vargas

**AMBIENTE DE PROGRAMAÇÃO *SCRATCH*
NA PERSPECTIVA DA ENGENHARIA SEMIÓTICA**

Dissertação apresentada ao Curso de Mestrado em Gestão do conhecimento e Sistemas de Informação da Universidade FUMEC – Fundação Mineira de Educação e Cultura –, na área de Gestão de Sistemas de Informação e de Conhecimento concentração, na linha de pesquisa Sistemas de Informação, como requisito parcial para a obtenção do título de Mestre em Sistemas de Informação e Gestão do Conhecimento.

Orientador: Prof. Dr. Rodrigo Fonseca e Rodrigues

Belo Horizonte
2016

Dados Internacionais de Catalogação na Publicação (CIP)

V297a Vargas, Rodrigo de Matos, 1974-
Ambiente de programação *scratch* na perspectiva da
engenharia semiótica / Rodrigo de Matos Vargas. - Belo
Horizonte, 2016.
78 f. : il.

Orientador: Rodrigo Fonseca e Rodrigues
Dissertação (Mestrado em Sistemas de Informação e
Gestão do Conhecimento), Universidade FUMEC, Faculdade de
Ciências Empresariais, Belo Horizonte, 2016.

1. Ensino auxiliado por computador. 2. Scratch
(Linguagem de programação de computador). 3. Semiótica. I.
Título. II. Rodrigues, Rodrigo Fonseca e. III. Universidade
FUMEC, Faculdade de Ciências Empresariais.

CDU: 37:681.3



Universidade FUMEC
Faculdade de Ciências Empresariais
Curso de Mestrado em Sistemas de Informação e
Gestão do Conhecimento
FACE-FUMEC

Dissertação intitulada “Ambiente de Programação Scratch Voltada para Crianças sob a Perspectiva da Engenharia Semiótica”, de autoria do mestrando Rodrigo de Matos Vargas, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Rodrigo Fonseca e Rodrigues
Universidade FUMEC
(Orientador)

Prof. Dr. Luiz Cláudio Maia
Universidade FUMEC

Profa. Dra. Adriana Maria Tonini
Universidade Federal de Ouro Preto

Belo Horizonte, 26 de Fevereiro de 2016.

AGRADECIMENTOS

Agradeço, em primeiro lugar, a Deus, por permitir que mais esta etapa de minha vida seja vencida.

À minha esposa Grazielle Bellis e aos meus filhos Rodrigo Bellis e Rafael Bellis, a bênção divina de minha vida.

Aos meus pais, amigos e familiares que contribuíram e me encorajaram para a conclusão desta dissertação.

Não posso deixar de destacar e de agradecer a enorme ajuda de meus valiosos amigos, os quais contribuíram diretamente com este estudo de caso, realizando os testes de comunicabilidade propostos.

Agradeço especialmente ao professor orientador Dr. Rodrigo Fonseca e Rodrigues, o qual dedicou inúmeras horas em suas revisões e em dicas de reestruturação e enriquecimento de conteúdo.

Aos professores da Banca Qualificadora, que contribuíram em suas argumentações e olhares críticos sobre a dissertação apresentada.

Por fim, agradeço a todos os amigos e companheiros de trabalho, que se desdobraram em minha ausência e, com muita competência e comprometimento, superaram esse desafio.

Muito obrigado a todos!

RESUMO

Neste atual cenário mundial, em que as mudanças acontecem em uma velocidade espantosa, novos desafios, novas tecnologias e novos atores aparecem e participam ativamente dessas mudanças. Esta pesquisa apresenta exatamente um desses novos atores, as crianças, as quais interagem cada vez mais cedo com essa diversidade tecnológica. A participação das crianças nesse cenário passa a ser mais ativa quando as mesmas começam a aprender linguagem de programação através de ambientes computacionais. Esses ambientes são avaliados através de sua metacomunicação com aplicação de métodos da Engenharia Semiótica. Assim, a pesquisa visa observar e compreender o processo de metacomunicação usuário-sistema e a contribuição do mesmo no processo de aprendizagem infantil. Com o objetivo de verificarmos essas ocorrências de rupturas entre o ambiente de programação Scratch e seu público-alvo específico, aplicamos a metodologia denominada MAC – Método de Avaliação da Comunicabilidade –, uma técnica conhecida pela Engenharia Semiótica desde 1999. Diversos resultados foram percebidos, porém um merece destaque especial, em que percebemos que as mensagens foram, em sua maioria, identificadas e interpretadas pelos usuários finais. Por outro lado, ocorreram rupturas de comunicação em todos os voluntários, mesmo para aqueles que concluíram todas as tarefas.

Palavras-chave: Engenharia Semiótica. Scratch. Ambientes de Programação. Aprendizado. Metacomunicação. Método de Avaliação da Comunicabilidade.

ABSTRACT

In this present world scenario, where changes happen in an amazing speed, new challenges, new technologies and new actors appear and actively participate in these changes. This research presents just such new actors, the children, which they interact increasingly early with this technological diversity. Children's participation in this scenario becomes more active when they begin to learn programming language through computational environments. These environments are assessed through its application to meta methods of Semiotics Engineering. Thus, the search attempts to understand and evaluate the process of meta-user system and its contribution in the early learning process. In order verify these occurrences breaks between the Scratch programming environment and your audience specific target. We apply a methodology called MAC - the communicability evaluation method, a technique known by the semiotic engineering since 1999. Several outcomes were perceived, but one deserves special mention where we realized that the messages were mostly identified and interpreted by end users, but communication breaks occurred in all volunteers, even for those who have completed all tasks.

Keywords: Semiotic Engineering. Scratch. Programming Environments. Learning. Metacommunication. Evaluation Method of Communicability.

LISTA DE FIGURAS

Figura 1 – Componentes dos Sistemas de Informação.....	14
Figura 2 – Interface Kodu com os elementos e ações – em blocos	18
Figura 3 – Interface Scratch com elementos e ações – em blocos	19
Figura 4 – Espiral do conhecimento	22
Figura 5 – Interface inicial plataforma Scratch	23
Figura 6 – As três entidades do signo	28
Figura 7 – Linha do tempo: evolução da Engenharia Semiótica	31
Figura 8 – Processo de comunicação da Engenharia Semiótica	33
Figura 9 – Relação triádica do signo	Erro! Indicador não definido.
Figura 10 – Interface da primeira fase do Scratch.....	39
Figura 11 – <i>Workflow</i> do Método MAC.....	44

LISTA DE QUADROS

Quadro 1 – Recursos aplicados à conversão do conhecimento.....	25
Quadro 2 – Classificação do sistema por tipo de falha	35
Quadro 3 – Etiquetas MAC.....	36
Quadro 4 – Grupos da expressão de comunicabilidade.....	37
Quadro 5 – Signos icônicos.....	40
Quadro 6 – Signos índices	40
Quadro 7 – Signos símbolos	40
Quadro 8 – Andamento das Tarefas pelos Usuários.....	50
Quadro 9 – Perfil Tecnológico dos usuários.....	51

LISTA DE ABREVIATURAS E SIGLAS

IHC	Interação homem-computador
MAC	Método de avaliação de comunicabilidade
MIS	Método de inspeção semiótica
MIT	<i>Massachusetts Institute Technology</i>
PC	<i>Personal Computer</i>

SUMÁRIO

1 INTRODUÇÃO	9
2. REFERENCIAL TEÓRICO	12
2.1 Sistemas de Informação e <i>Frames</i> Tecnológicos	12
2.1.1 <i>Componentes de Sistemas de Informação</i>	12
2.1.2 <i>Frames Tecnológicos</i>	14
2.2. Tecnologias e pessoas	16
2.2.1 <i>Linguagem de programação</i>	16
2.2.2 <i>Ambiente de programação Scratch</i>	17
2.2.3 <i>Novos Usuários</i>	20
2.3 Informação e teoria do conhecimento	21
2.3.1 <i>Espiral do conhecimento</i>	21
2.3.2 <i>Aprendizagem e conhecimento</i>	25
2.4 Semiótica peirceana e a Engenharia Semiótica	26
2.4.1 <i>Semiótica: ícone, índice e símbolo</i>	26
2.4.2 <i>A semiótica peirceana e as três fases da experiência: primeiridade, secundidade, terceiridade</i>	29
2.4.3 <i>A semiótica aplicada a ambientes digitais</i>	31
2.4.4 <i>Comunicação na Engenharia Semiótica</i>	33
2.4.5 <i>Método de Avaliação de Comunicabilidade – MAC</i>	34
3. SEMIÓTICA APLICADA AO SCRATCH	37
3.1 Ícone, índice e símbolo aplicados à plataforma Scratch	37
3.2 As três fases da experiência aplicadas aos usuários	41
4 METODOLOGIA DA PESQUISA	42

5 ANÁLISE DOS RESULTADOS	46
5.1 Aplicação do Método de Avaliação da Comunicação (MAC)	47
5.2 Análise dos dados: ambiente Scratch de programação	48
6 CONCLUSÃO	58
REFERÊNCIAS.....	61
APÊNDICE A – DOCUMENTOS E TERMOS UTILIZADOS PARA OS TESTES	65
APÊNDICE B – RESUMO DO PERFIL DOS VOLUNTÁRIOS.....	68
ANEXO A – TELAS DE ACESSO À PLATAFORMA.....	69

1 INTRODUÇÃO

Em fevereiro de 2013, nomes expressivos do mundo tecnológico, em especial Bill Gates, fundador da Microsoft, e Mark Zuckerberg, fundador do Facebook, se juntaram a uma campanha denominada *code.org*¹, a qual visa estimular o ensino de linguagens de programação nas escolas fundamentais. Diversas escolas, principalmente as particulares, já estão disponibilizando para seus alunos a experiência única de aprender os princípios de Ciência da Computação, em especial a programação de computadores, ofertando disciplinas optativas. Algumas escolas chegam a oferecê-las como disciplinas obrigatórias, sendo descrita em sua matriz curricular.

Inúmeros fatores motivam educadores, pedagogos e outros profissionais a adotarem a programação como ferramenta de estudos, sendo citados, principalmente, o desenvolvimento de habilidades dos alunos em raciocínios lógico, matemático, linguístico e sua relação causa-efeito, visivelmente presente nessa disciplina. Independentemente do motivo apresentado, ou seja, pela carência de profissionais, visão claramente defendida pelo movimento *code.org* ou pela visão do desenvolvimento intelectual e lógica do indivíduo, somente esses dois fatores já motivam a aplicabilidade da disciplina. Em resumo, independentemente do estudante, seguir carreira em Tecnologia da Informação ou não, os ensinamentos propostos por essa disciplina garantem uma dose extra de sucesso tanto em tarefas do dia a dia quanto no apoio às demais profissões que venham a ser escolhidas por ele.

Para cativar e exercer maior atração aos alunos, o projeto *code.org*, assim como outras empresas, em iniciativas isoladas, desenvolveram diversas plataformas, as quais possuem em comum a utilização de símbolos, ícones (blocos) e um ambiente mais lúdico de aprendizagem, pretendendo, assim, desmistificar a ideia de que programação de computadores é extremamente difícil e não pode ser aprendida por crianças. Na confluência da relação dessas crianças com os signos em contexto de interface *on-line* e suas interações constrói-se um ambiente lúdico de

¹ Uma nova organização sem fins lucrativos chamada *code.org* foi criada com o objetivo de difundir o ensino de programação, com o apoio de grandes instituições como a Microsoft e o Facebook.

aprendizagem. Isso se dá pela investigação de como a interseção entre Linguagem de Programação e Engenharia Semiótica podem auxiliar na concepção de processos alternativos de aprendizagem para o universo da experiência infantil.

Sob a ótica da Engenharia Semiótica, podemos entender a comunicabilidade como uma metamsagem transmitida entre a aplicação e o usuário final. Para avaliar tal comunicabilidade, empregaremos o uso do Método de Avaliação da Comunicabilidade (MAC). Esse método identifica falhas existentes na comunicação entre o que foi planejado e o que realmente foi percebido pelo usuário final.

A geração do perfil semiótico representa a última etapa do Método de Avaliação da Comunicabilidade (MAC), que consiste em um levantamento sobre a metacomunicação entre a aplicação e o usuário. O perfil semiótico baseia-se na avaliação dos resultados obtidos com a etiquetagem e a interpretação do usuário final, tendo como base a observação dessas evidências através do comportamento do usuário. Segundo De Souza e Leitão (2009), o avaliador trabalha com o material etiquetado, buscando identificar os problemas de metacomunicação.

Com ênfase em nossa linha de pesquisa em Sistemas de Informação, a perspectiva de uma análise de interpretação da tecnologia por um indivíduo ou por um grupo de pessoas caracteriza a aplicação de sistemas de informação e seus pilares de sustentação, sendo embasada teoricamente nos conceitos de *frames tecnológicos*, explicitadas no capítulo 2 que apresenta a fundamentação teórica de nossa pesquisa.

Diversas escolas, principalmente da rede privada, já começam a introduzir a linguagem de programação em suas grades curriculares. Porém, por se tratar de uma iniciativa ainda embrionária, várias questões ainda são de difíceis respostas, em especial, a seguinte: “Pode-se alcançar maior eficiência na aprendizagem infantil através da linguagem de programação apoiando-se no conceito de ‘comunicabilidade’ da Engenharia Semiótica? ”. Assim, essa é a questão de pesquisa dessa dissertação.

Desse modo, o objetivo geral desta dissertação visa observar e compreender o processo de aprendizagem infantil de linguagem de programação a partir dos pressupostos conceituais da Engenharia Semiótica aplicados sobre o ambiente de

programação Scratch, em especial identificar as rupturas de comunicação usuário-sistema.

Para atingir objetivo geral definido, os demais objetivos específicos visam ser alcançados, sendo eles:

- Identificar e apresentar o perfil semiótico dos usuários do ambiente de programação – Scratch (Desenvolvida pelo MIT ²).
- Aplicar um teste de interface em um grupo focal definido, utilizando-se do método de avaliação da comunicabilidade, denominado MAC.
- Apresentar, através dos testes aplicados, as falhas de comunicabilidade, se existentes do ponto de vista do público-alvo por meio do levantamento do perfil semiótico do mesmo.

Inúmeros fatores motivam educadores, pedagogos e outros profissionais a adotarem a programação como ferramenta de estudos, sendo citados principalmente o desenvolvimento de habilidades dos alunos em raciocínios lógico, matemático, linguístico e sua relação causa-efeito, visivelmente presente nessa disciplina. Enfim, diversas justificativas podem ser salientadas, mas apenas o benefício de aprendizagem citado acima, justifica a importância da capacitação de pessoas em futuros processos decisórios, condicionais e tantos outros, cada vez mais presente em nossas vidas.

O presente documento está organizado da seguinte forma: O capítulo 1 apresenta a relevância do tema, a questão de pesquisa, os objetivos e a justificativa da dissertação. O capítulo 2 apresenta o referencial teórico que orienta nosso trabalho, abordando os conceitos da Engenharia Semiótica, a qual fundamenta a proposta de comunicabilidade entre aplicação e usuário e os outros tópicos teóricos do trabalho. O capítulo 3 detalha a aplicabilidade de conceitos da semiótica ao ambiente de programação Scratch. O capítulo 4 detalha a metodologia utilizada e aplicada sobre o grupo focal específico. O capítulo 5 apresenta os resultados obtidos com base nos critérios e na metodologia proposta pelo Método de Avaliação da Comunicabilidade. Por fim apresenta as considerações finais e sua correlação com

² MIT – *Massachusetts Institute of Technology* – O Instituto de Tecnologia de Massachusetts é um centro universitário de educação e pesquisa privado localizado em Cambridge, Massachusetts, nos Estados Unidos.

os objetivos do trabalho, além de sugerir trabalhos futuros. Finalmente, são apresentadas as referências, que listam toda a bibliografia utilizada para a elaboração desta dissertação de mestrado, e, por fim, os apêndices, com o termo de autorização de participação na pesquisa, os questionários pré e pós-teste e o perfil dos voluntários desta pesquisa, e o anexo, com as telas do sistema.

2. REFERENCIAL TEÓRICO

Neste capítulo, concentra-se o referencial teórico, o qual percorrerá: 1) os conceitos dos sistemas de informação e seus componentes, 2) os usuários e as tecnologias como elementos dos sistemas de informação, 3) os pilares da informação e sua relação com a teoria do conhecimento, 4) os pressupostos da Engenharia Semiótica e, por fim, 5) a metodologia de comunicação utilizada para avaliação das rupturas de comunicabilidade do ambiente estudado.

2.1 Sistemas de Informação e *Frames* Tecnológicos

Neste tópico, vamos abordar os conceitos primários dos elementos e os componentes de um sistema de informação, assim como outros assuntos relevantes sobre o tema, visando o aprofundamento sobre os pilares teóricos nas seções posteriores.

2.1.1 Componentes de Sistemas de Informação

Abordar os conceitos de sistemas de informação e seus elementos é a base para o entendimento das organizações, sistemas e suas interações. Por uma visão macro, podemos fazer uma analogia entre sistemas de informação e os sistemas tradicionais que conhecemos, ou seja, sistemas podem ser vistos como um conjunto de partes que agem em conjunto, visando atingir um propósito comum.

Reforçando a explanação, Para O'Brien (2002), Sistema é um grupo de componentes inter-relacionados que trabalham juntos rumo a uma meta comum,

recebendo insumos e produzindo resultados em um processo organizado de transformação. Outros conceitos de sistemas de informação podem ser citados por diversos autores. Para Laudon e Laudon (1999), temos, como sistemas de informação:

Um conjunto de componentes inter-relacionados trabalhando juntos para coletar, processar, armazenar e distribuir informação com a finalidade de facilitar o planejamento, o controle, a coordenação, a análise e o processo decisório em empresas e organizações. (LAUDON; LAUDON, 1999, p. 4)

Fundamentado nesses autores e em outros estudiosos, uma afirmação é unânime: sistemas podem ser vistos como a junção de partes integradas visando trabalhar em conjunto. Contudo, quais são essas partes integradas nos sistemas de informação?

Sob a ótica geral do sistema de informação, apresentamos a figura 1, a tríade Pessoas, Tecnologia e Informação pode ser citada como elemento dos sistemas de informação, ou seja, as pessoas podem ser vistas como todos e quaisquer recursos humanos que interagem com nosso sistema, sejam elas *designers*, desenvolvedores, usuários finais, etc. Como segundo elemento, temos a tecnologia, a qual se refere a toda estrutura tecnológica de um sistema de informação, ou seja, software³ e hardware⁴ dos sistemas de informação. Como terceiro elemento, mencionamos a informação que pode ser vista como um processo que confere sentido às coisas, aos processamentos dos dados, propriamente ditos.

³ Software é uma sequência de instruções escritas para serem interpretadas por um computador com o objetivo de executar tarefas específicas. Também pode ser definido como os programas que comandam o funcionamento de um computador.

⁴ Hardware é a parte física de um computador, é formado pelos componentes eletrônicos, como, por exemplo, circuitos de fios e luz, placas, utensílios, correntes e qualquer outro material em estado físico, que seja necessário para fazer que computador funcione.

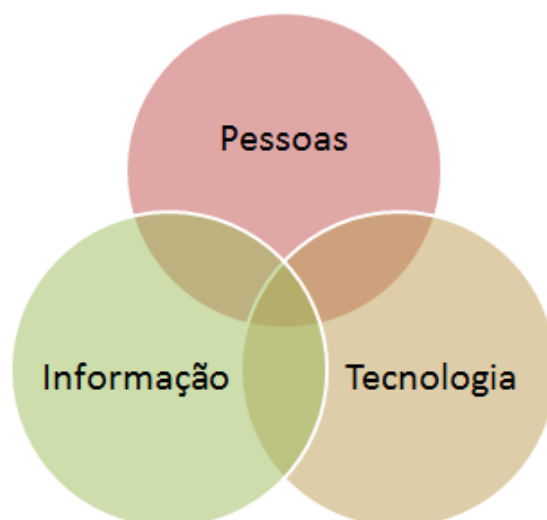


Figura 1 – Componentes dos Sistemas de Informação

Fonte: Elaborado pelo autor

Sabendo que nossa pesquisa envolve cada vertente de forma bem definida, temos: as pessoas, em nosso estudo limitada às crianças; a tecnologia que, de acordo com nossa pesquisa, é o ambiente de programação *Scratch*; e, por fim, a informação, na qual temos a comunicação entre aplicação-usuário, envolvendo os conceitos de metacomunicação do processo de aprendizagem. A partir desse momento, já conhecemos as partes integrantes de nosso sistema de informação e aprofundaremos nas próximas seções em cada um desses componentes, mas, antes disso, podemos conhecer o princípio denominado *Frames Tecnológicos*, apresentado a seguir.

2.1.2 *Frames Tecnológicos*

A partir dos conceitos e pilares apresentados anteriormente, é importante interligar alguns conceitos elementares dos sistemas de informação e educação sobre a visão do processo de aprendizagem infantil em ambientes de programação. Um desses conceitos que merece nossa atenção é o conceito de *frames tecnológicos*, apresentado em 1994 por Orlikowski e Gash. *Frames tecnológicos* influenciam como os atores pensam e agem em relação à tecnologia e, portanto, o estudo de tal assunto permite compreender as interpretações sociais da tecnologia, explicar ações e aprender significados que não seriam facilmente obtidos com outras

lentes teóricas. *Frames* tecnológicos estão diretamente ligados aos estudos cognitivos e a uma abordagem sociológica denominada Construção Social da Tecnologia. Para Walsh (1995), cognição em organizações é uma área de pesquisa que tem merecido bastante atenção. Esses estudos consideram como os indivíduos interpretam seu ambiente, mais especificamente sua cognição. Conceitos como esquemas, modelos, paradigmas e *frames* representam como os indivíduos interpretam informações de um ambiente e dão significados a elas. Assim, o conceito de *frames* tecnológicos diz respeito aos pressupostos, expectativas, conceitos e conhecimento que os usuários utilizam para compreender a tecnologia nas organizações.

Uma análise de *frames* tecnológicos pode ser vista como uma análise da interpretação da tecnologia por um indivíduo ou por um grupo de indivíduos. Nosso trabalho se enquadra perfeitamente com esse cenário apresentado, alinhando os sistemas de informação e a interpretação dos usuários. Essa interpretação da tecnologia pelos indivíduos, em nosso caso, as crianças, enquadra nosso trabalho com esse pilar dos sistemas de informação. Segundo Lin e Silva (2005), o caráter coletivo de estruturas cognitivas disseminadas entre grupos sociais para observar, interpretar, atribuir significado e utilizar a tecnologia a seu favor. Outro aspecto comum aos grupos quanto à utilização do sistema na prática é a forma como aprenderam e aprendem a operá-lo: iniciativas autônomas de aquisição de habilidades e conhecimentos sobre a tecnologia através de seu próprio uso, da ajuda de terceiros ou através de um profissional em sistemas de informação. Destacam-se, em geral, o autoaprendizado ligado diretamente à cognição e à semiose. Segundo Doherty, Coombs e Loan-Clarke (2006), as características de um sistema de informação não apenas influenciam como um usuário irá interpretar seus símbolos mas a própria funcionalidade do sistema.

Por fim, podemos entender que abordagem de *frames* tecnológicos enquadra um conjunto de sistemas de informação, principalmente aqueles com forte cognição. Ambientes de aprendizados das linguagens de programação voltadas para crianças envolvem os principais pilares da Engenharia Semiótica e dos *frames* tecnológicos, visando, assim, identificar, analisar e entender como os usuários estão interpretando os sistemas e adquirindo a sua experiência. De fato, as interações sociais entre

indivíduos moldam sua visão e interpretação da realidade vivida no espaço socio-organizacional em que eles vivem.

Assim, nas próximas seções, vamos explorar um pouco mais sobre esses componentes já citados, abordando os componentes Pessoas e Tecnologias e, em seguida, o pilar da informação.

2.2. Tecnologias e pessoas

Em continuidade ao nosso estudo, sob os elementos dos sistemas de informação, sobre a vertente da Tecnologia e Pessoas, apresentaremos o ambiente de programação Scratch e seus novos usuários, mas antes de detalharmos esse elemento de estudo, vamos introduzir um pouco a história da Computação.

2.2.1 Linguagem de programação

O homem, desde seus primórdios, sempre buscou criar máquinas que o auxiliassem na execução de suas tarefas, reduzindo o esforço e maximizando seu tempo; com os computadores, não tem sido diferente. Atualmente, é quase impossível conhecer alguém que não tenha ouvido falar em linguagem de programação ou programação de computadores. As linguagens de programação podem ser classificadas e avaliadas através de diversos critérios, sendo os critérios de legibilidade, capacidade de escrita e confiabilidade os mais aceitos para uma avaliação de *benchmarks*⁵.

A comunicação de instruções a um computador é realizada pelas linguagens de programação. A primeira linguagem a surgir denomina-se ADA, em homenagem à primeira programadora da história, de nome Ada Lovelace⁶. Atualmente, existem

⁵ Uma comparação entre *hardware* ou *software* sobre um cenário previamente definido.

⁶ Ada Lovelace, foi uma matemática e escritora inglesa. Hoje é reconhecida principalmente por ter escrito o primeiro algoritmo para ser processado por uma máquina, a máquina analítica de Charles Babbage

inúmeras linguagens de programação, sendo que cada uma se enquadra em um critério e possui uma sintaxe e semântica próprias. Contudo, independentemente de qual for a linguagem adotada, todas elas dependem da construção de algoritmos. Segundo Salvetti e Barbosa (1999), algoritmo é uma sequência finita de instruções ou operações cuja execução, em tempo finito, resolve um problema computacional qualquer que seja sua instância. Para Manzano (1997), algoritmos são regras formais para obtenção de um resultado ou da solução de um problema, englobando fórmulas de expressões aritméticas, como afirma Farrer (1999) :

Ação é um acontecimento que, a partir de um estado inicial, após um período de tempo finito, produz um estado final previsível e bem definido. Portanto um algoritmo é a descrição de um conjunto de comandos que, obedecidos, resultam em uma sucessão finita de ações (FARRER, 1999, p. 14).

Analisando os conceitos anteriores, é possível perceber que produzimos em nosso cotidiano uma série de algoritmos, muitas vezes sem perceber, por exemplo, quando seguimos a receita de um simples prato, visto que o mesmo possui uma série de instruções lógicas sequenciais. O que nos leva à indução de que podemos, sim, produzir esses algoritmos de maneira intuitiva, desmistificando que a programação de computadores é uma tarefa árdua e de difícil aprendizagem. Vamos conhecer, a seguir, nosso ambiente de estudo, o Scratch, no próximo tópico.

2.2.2 Ambiente de programação Scratch

O foco nessas premissas de facilidade de uso, desmitificação da complexidade e aproveitamento da capacidade de absorção de novos conhecimentos pelas crianças é que o movimento code.org busca a inserção desses novos atores nesse novo universo computacional. Existem diversos ambientes de programação voltados para crianças, por exemplo, o Kodu, ambiente visual de programação desenvolvido pela Microsoft, voltado para crianças cujo objetivo é incentivar a programação de jogos para a plataforma Xbox e computadores compatíveis com o sistema operacional Windows, de propriedade da mesma empresa. Porém, o outro ambiente de programação disponível e foco deste trabalho

é o *Scratch*, desenvolvido pelo MIT (*Massachusetts Institute Technology*), voltado para o mesmo público infantil.

De acordo com o próprio manual da Microsoft (2009), que pode ser acessado diretamente pelo endereço eletrônico <http://www.kodugamelab.com>⁷ (hotsite dessa ferramenta), o Kodu, apresentado na figura 2, é um ambiente de programação visual que permite que qualquer pessoa, mesmo sem um conhecimento dos conceitos básicos de programação, consiga criar jogos em 3D, assim como distribuir e compartilhar os mesmos, sendo esses compatíveis com o console Xbox e para computadores pessoais (PCs).

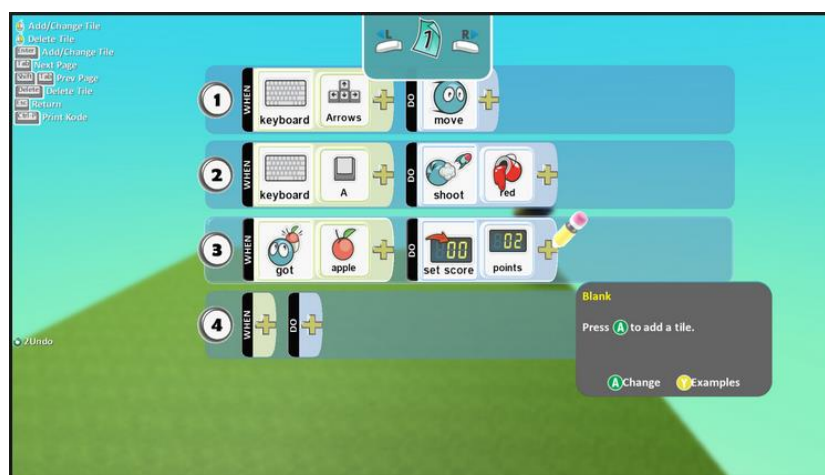


Figura 2 – Interface Kodu com os elementos e ações – em blocos
Fonte: MICROSOFT, 2014.

O ambiente de programação avaliado e escolhido para pesquisa é o Scratch⁸, visualizado na figura 3, desenvolvido pelo MIT (*Massachusetts Institute Technology*), voltado para o mesmo público infantil, também com uma abordagem interativa e em blocos de construção.

⁷ - kodu – ambiente de programação da Microsoft – o qual pode ser acessado e baixado pelo site: <http://www.kodugamelab.com>

⁸ - Scratch pode ser acessado no endereço eletrônico - <https://studio.code.org/hoc/1>

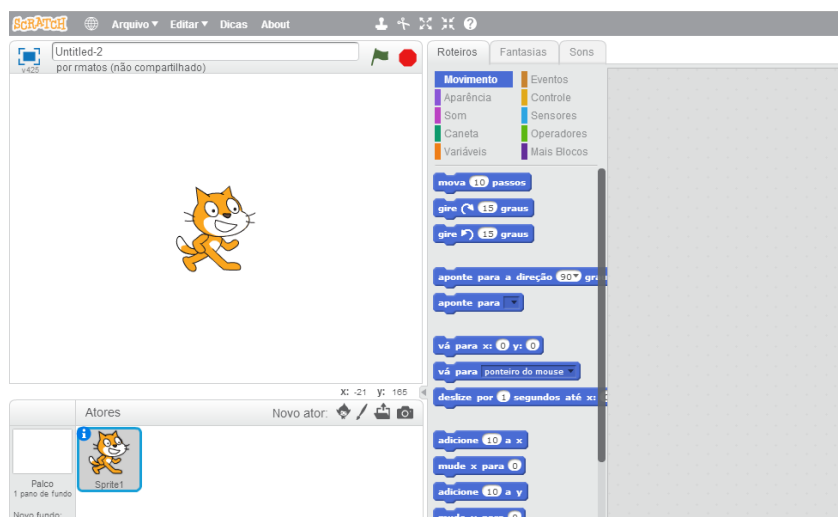


Figura 3 – Interface Scratch com elementos e ações – em blocos
Fonte: SCRATCH, 2015.

É possível perceber que ambas interfaces oferecem um ambiente mais lúdico e intuitivo, visando alcançar seus objetivos de comunicabilidade com os seus usuários finais. Percebe-se, também, que o recurso arrastar-soltar, técnica conhecida como *drag and drop*⁹, é fortemente valorizada nas interfaces gráficas. Além do recurso *drag and drop*, temos um conjunto de objetos que podem ser associados para invocar ações e comportamentos.

O *Scratch* é um ambiente acessado totalmente via web, não sendo necessário instalação e/ou configuração dessa plataforma na máquina do usuário, sendo este um ponto positivo dessa plataforma. Outro fator que merece destaque é a possibilidade de o usuário ter acesso ao código fonte¹⁰, podendo, assim, visualizar a estrutura e o código construído. Esse código fonte está disponível na linguagem *java-script*¹¹, uma das mais populares do mundo.

Assim, visando atingir esse público especificamente, existem diversas plataformas, mas optamos pelo *Scratch*, devido aos pontos positivos já citados.

⁹ Drag and Drop é a ação de clicar em um objeto virtual e "arrastá-lo" a uma posição diferente ou sobre um outro objeto virtual.

¹⁰ Código Fonte é o conjunto de palavras ou símbolos escritos de forma ordenada, contendo instruções em uma das linguagens de programação existentes, de maneira lógica

¹¹ *JavaScript* é uma linguagem de programação interpretada. Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade deste script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido

Na próxima seção, será apresentado os novos usuários dos sistemas de informação.

2.2.3 Novos Usuários

Enquanto estudos tradicionais, até meados da década de 1970, estavam voltados aos sistemas de informação em sua essência, conforme conceitos apresentados na página 14, sendo os indivíduos utilizadores agrupados demograficamente, por etnia, idade, escolaridade ou qualquer outra característica comum, não sendo tratados nessa abordagem como elemento crucial dos componentes dos sistemas de informação, não importava o conceito de experiência do usuário. O foco de pesquisa dessa época tinha predominância no caráter quantitativo, porém esses estudos passam a ter um caráter mais qualitativo a partir dos anos 1980, quando os estudiosos do comportamento começaram a perceber a ineficiência das pesquisas quantitativas em relação à contribuição das necessidades dos usuários finais.

A partir dos anos 1980, surgiu uma nova abordagem denominada alternativa, na qual os indivíduos foram vistos com necessidades cognitivas, afetivas e fisiológicas. Com base nessa nova abordagem, esses indivíduos, segundo Dervin (1989), são indivíduos que valorizam os atos de questionar, planejar, interpretar, criar, resolver e responder. Assim, por meio dessa abordagem não tradicional, percebe-se a mudança de necessidade desse novo usuário dos sistemas de informação. A pesquisa qualitativa tem como foco as reações e o comportamento dos usuários, tendo uma visão mais holística do que o método quantitativo. Outros fatores, tais como os aspectos subjetivos da experiência e do comportamento humano, são considerados nesse tipo de pesquisa. Wilson (2000) sintetiza o comportamento do usuário e suas experiências tecnológicas, como:

Um melhor entendimento do usuário para projeção de serviços de informação mais efetivo e um melhor entendimento para a criação de teorias sobre o comportamento e uso da informação; focaliza a sua atenção nas causas das reações dos usuários de informação e na resolução do problema informacional, e tende a aplicar um enfoque mais holístico do que o modelo quantitativo. Além disso, dá-se mais atenção aos aspectos subjetivos da experiência e do comportamento. A coleta de dados é vista mais como um processo do que um procedimento, requerendo constantes julgamentos analíticos (WILSON, 2000, p. 31).

As crianças não são exceções e estão cada vez mais próximas e curiosas sobre este assunto, a programação de computadores. Esses novos usuários estão cada vez mais inseridos e muitos já nascem nesse mundo digital, sendo nosso foco e objeto principal do estudo o conceito de experiência do usuário através dos pressupostos da Engenharia Semiótica que visa, por meio da aplicação de testes propostos por meio de uma metodologia específica, avaliar o perfil semiótico e as rupturas de comunicação porventura existentes na interação usuário/sistema.

Assim, conhecemos nosso segundo elemento dentre os três componentes dos sistemas de informação, as pessoas, aqui denominadas usuários finais. Em seguida, vamos tratar do último pilar, a informação.

2.3 Informação e teoria do conhecimento

Por fim, adentramos no terceiro componente de nosso sistema de informação, a informação propriamente dita e sua participação na gestão do conhecimento.

Em conjunto com a percepção difusora do conhecimento educacional, em confluência com as diretrizes e etapas da teoria da criação do conhecimento, buscamos relacionar essa ligação binomial entre a teoria de gestão do conhecimento e os propósitos do projeto code.org. Em continuidade à fundamentação desta pesquisa, na próxima seção serão apresentados os pilares e as características básicas da teoria da criação do conhecimento.

2.3.1 Espiral do conhecimento

É de suma importância e significância entendermos o conceito de gestão do conhecimento para prosseguirmos com o trabalho. Davenport e Prusak (1998) definem o conhecimento como uma mistura das experiências, dos valores, da informação contextual e da intuição que fornece uma estrutura para avaliar e incorporar experiências e novas informações. A gestão do conhecimento tem um caráter interdisciplinar, que envolve profissionais de diversas áreas: Administração, Computação, Ciência da Informação, Educação, dentre outros.

Além desse caráter interdisciplinar, podemos perceber que não existe um consenso entre os processos existentes na gestão de conhecimento. Segundo Murray e Myers (1997), temos três processos: criação, disseminação e utilização. Já para outros autores (SALMAZO, 2004; SVEIBY 1998; EBOLI, 1999) quatro etapas são indicadas: criação, aquisição, compartilhamento e utilização. Para Young (2010), existem cinco etapas, sendo elas: identificação do conhecimento, criação do conhecimento, armazenagem do conhecimento, compartilhamento do conhecimento e aplicação do conhecimento.

Independentemente das etapas, dos processos e das visões de diversos autores, nossa abordagem baseia-se no princípio de funcionamento do processo de criação do conhecimento organizacional proposto e desenvolvido por Nonaka e Takeuchi (1997). A criação do conhecimento organizacional, segundo Nonaka e Takeuchi (1997) representa a capacidade da empresa em criar um novo conhecimento, difundi-lo e incorporá-lo a produtos/serviços e sistemas de processos. Para os autores, o processo de criação do conhecimento atua sobre duas dimensões: uma epistemológica e outra ontológica. Do ponto de vista ontológico, podemos identificar os níveis das entidades criadoras do conhecimento (individual, grupal, organizacional e interorganizacional). Para a dimensão epistemológica, temos a distinção entre o conhecimento tácito e o explícito. Visando o entendimento dessas duas dimensões, a espiral do conhecimento (Figura 4) proposta por Nonaka e Takeuchi (1997), está apresentada abaixo:

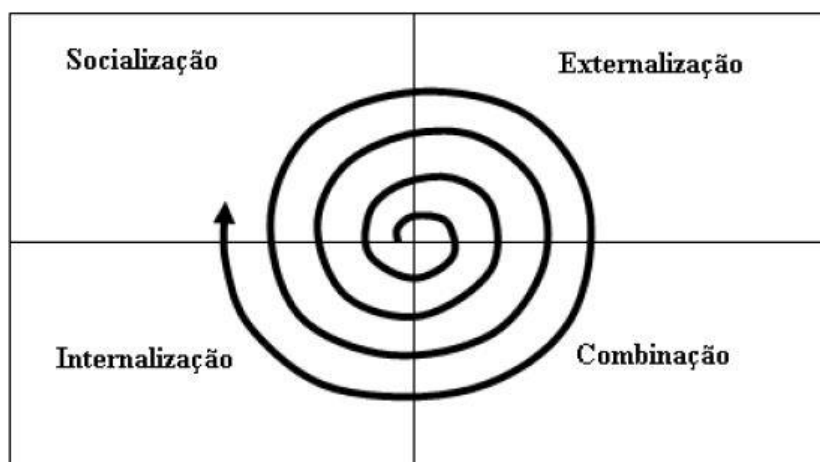


Figura 4 – Espiral do conhecimento
Fonte: NONAKA; TAKEUCHI, 1997, p. 80.

A espiral do conhecimento mostra o fluxo do conhecimento entre os 04 modos de conversão. No primeiro modo, denominado socialização, a troca de conhecimento se inicia com as pessoas, aplicando ali seu conhecimento tácito, porém é preciso avançar e compartilhar esse conhecimento com a externalização do mesmo. Na externalização, temos a primeira conversão de tácito para explícito, visto que o conhecimento individual passa a ser compartilhado com um grupo. Na combinação, o conhecimento ainda trafega do explícito para explícito, porém passa-se de um grupo para uma organização. Por fim, na internalização ocorre a conversão do conhecimento de explícito para o tácito, no qual o conhecimento passa da organização para o indivíduo.

Ao acessarmos o *website* www.code.org, apresentado na figura 5, podemos realizar o acesso direto ou através de login e senha a ser definido pelo usuário. Optamos por ter nossa própria conta, com nosso estudo analisando as interfaces e os recursos disponibilizados a partir do *link* <http://learn.code.org/hoc/1>.

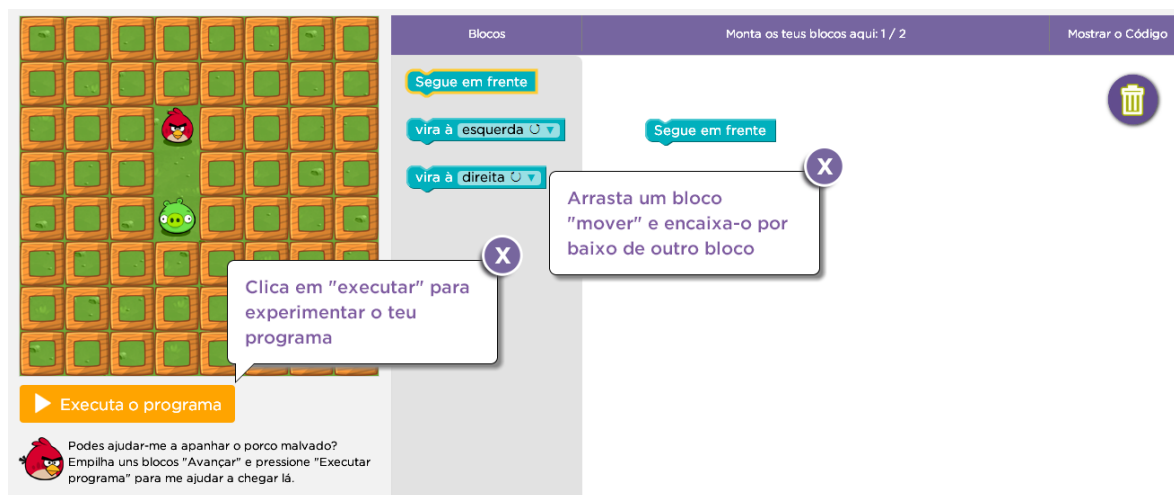


Figura 5 – Interface inicial plataforma Scratch

Fonte: PROJETO CODE ORG, 2015.

Nesse ambiente, identificamos o padrão de design mais limpo, representando uma “limpeza visual” (Clean Design), no qual é possível observar um visual agradável e com os elementos devidamente dispostos em tela. Além dessa interface apresentada acima, existem inúmeros outros recursos disponíveis no endereço citado. A ferramenta denominada Scratch pode ser vista como um ambiente lúdico de aprendizagem que tem como propósito a disseminação do conhecimento, em

específico a lógica das linguagens de programação. Assim, percebe-se que para alcançar o objetivo primário do projeto code.org, a troca de conhecimento é fundamental. O fluxo com base na teoria do conhecimento será correlacionado a seguir.

O modo de combinação do conhecimento, apresentado na Figura 4 – Espiral do conhecimento, pode ser percebido através da disponibilização da ferramenta, com a junção de um conjunto de empresas, além do MIT (*Massachusetts Institute of Technology*), que criou e mantém a ferramenta Scratch. Percebe-se que o conhecimento explícito das organizações está estruturado, visando transmitir o conhecimento da lógica de programação para os indivíduos, ou seja, a conversão do conhecimento explícito das organizações para o conhecimento tácito do indivíduo. Em continuidade ao fluxo da teoria do conhecimento, os indivíduos podem explorar essa ferramenta, trocando experiência e gerando novos modelos mentais de conhecimento, ou seja, aqui, o indivíduo, através dessa experiência exploratória, cria o conhecimento tácito da etapa de internalização.

O projeto disponibiliza uma área exclusiva para multiplicadores do conhecimento (professores, tutores e/ou instrutores), assumindo um caráter grupal, em que a troca da experiência individual se transforma em uma troca de experiência coletiva. Assim, o conhecimento ainda persiste no conhecimento tácito. Esse caráter grupal, coletivo, caracteriza o modo de conversão denominado socialização. O projeto disponibiliza ainda um conjunto de recursos, tais como fórum, chats, tutoriais, manuais e uma rede de colaboradores, que estão voltados a absorver a experiência coletiva e suas informações devolutivas, visando garantir a transformação do conhecimento tácito da etapa da socialização em conhecimento explícito da etapa da externalização. Com isso, o processo de melhoria continua através da experiência do público-alvo e pode ser percebida, visando, assim, garantir a qualidade da mesma e de todo o processo.

Agora que foi apresentada as etapas da teoria do conhecimento com o projeto code.org, é possível entender que a conversão do conhecimento de tácito para explícito ocorre com o auxílio de diversas ferramentas, recursos, conforme quadro 1 visualizado a seguir:

Quadro 1 – Recursos aplicados à conversão do conhecimento

CONVERSÃO	RECURSOS
Tácito para Tácito	• Experiência do usuário, entre outros.
Tácito para Explícito	• Uso de vídeos, tutoriais, filmes, podcasts, entre outros.
Explícito para Explícito	• Definição de padrões, melhores práticas e workflow de processos.
Explícito para Tácito	• Leitura de tutoriais, fluxos de processos, instruções normativas, manual dos usuários, entre outros.

Fonte: Elaborado pelo autor

Por fim, pode-se identificar um layout simples, agradável e intuitivo. As etapas da teoria do conhecimento propostos por Nonaka e Takeuchi (1997) podem ser devidamente observadas e identificadas no ambiente apresentado. As etapas da transmissão do conhecimento (Internalização, Socialização, Externalização e Combinação) estão presentes nesse ambiente de aprendizagem e vão ao encontro dos conceitos teóricos apresentados ao longo deste trabalho.

Na próxima seção, o conhecimento em sua essência com o processo de aprendizagem e conhecimento será apresentada.

2.3.2 Aprendizagem e conhecimento

Um dos pontos principais na metacomunicação usuário-sistema atua no elo binário aprendizado-conhecimento. O processo de aprendizagem-conhecimento é possível quando a transmissão das mensagens do sistema são devidamente interpretadas por seus usuários, gerando a transformação desse aprendizado em um conhecimento. Segundo Nonaka e Takeuchi (1995), a construção do conhecimento é reconhecida através da sinergia entre o conhecimento tácito e o conhecimento explícito e ainda quando é possível transformar esse conhecimento tácito em conhecimento explícito.

Segundo Oliveira e Porrozzini (2009), o processo de aprendizagem não é apenas o repasse de informações, é algo muito mais amplo que exige adaptação e reconstrução das informações recebidas, gerando, assim, o conhecimento. Para as interfaces e os ambientes abordados em nossa pesquisa, esse binômio aprendizado-conhecimento se enquadra perfeitamente com o aprendizado lúdico e cognitivo infantil. É importante salientar que a metacomunicação entre sistema-

usuário deve ser capaz de permitir que cada indivíduo siga e aprenda em seu próprio ritmo e construa gradativamente sua experiência semiótica. Com base em todos os embasamentos citados acima, percebe-se a necessidade de um estudo mais detalhado sobre a metacomunicação e a transmissão do conhecimento, principalmente nessa era digital, em que as crianças estão cada dia mais envolvidas e, dessa forma, podemos repensar como ocorrerá esse processo de conhecimento nas futuras gerações.

Para De Souza e Leitão (2009), a Engenharia Semiótica¹² deve ser aplicada visando investigar a estrutura, os processos e os efeitos da metacomunicação no binômio *designer*-usuário e seu contexto de interação. A Engenharia Semiótica é uma abordagem em que essa relação *designer*-usuário ocorre através de um processo de comunicabilidade. Entender o conceito de “experiência” da Engenharia Semiótica, assim como os métodos de avaliação de comunicabilidade e os conceitos básicos de aprendizagem-conhecimento, é algo que será tratado a seguir, visando uma melhor compreensão desta pesquisa.

2.4 Semiótica Peirceana e a Engenharia Semiótica

A Semiótica é a disciplina que estuda ícones, índices e símbolos, seus significados, sua comunicação e seus processos. A seguir, vamos adentrar nos conceitos elementares da Engenharia Semiótica.

2.4.1 Semiótica: ícone, índice e símbolo

Para o pensamento semiótico de Charles Sanders Peirce (2005), um signo não tem necessariamente de ser uma representação mental, mas pode ser uma ação ou experiência, ou mesmo uma mera qualidade de impressão. O signo é uma

¹² Semiótica é a disciplina que estuda os signos, índices e ícones, seus significados, sua comunicação e todos os processos culturais (ECO, 1976). Segundo Silveira, De Souza e Barbosa, os conceitos de semiose e abdução sustentam essa disciplina. Semiose representa um conjunto de signos disparados por um signo inicial (signo, índice ou ícone). Abdução representa um tipo especial de inferência que envolve elementos do raciocínio dedutivo e intuitivo.

coisa que representa uma outra coisa: seu objeto. A noção de *interpretante* é, para Peirce, um processo relacional que se cria na mente do intérprete.

Em resumo, como afirma D'Alviella (1995), o termo símbolo passou gradualmente a se referir a tudo aquilo que, seja por acordo geral ou analogia, representava convencionalmente alguma coisa ou alguém. Santaella (1999) chama a atenção para a separação entre iconicidade, indexicalidade e simbolicidade que existe com fins analíticos apenas e apresenta uma citação de Peirce *apud* SANTAELLA (1999), extremamente enriquecedora:

Um substantivo próprio, quando nos deparamos com ele pela primeira vez, está existencialmente conectado a algum percepto, ou outro conhecimento individual e equivalente, do individual que esse nome designa. Então, e somente então, é o referido nome um índice genuíno. Na próxima vez que nos deparamos com ele, é preciso considerá-lo como um ícone daquele índice. Uma vez adquirida uma familiaridade habitual com ele, o nome torna-se um símbolo cujo interpretante o representa como ícone de um índice do individual nomeado (PEIRCE *apud* SANTAELLA, 1999, p. 238).

Peirce afirma que algo se constitui como um signo desde que assim seja "(...) usado e compreendido como tal" (PEIRCE, 2005, p. 76), se for uma convenção ou um hábito natural, não sendo necessário levar em consideração a origem que o levou a ser selecionado como tal. Peirce construiu uma definição de signo em que se expõem os elementos que o formam e na qual entram as categorias, e mostra que se trata de algo em constante evolução.

Um signo é tudo aquilo que está relacionado com uma segunda coisa, seu objeto, com respeito a uma qualidade, e modo tal a trazer uma terceira coisa, seu interpretante, para uma relação com o mesmo objeto, e de modo tal a trazer uma quarta para uma relação com aquele objeto na mesma fórmula, *ad infinitum* (PEIRCE, 2005, p. 28).

Numa outra definição, Peirce (2005) apresenta, de uma só vez, os conceitos de Signo, ou *Representâmen*, Objeto e Interpretante:

Um signo, ou *representâmen*, é aquilo que, sob certo aspecto ou modo, representa algo para alguém. Dirige-se a alguém, isto é, cria, na mente dessa pessoa, um signo equivalente, ou talvez um signo mais desenvolvido. Ao signo assim criado denomino interpretante do primeiro signo. O signo representa alguma coisa, seu objeto. Representa esse objeto não em todos os seus aspectos, mas com referência a um tipo de ideia que eu, por vezes, denominarei fundamento do *representâmen* (PEIRCE, 2005, p. 46).

Sendo assim, o autor explica como esse interpretante age na mente humana. Segundo Peirce (2005), uma relação triádica é genuína quando os três elementos (*Representâmen*, Objeto e Interpretante) estão ligados “[...] de um modo tal que não consiste em nenhum complexo de relações diádicas” (PEIRCE, 2005, p. 63). Por isso, só o símbolo é genuíno, pois o índice e o ícone não estão ligados dessa forma ao interpretante. Para essa relação, o gráfico a seguir explica a interconexão desses elementos.

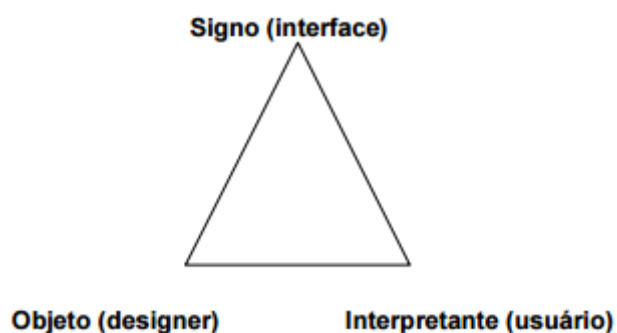


Figura 6 – As três entidades do signo

Fonte: Elaborado pelo autor

Segundo Santaella (1999), as relações semióticas são triádicas, isto é, envolvem basicamente o signo, o objeto que o signo representa e o interpretante. O signo pode ser genericamente considerado um elemento de comunicação que representa algo para alguém; o objeto que o signo representa é algo que transmitirá uma mensagem a alguém; o interpretante é o receptor do signo. Para um melhor entendimento sobre este conceito de interpretante, é importante assinalar que para o vocabulário conceitual peirceano, o interpretante não se trata do indivíduo que recebe o signo, mas sim do resultado do processo de semiose, compondo a sua experiência.

Segundo a teoria peirceana, os signos podem ser divididos em três tipos: ícones, índices e símbolos. Ícones são vistos como qualidades que não representam nada, apenas se apresentam. Ora, se não representa, o ícone é um quase-signo. Uma pintura não figurativa, por exemplo, desconsiderando o fato de que é um quadro que está lá, o que já faria dela um existente singular e não uma pura qualidade, mas considerando-a apenas no seu caráter qualitativo (cores,

luminosidade, volumes, textura, formas) é um ícone. O objeto do ícone, portanto, é sempre uma simples possibilidade, isto é, possibilidade do efeito de impressão que ele está apto a produzir ao excitar nosso sentido.

A semiótica peirceana e seus epígonos, pilar que fundamenta este trabalho, afirma que símbolos, muitas vezes, surgem pelo desenvolvimento de outros signos, especialmente ícones ou signos com características icônicas e simbólicas ao mesmo tempo. O índice, por sua vez, é sempre binário: a ligação de uma coisa com outra. O índice representa ou evoca, tem relação direta com o objeto que assinala, por contiguidade. O índice atua como um elemento de conexão, de inferência, ou seja, atua como um indicador. Quando, em si mesmo, o signo é de lei, por convenção ou pacto coletivo e determina que aquele signo represente seu objeto, é um símbolo. As palavras são exemplos desses signos gerais. Desse modo, o objeto de uma palavra é uma ideia abstrata.

O último elemento citado é o símbolo. O símbolo é um signo que representa o objeto de maneira arbitrária e intuitiva ou evoca um conceito. Símbolos, entretanto, trazem embutidos em si caracteres icônicos e indiciais. O que seria de uma frase, por exemplo, sem a ordem das palavras e o padrão de sua estrutura, justamente seu caráter icônico e indicial que nos leva a compreendê-la?

Uma vez que os conceitos fundamentais da semiótica peirceana, além dos modos dos signos foram apresentados, vamos examinar os modos pelos quais processamos sensações em percepções e cognições – a semiose -, o que foi denominado por Peirce (2005) como as três fases da experiência.

2.4.2 A semiótica peirceana e as três fases da experiência: primeiridade, secundidade, terceiridade

A tríade peirceana (primeiridade, secundidade e terceiridade) pode ser entendida como um processo gradativo de atingirmos nossa experiência: 1) sensitiva; 2) perceptiva; 3) cognitiva, em outros termos, a “semiose”. Em resumo, Peirce (2005) classifica em três momentos ou elementos que constituem toda a experiência, ou seja, nosso aprendizado. As categorias peirceanas citadas são,

segundo Santaella (1999), como categorias fenomenológicas de Peirce, que não são entidades mentais, mas modos de operação do pensamento signo que se processam na mente e podem ser apresentados como:

- **Primeiridade:** Pode ser vista como a primeira sensação, contato com alguma coisa em qualidade absoluta. Não possui parâmetros comparativos, pois isso o levaria à secundidade. A primeiridade representa a pura qualidade de ser e de sentir, categoria de caráter qualitativo.
- **Secundidade:** Pode ser vista como a segunda sensação que conseguimos associar a algo já sentido, conhecido. A secundidade refere-se a uma binaridade – agir e interagir – do domínio do mundo conhecido, categoria de caráter existencial.
- **Terceiridade:** Pode ser vista a área da inteligibilidade ou do pensamento em signos através dos quais representamos e interpretamos o mundo. Essa interpretação obedece a leis, padrões, normas, paradigmas, modelos, teoria, etc., é a categoria de caráter legislativo.

Para o autor Almeida (2009) que também se preocupa em definir os conceitos da fase da experiência:

A noção de qualidade é encontrada na primeira categoria, por incluir qualidades de sentimento e sensação. Essa qualidade, quando se relaciona a outras por um tipo de colisão ou reação, constitui uma experiência direta, pertencente à segunda categoria. Os fatos e as experiências, ao serem mediados ou representados por um terceiro elemento, subsomem-se à terceira categoria. (ALMEIDA, 2009, p. 195).

Esses conceitos apresentados além de fundamentar nosso trabalho são relevantes para que possamos, através da experiência de nosso grupo focal com base nos conceitos semióticos de “experiência” ou semiose, avaliar a ferramenta de programação estudada com base nesses pressupostos da Engenharia Semiótica. Contudo, visando um melhor entendimento dessa experiência semiótica, outros conceitos complementares são necessários para uma análise mais efetiva. Porém, antes de avançarmos, assim como anunciado, vamos correlacionar os conceitos teóricos de experiência com nossos usuários.

2.4.3 A semiótica aplicada a ambientes digitais

A Engenharia Semiótica é o elemento-chave e o pilar de sustentação deste trabalho. De uma maneira global, pode-se afirmar que a Engenharia Semiótica não tem o propósito de criar métodos preditivos que garantem resultado direto, único e absoluto. A Engenharia Semiótica visa explicar a interação Homem-Computador (IHC), utilizando-se, para isso, uma avaliação de comunicabilidade entre os elementos envolvidos. Sabemos que a Engenharia Semiótica é uma disciplina em evolução. A Figura 7 ilustra os principais marcos dessa linha evolutiva.

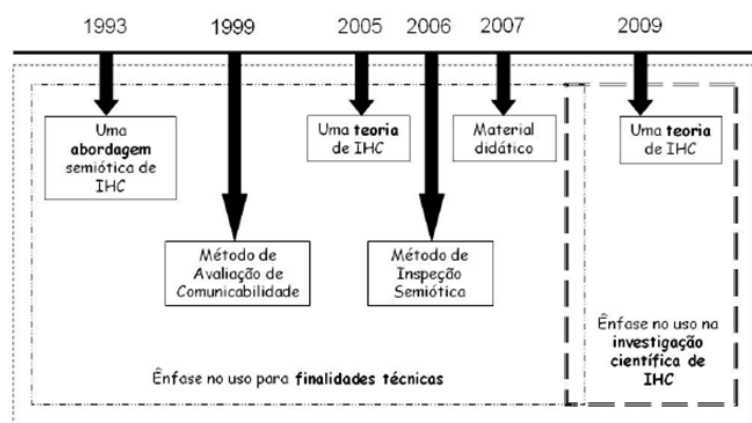


Figura 7 – Linha do tempo: evolução da Engenharia Semiótica

Fonte: RODRIGUES, 2014.¹³

A Figura 7 nos mostra o quadro evolutivo dessa disciplina, no qual os anos de 1999 e de 2006 passam a ser marcantes, visto que os mesmos apresentam um método para a verificação dessa experiência.

O desenvolvimento de interface de comunicação requer cuidado e atenção pelo lado dos desenvolvedores, pois falhas de conceitos, erros de projetos e/ou especificação podem gerar rupturas de comunicação. Para De Souza e Leitão (2009), as rupturas encontradas significam que a mensagem não foi recebida adequadamente pelo destinatário, gerando assim um problema de comunicabilidade.

As interfaces de aplicações podem ser avaliadas sobre três vertentes, sendo elas a usabilidade, que os projetistas julgam prioritárias para os sistemas (PREECE;

¹³ RODRIGUES, Rodrigo Fonseca. *Caderno de estudos da disciplina Engenharia Seimiótica*. Belo Horizonte: Universidade Fumec, 2014. Não publicado.

ROGERS; SHARP, 2002), a comunicabilidade, que segundo Prates, De Souza e Barbosa (2000), refere-se à capacidade dos usuários em entenderem as mensagens transmitidas pelos signos de interface e, por fim, a acessibilidade, que se refere à capacidade que pessoas com deficiência possam entender e utilizar os sistemas computacionais (W3C, 2015).

A Engenharia Semiótica dispõe de dois grandes métodos que visam a avaliação da comunicabilidade de sistemas interativos. Um dos métodos disponíveis é o Método de Avaliação de Comunicabilidade, também conhecido pela sigla MAC. Para De Souza (2005), esses métodos são qualitativos e interpretativos, tendo ambos, como objetivo, identificar os potenciais problemas na qualidade da comunicação projetista-usuário, ou seja, os potenciais problemas de comunicabilidade.

Quanto mais rupturas ou quanto mais severas as rupturas, mais baixa a comunicabilidade da interface. Atualmente, existem dois métodos para se avaliar a comunicabilidade de uma interface: O Método de Inspeção Semiótica (MIS) e o Método de Avaliação de Comunicabilidade (MAC). (PRATES; BARBOSA, 2007, p. 273).

O Método de Inspeção Semiótica (MIS) é um método que propõe a verificação da comunicabilidade pela visão de um especialista. O Método de Avaliação da Comunicabilidade (MAC), por sua vez, visa identificar a comunicabilidade por meio da interpretação do avaliador sobre a experiência real dos usuários. Dessa forma, justifica-se o uso do método MAC como o método mais adequado para nosso trabalho, pois o mesmo visa identificar as rupturas de comunicação entre sistemas e seus usuários finais. O Método MAC está focado na validação de determinadas partes, consideradas críticas, da aplicação. Esse método basicamente é composto em uma abordagem sequencial, em três fases, que consistem em preparar, executar os testes e, por fim, realizar a análise dos dados.

Como o foco de nosso estudo visa avaliar a comunicabilidade entre usuário-sistema, teremos como foco esse marco apresentado em 1999, conhecido como Método de Avaliação de Comunicabilidade, que será detalhado no próximo tópico.

2.4.4 Comunicação na Engenharia Semiótica

O processo de comunicação envolve diversos elementos aplicados. Em geral, temos os *designers*, responsáveis pelo desenvolvimento de uma interface amigável de comunicação, os usuários que irão interagir com essa interface desenvolvida, e a própria aplicação disponibilizada.

O modelo de comunicação apresentado a seguir se fundamenta no modelo de Jakobson (1973) e, baseado nele, a Engenharia Semiótica propõe os elementos para a elaboração dessa metacomunicação. Os elementos presentes são emissor, destinatário, contexto, canal, código e mensagem.

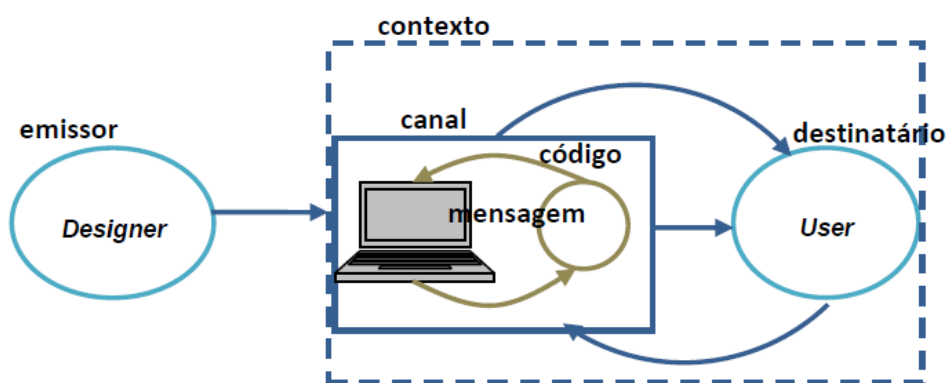


Figura 8 – Processo de comunicação da Engenharia Semiótica
Fonte: DE SOUZA, 2005, p. 40.

Segundo Eco (1984), o processo de comunicação ocorre quando um Emissor (E) envia uma Mensagem (M) através de um canal para um Destinatário (D) utilizando-se de um código que deve ser comum ao emissor e ao receptor.

A comunicabilidade entre emissor e destinatário se dá através das mensagens, as quais são emitidas pelos signos de interface e interpretadas pelos usuários, de modo a estabelecer uma comunicação interativa. De Souza e Leitão (2009) expõem que o envio de uma mensagem ao usuário é realizado através dos elementos de interface do sistema desenvolvido.

Os usuários, por sua vez, vão entendendo as mensagens à medida que interagem com a aplicação. Nesse processo de comunicação, os elementos

fundamentais da semiótica – ícone, índices e símbolos – responsáveis por essa metacomunicação serão apresentados na próxima seção.

2.4.5 Método de Avaliação de Comunicabilidade – MAC

O Método de Avaliação da Comunicabilidade exige a participação de usuários, os quais irão executar tarefas predefinidas em um ambiente que deve estar livre de interferências externas, sem o perigo de desvios de atenção e interrupções desnecessárias. O método MAC é dividido na aplicação de 03 etapas, as quais são detalhadas a seguir.

Leite (2012, p. 25) cita o método MAC estudado por De Souza (2005) em três fases preliminares, sendo elas: inspeção do software, seleção de atividades de inspeção e observação dos usuários. Posteriormente, é realizada a análise dos dados, que também é dividida em três passos: etiquetagem, interpretação da etiquetagem e geração do perfil semiótico.

Para um melhor entendimento, as três fases são descritas abaixo:

- **Fase da Etiquetagem:** Nessa etapa, o avaliador assiste a interação usuário-sistema e anota e/ou grava todas as dúvidas citadas pelo usuário ao longo do teste. Para facilitar posteriormente essa avaliação, para todos os questionamentos levantados pelos usuários, são associadas etiquetas (expressões) que melhor classifiquem esse tipo de ruptura.

Segundo Leite (2012), o avaliador ou especialista realiza a análise, buscando identificar os principais problemas de metacomunicação.

Os tipos de falhas de comunicação são definidos pela relação entre a intenção de comunicação e o efeito que ela causa. Quando a intenção é consistente com o efeito causado por ela, a comunicação é bem sucedida. Caso contrário, ela apresenta algum problema que pode ou não ser percebido pelos usuários (MATTOS, 2010, p. 33).

Para cada dúvida, espera-se uma expressão associada, que estará associada a uma falha, a qual pode ser definida como completa, parcial ou temporária. Segundo De Souza (2005), a falha é completa quando a intenção da comunicação e seu efeito são inconsistentes; parcial, quando parte do efeito pretendido da comunicação não é atingido; e temporária, aquela que acontece na intenção de um ato comunicativo entre usuário e sistema, quando percebida pelo usuário que tenta, então, superá-la. O Quadro 2 apresenta essas expressões e sua classificação segundo Mattos (2010).

Quadro 2 – Classificação do sistema por tipo de falha

Expressão	Tipo de Falha
Cadê?	Temporária
Socorro!	Temporária
Ué, o que houve?	Temporária
Epa!	Temporária
Onde estou?	Temporária
Assim não dá	Temporária
E agora?	Temporária
Por que não funciona?	Temporária
Vai de outro jeito	Parcial
Não, obrigado!	Parcial
Para mim está bom	Completa
Desisto	Completa

Fonte: MATTOS, 2010, p. 36.

- Fase de interpretação da etiquetagem: Nessa etapa, o avaliador assiste a interação usuário-sistema e, com base no quadro de classificação de etiquetas por tipo de falha, associa a dúvida do usuário com a etiquetagem interpretada.
- Fase de geração do perfil semiótico: Nessa etapa, o avaliador irá determinar o perfil semiótico, com base nos dados coletados e de acordo com o processo de metacomunicação transmitido pelo usuário. Segundo Mattos (2010), esse perfil é realizado com base nas seguintes perguntas:
 - 1- Quem são os destinatários da metacomunicação dos projetistas?
 - 2- Quais necessidades e desejos foram contemplados e por quê?
 - 3- Quais formas de comunicação foram julgadas preferenciais e por quê?
 - 4- Como funciona, o que realiza e não realiza a comunicação usuário-sistema e por quê?
 - 5- Qual é a visão e a razão do projeto?

O MAC pode produzir resultados significativos, visto que o mesmo avalia as experiências dos usuários, o que está em conformidade com o propósito deste trabalho. O MAC é um método que pode produzir resultados significativos, porque as experiências dos usuários sempre surpreendem os projetistas. Esses resultados podem, então, ser explorados, analisados e interpretados, visando descobrir aspectos implícitos e explícitos da metacomunicação, diferenciando, assim, quando aplicamos o MIS (DE SOUZA, 2005; DE SOUZA; LEITÃO, 2009).

Segundo Mattos (2010), o Quadro 3 descreve as expressões e a descrição de seus significados.

Quadro 3 – Etiquetas MAC

Expressão	Descrição
Cadê?	Ocorre quando o usuário sabe o que deve fazer, mas não encontra de imediato em sua interface.
Socorro!	Ocorre quando o usuário não sabe que atitude tomar e solicita apoio para o procedimento.
Ué, o que houve?	Ocorre quando o usuário não percebe a resposta enviada pelo sistema.
Epa!	Ocorre quando o usuário realiza uma tarefa indesejada e tenta retornar.
Onde estou?	Ocorre quando o usuário tenta acionar procedimentos não permitidos ao contexto em que se encontra.
Assim não dá	Ocorre quando o usuário tenta repetir um processo diversas vezes, achando que está no caminho correto, mas percebe que esse não é o caminho correto para o procedimento.
E agora?	Ocorre quando o usuário não sabe o que fazer e tenta descobrir o próximo passo.
Por que não funciona?	Ocorre quando a operação realizada não produz o resultado esperado.
Vai de outro jeito	Ocorre quando o usuário não consegue seguir as etapas pensadas pelo projetista e acaba tentando encontrar um caminho alternativo.
Não, obrigado!	Ocorre quando o usuário faz a opção de um caminho alternativo, mesmo o sistema sugerindo um caminho a seguir.
Para mim, está bom	Ocorre quando o usuário acha que concluiu todas as etapas e encerra as atividades.
Desisto	Ocorre quando o usuário não consegue cumprir a tarefa e desiste.

Fonte: MATTOS, 2010, p. 34.

As falhas de comunicação também pode ser agrupadas em categorias, as quais englobam um conjunto de expressões de comunicabilidade. Para De Souza e Leitão (2009), as categorias e suas expressões caracterizam as falhas de comunicação. Elas podem ser visualizados no Quadro 4.

Quadro 4 – Grupos da expressão de comunicabilidade

Categoria	Expressão de comunicabilidade
Falhas Completas	Desisto. Para mim, está bom...
Falhas Parciais	Não, obrigado. Vai de outro jeito.
Falhas Temporárias	Cadê? Ué, o que houve? E agora? Onde estou? Epa! Assim não dá. O que é isso? Socorro! Por que não funciona?

Fonte: DE SOUZA; LEITÃO, 2009, p. 38.

No próximo capítulo, vamos detalhar a aplicabilidade da semiótica junto ao ambiente Scratch.

3. SEMIÓTICA APLICADA AO SCRATCH

Nesta seção, dois componentes importantes da semiótica são correlacionados com os elementos da plataforma Scratch de maneira prática, visando apresentar esses conceitos de maneira prática e aplicável. Inicialmente, apresentamos os signos apresentados pela plataforma; em continuidade, uma abordagem sobre o conceito de experiência e suas temporariedades será apresentado; por fim, a metodologia de pesquisa será detalhada.

3.1 Ícone, índice e símbolo aplicados à plataforma Scratch

O estudo das linguagens e dos signos é muito antigo e não são fruto da era tecnológica. Ao contrário, a tecnologia tem recorrido a esses signos para difundir e

ampliar seu número de usuários. O projeto code.org visa a formação e a capacitação de crianças no uso das linguagens de programação. Para Peirce (2005), autor pioneiro dessa ciência conhecida como semiótica, ela estuda todos os processos de formação do significado na mente de cada indivíduo. Do mesmo modo, para Santaella (1999), o nome semiótica possui origem grega, denominada *semeion*, que significa signo.

Segundo a teoria peirceana, os signos podem ser divididos em três tipos: ícones, índices e símbolos. Ícones são vistos como qualidades que não representam nada, apenas se apresentam. Ora, se não representa, o ícone é um quase-signo. Uma pintura não figurativa, por exemplo, desconsiderando o fato de que é um quadro que está lá, o que já faria dela um existente singular e não uma pura qualidade, mas considerando-a apenas no seu caráter qualitativo (cores, luminosidade, volumes, textura, formas) é um ícone. O objeto do ícone, portanto, é sempre uma simples possibilidade, isto é, possibilidade do efeito de impressão que ele está apto a produzir ao excitar nosso sentido. Segundo Peirce (2005), autor que fundamenta esta dissertação, símbolos, muitas vezes, surgem pelo desenvolvimento de outros signos, em especial os ícones ou signos com características icônicas e simbólicas simultâneas.

O índice, por sua vez, é sempre dual: a ligação de uma coisa com outra. O índice representa ou evoca, tem relação direta com o objeto que assinala, por contiguidade. Quando em si mesmo, o signo é de lei, por convenção ou pacto coletivo e determina que aquele signo represente seu objeto, é um símbolo. As palavras são exemplos desses signos gerais. Desse modo, o objeto de uma palavra é uma ideia abstrata. O índice atua como um elemento de conexão, de inferência, ou seja, atua como um indicador.

O símbolo é um signo que representa o objeto de maneira arbitrária e intuitiva ou evoca um conceito. Símbolos, entretanto, trazem embutidos em si caracteres icônicos e indiciais. O que seria de uma frase, por exemplo, sem a ordem das palavras e o padrão de sua estrutura, justamente seu caráter icônico e indicial que nos leva a compreendê-la?

Existe ainda o qualissigno, que é uma qualidade sensível considerada como signo ou veículo (uma cor, um som), o sin-signo, que é um objeto ou evento (organograma, desenho), e o legi-signo, que é uma lei ou norma.

O primeiro signo cria em uma mente um interpretante, ou seja, um outro signo equivalente ou mais desenvolvido. A esse signo equivalente denominamos objeto (ou referente). A linha tracejada apenas nos alerta que a relação entre objetos e signos é diferente das relações entre signos e interpretantes e também da relação entre objeto e interpretante. A relação entre signos e objetos varia em função dos tipos de signos, como ícone e índices. NETTO (2003) deixa claro, todavia, que essa linha pode ser “cheia” porque será visto que “[...] dificilmente um tipo de signo deixa de estar marcado pela presença de outro ou outros” (NETTO, 2003, p. 57).

Ao acessarmos o *website* www.code.org, podemos realizar o acesso direto ou através de login e senha a ser definido pelo usuário. Optamos em ter nossa própria conta, com nosso estudo analisando as interfaces a partir do *link* <http://learn.code.org/hoc/1>.

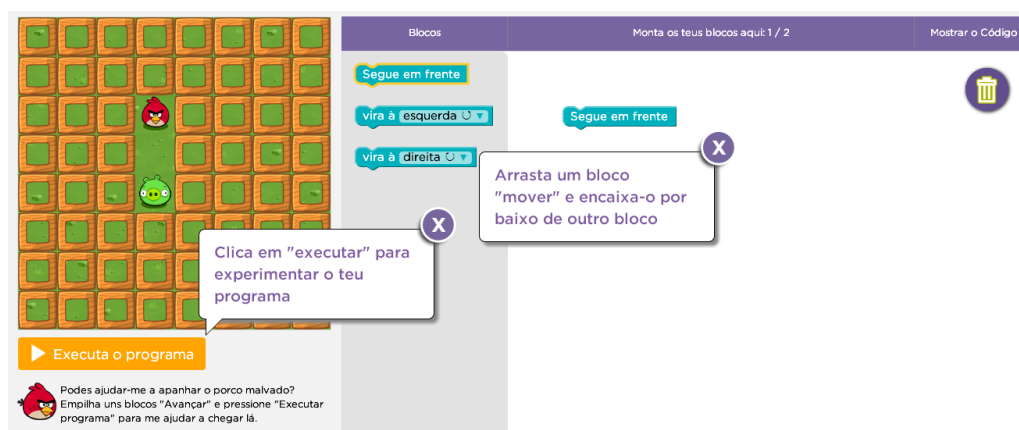




Figura 10 – Interface da primeira fase do Scratch

Fonte: PROJETO CODE ORG, 2015.

Em resumo, como afirma D’Alviella (1995, p. 145), o termo símbolo passou gradualmente a se referir a tudo aquilo que, seja por acordo geral ou por analogia, representava convencionalmente alguma coisa ou alguém. Assim, com base nessa argumentação, podemos indicar os elementos abaixo como símbolos-icônicos, visto que os mesmos conseguem representar convencionalmente alguma coisa.

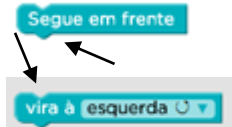
Quadro 5 – Signos icônicos

	<p>Convencionalmente, este objeto é um elemento que podemos interpretar como um botão que, ao ser clicado, gera um evento que fecha alguma janela ou caixa. Nessa aplicação, o elemento é utilizado para fechar a caixa de mensagem.</p>
	<p>Este objeto também permite a interpretação desse símbolo-icônico, pois, em interfaces computacionais, indica a remoção ou exclusão de algo. Nessa aplicação, esse elemento permite que ao arrastarmos elementos para ele, o mesmo remova os blocos de comandos inseridos na janela da aplicação.</p>

Fonte: SCRATCH, 2015.

Para representar os índices, podemos entender que os mesmos indicam algo ou sugerem alguma ação; assim o índice atua como um elemento de conexão, de inferência, ou seja, atua como um indicador.


Quadro 6 – Signos índices

	<p>O formato desses signos, destacados pelas setas, podem ser citados como índices, visto que sugerem um encaixe entre os elementos, assim como um jogo de lego.</p>
--	--

Fonte: SCRATCH, 2015.

Por fim, podemos apresentar os símbolos tradicionais, ou seja, símbolo é um signo que representa o objeto de maneira arbitrária e intuitiva ou evoca um conceito. Símbolos, entretanto, trazem embutidos em si caracteres icônicos e indiciais. Frases podem ser vistas como símbolos, pois representam comandos ou ordens, mais especificamente legi-signo, que é uma lei ou norma.

Quadro 7 – Signos símbolos

	<p>Este símbolo nos mostra uma ordem, na qual o intérprete, ao clicar, poderá executar sua aplicação.</p>
---	---

Fonte: SCRATCH, 2015.

Os signos da Engenharia Semiótica estão presentes nesse ambiente de aprendizagem e vão ao encontro dos conceitos teóricos apresentados ao longo deste trabalho. Tem-se, ainda, que essa correlação possa contribuir de maneira mais prática no entendimento dos signos da Engenharia Semiótica e que todos os interessados nesse assunto consigam usufruir desse exemplo.

Na próxima seção, vamos apresentar as três fases da experiência da semiótica peirceana e, na sequência, realizaremos a correlação da mesma com nosso público-alvo e suas experiências.

3.2 As três fases da experiência aplicadas aos usuários

Os conceitos apresentados na seção anterior, além de fundamentar nosso trabalho são relevantes para que possamos, através da experiência de nosso grupo focal, com base nos conceitos semióticos de “experiência” ou semiose, avaliar a ferramenta de programação referenciada e as experiências dos usuários da mesma.

Percebe-se a primeiridade nas expressões faciais dos usuários, em que se notam expressões surpresas durante a utilização da aplicação, independentemente do sucesso ou da frustração que o usuário experimenta. Esse primeiro contato pode ser visto como a primeira sensação, contato com alguma coisa em qualidade absoluta, em que o usuário não possui parâmetros comparativos, pois isso o levaria à secundidade. A secundidade, por sua vez, é adquirida ao caminhar das etapas propostas pela aplicação. Nesse momento, o usuário começa a perceber que, repetindo certas ações, instruções e operações, um comportamento já conhecido acontece. Por fim, os usuários avançam e alcançam a terceiridade, pois, a partir desse momento, o mesmo entende a lógica e o propósito da plataforma e, conseqüentemente, uma síntese dos elementos já lhe é conhecida; sua lógica de funcionamento e seus elementos já soam familiares.

Assim, podemos identificar a aplicação da tríade peirceana (primeiridade, secundidade e terceiridade) aplicada e exemplificada em nosso ambiente de programação, sendo esse um processo gradativo para atingirmos a experiência de nossos usuários. Como complemento à aplicação dos elementos semióticos e a experiência dos usuários, vamos explorar a metodologia aplicada ao estudo, no próximo capítulo.

4 METODOLOGIA DA PESQUISA

A metodologia tem, inicialmente, um caráter exploratório, devido ao estágio ainda inicial da aplicação das linguagens de programação nas escolas fundamentais¹⁴. Tal metodologia proporciona uma maior familiaridade do tema pesquisado. Nesta etapa de nossa pesquisa, vamos realizar o embasamento teórico-científico para a sustentabilidade de nosso trabalho. No segundo momento, a metodologia tem caráter descritivo, pois, através dessa metodologia, é possível avaliar temas já conhecidos com os novos identificados, ou seja, essa metodologia poderá auxiliar na conexão da Engenharia Semiótica com as descobertas da metodologia exploratória. Por fim, a utilização do grupo focal proporcionará uma interação entre pesquisador e participantes, visando colher dados a partir de uma discussão focalizada em um determinado assunto.

De acordo com diversos autores, como Debus (1997), Dall'agnol e Trench (1999), Iervolino e Pelicione (2001) e Meier e Kudlowicz (2003), o número de participantes deve variar de 8 a, no máximo, 10 pessoas, devendo o tamanho do grupo estar adaptado aos propósitos da tarefa. Para Vergara (2004), a utilização de grupo focal objetiva explicar como as pessoas consideram uma experiência, em nosso caso, a experiência semiótica de comunicação. Para Malhotra (2006), grupo focal visa ouvir um grupo de pessoas para falar de problemas que interessam ao pesquisador.

Em resumo, podemos dizer que a técnica de grupo focal utiliza-se de coleta de dados para pesquisas qualitativas apropriadas para avaliação de produtos, testes, programas promocionais e outros, visando obter um retorno sobre as expectativas de um grupo minoritário de um público-alvo. De acordo com nossa pesquisa, a utilização de testes aplicados no grupo focal definido visa obter um retorno das crianças e suas experiências em relação ao ambiente Scratch.

Na segunda e na terceira etapas, utilizamos o MAC – Método de Avaliação da Comunicabilidade –, sendo este método já definido como avaliativo da Engenharia

¹⁴ Ensino fundamental é o nome dado a uma das etapas da educação básica no Brasil. Tem duração de nove anos, sendo a matrícula obrigatória para todas as crianças com idade entre 6 e 14 anos

Semiótica. Esse método de avaliação foi aplicado ao grupo focal previamente definido. Nosso grupo focal foi composto de 6 a 10 integrantes, com faixa etária de 8 a 12 anos de idade, ou seja, todos ainda no nível fundamental de escolaridade. Inicialmente apresentamos os contextos básicos, tais como: o que é um software e linguagem de programação, seus conceitos e suas aplicabilidades, sendo todos esses assuntos tratados de maneira introdutória. Logo após os questionamentos serem devidamente esclarecidos, os usuários realizaram os testes de comunicabilidade, que consistem em cumprir 20 tarefas sequenciais e gradativamente dificultadas. Em seguida à aplicação desses testes, os dados foram tabulados de maneira qualitativa, de acordo com os pressupostos aplicados dos conceitos da Engenharia Semiótica, de acordo com as etiquetas realizadas pelos integrantes do grupo focal e submetidas à análise do pesquisador. Por fim, um questionário pós-teste foi preenchido pelos usuários voluntários de nossa pesquisa.

Esse trabalho tem como base a perspectiva interpretativa que pode contribuir para que pesquisadores de sistemas de informação compreendam o pensamento e a ação de indivíduos em contextos social e organizacionais (KLEIN; MYERS, 1999; WALSHAM, 2006). Assim como outras abordagens, a pesquisa interpretativa não está livre de princípios e critérios a serem seguidos que caracterizam e nela imprimem qualidade.

Segundo Yin (2009), uma abordagem por meio de um estudo de caso incorporado pode ser único por se tratar de uma única organização e incorporado, já que envolveu um grupo focal definido. Podemos ainda afirmar que a pesquisa aplica a técnica da observação participante (NANDHAKUMAR; JONES, 2002), visto que um instrutor atuará como interpretador das dúvidas e dos questionamentos do grupo focal. Para Yin (2009), a técnica denominada análise de artefatos físicos, que consiste na avaliação de uma ferramenta, tecnologia, um instrumento ou até mesmo uma obra de arte, está sendo aplicada nessa pesquisa devido à análise do observador sobre o uso de uma tecnologia. Assim, a tecnologia da informação, através dos ambientes de programação estudados, pode ser citado como um artefato físico utilizado como fonte de dados.

Em síntese, a Figura 11 apresenta uma visão macro do Método de Avaliação da Comunicabilidade utilizado.

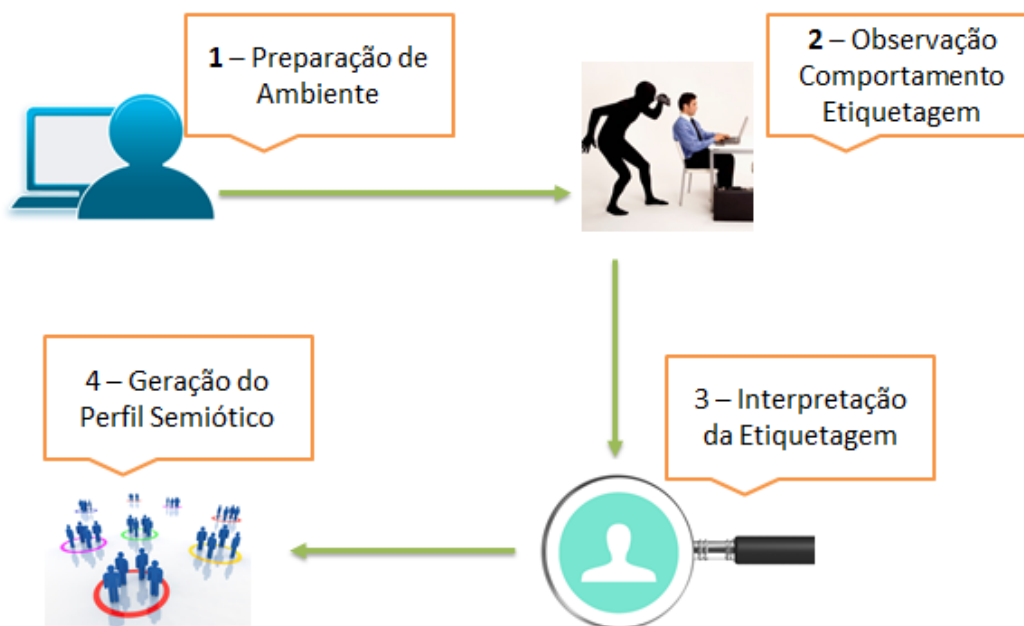


Figura 11 – Workflow do Método MAC
Fonte: Elaborado pelo autor

O processo definido pelo MAC tem como objetivo identificar como o usuário entende a metacomunicação emitida pelos elementos do sistema. Para identificarmos a metacomunicação e a interpretação dessas mensagens, os usuários finais são observados durante os testes em um ambiente controlado e suas ações e etiquetas são registradas para posterior análise e geração do perfil semiótico.

Para um melhor entendimento, uma descrição das 4 etapas definidas na Figura 11 são detalhadas abaixo:

1- Preparação do ambiente.

Nesse contexto, os usuários são convidados a participarem dos testes. Como nosso público-alvo são menores de idade, uma autorização formal dos pais pode ser vista no Apêndice A. Além da autorização formal, uma entrevista pré-teste é realizada para definir o perfil desse usuário, além de uma entrevista pós-teste, ambas disponíveis no Apêndice A deste trabalho. Para aplicação dos testes, recomenda-se um ambiente controlado, livre de interferências internas e externas e com toda atenção focada por parte do usuário final e dos observadores.

2- Observação, Comportamento e Etiquetagem.

Com o teste devidamente iniciado, é importante que o observador fique atento ao comportamento do usuário. A gravação em vídeo é um ótimo recurso para esclarecimentos futuros dessa comunicabilidade. Os usuários realizam os testes e etiquetam suas dúvidas de acordo com as expressões citadas no Quadro 2. Recomenda-se, aqui, a presença de dois observadores, de modo que a qualidade desse processo possa ser garantida com maior precisão.

3- Interpretação da Etiquetagem.

Neste momento, o avaliador analisa e contabiliza o conjunto de etiquetas aplicadas ao longo das etapas do teste, com o objetivo de avaliar a comunicabilidade. Para isso, as etiquetas devem ser avaliadas sobre os fatores: a) Tipo de falha (completa, parcial ou temporária) – b) Frequências (Número de vezes que as expressões do Quadro 2 foram utilizadas).

4- Geração do Perfil Semiótico.

Nesta última etapa, um perfil semiótico é definido com base nas tarefas anteriores realizadas e devidamente registradas. O Perfil Semiótico visa identificar as falhas de metacomunicação entre aplicação e usuário, também denominada ruptura de comunicação.

Agora que detalhamos todas as etapas do processo aplicado sobre o Método de Avaliação de Comunicabilidade, já temos uma metodologia avaliativa pronta para ser aplicada pelo avaliador ao grupo focal. Entre os diversos conceitos metodológicos e técnicas envolvidas, faz-se necessário salientar que todos os métodos e técnicas aplicadas visaram fundamentar este trabalho.

No próximo capítulo, apresentamos os resultados esperados, que foram ou não alcançados com a futura aplicação, análise e tabulação final de nossa pesquisa.

5 ANÁLISE DOS RESULTADOS

Neste tópico, os resultados obtidos com os testes aplicados em 08 crianças é apresentado, de acordo com a aplicação da metodologia descrita ao longo deste trabalho. Para avaliar a interação usuário-máquina, através da comunicabilidade referenciada pelo MAC, conforme bibliografia e embasamento teórico do trabalho, objetivou-se, com os testes, analisar e identificar as rupturas de comunicabilidade por ventura existentes.

O Método de Avaliação da Comunicabilidade (MAC) foi aplicado em um ambiente controlado, conforme sugere a bibliografia, no qual os usuários não sofreram quaisquer interferências internas e externas. Os testes foram aplicados em um laboratório de informática, cedidos pela Dwebnet Soluções Tecnológicas Ltda., com internet liberada para acesso à plataforma Scratch. Os celulares foram desligados, assim como os demais meios de comunicação.

Os testes foram realizados e monitorados, visando cumprir as 20 tarefas disponibilizadas pela plataforma, com acesso pelo *link* <https://studio.code.org/hoc/1>. As tarefas foram sequenciais e gradativas, de acordo com o andamento do usuário, pelos desafios propostos. As telas das tarefas estão disponibilizadas no Anexo A.

No quadro 8, um quadro resumo com as tarefas e ocorrências ou não de rupturas com os usuários é apresentado.

Quadro 8 – Andamento das tarefas pelos usuários

Identificação do Usuário	Ocorrência de Ruptura completa
V1	Na atividade 17
V2	Não ocorreu ruptura completa
V3	Não ocorreu ruptura completa
V4	Na atividade 17
V5	Não ocorreu ruptura completa

V6	Na atividade 17
V7	Na atividade 7
V8	Não ocorreu ruptura completa

Fonte: Elaborado pelo autor

Assim, percebe-se a ocorrência de ruptura de comunicação em 50 % dos usuários finais.

5.1 Aplicação do Método de Avaliação da Comunicação (MAC)

Os participantes foram convidados, de acordo com o perfil e público-alvo a que se destina a plataforma. Os convidados foram voluntários, e a autorização dos pais e/ou responsáveis foi obtida, visando garantir o compromisso ético e legal de nosso estudo. Os participantes obtiveram a explicação básica de como desenvolver software, conceitos do que é uma linguagem de programação, entre outros tópicos. Assim, de posse do termo de consentimento devidamente assinado, o questionário de pré-teste foi aplicado, assim como o questionário de pós-teste, logo que o usuário concluía as tarefas ou desistia do processo.

O Quadro 9, apresentado a seguir, sintetiza os voluntários e seu perfil tecnológico. Usuários foram categorizados como usuários avançados quando respondiam sim as perguntas: Envio e recebo e-mails pelo computador? ou Tenho meu próprio canal no youtube?. Essas perguntas estão disponíveis no questionário pré-teste. Os demais usuários foram considerados iniciantes.

Quadro 9 – Perfil Tecnológico dos usuários

Identificação do Usuário	Perfil Tecnológico dos usuários
V1	Iniciante
V2	Avançado
V3	Iniciante

V4	Iniciante
V5	Iniciante
V6	Avançado
V7	Iniciante
V8	Iniciante

Fonte: Elaborado pelo autor

Assim, percebe-se a ocorrência de 2 usuários considerados avançados e 6 usuários considerados iniciantes.

5.2 Análise dos dados: ambiente Scratch de programação

Com a plataforma liberada para acesso, os usuários iniciaram seu processo exploratório, e para cada tela, à medida que o mesmo navegava, o usuário etiquetava as expressões previstas em nosso quadro de etiquetagem, conforme Quadro 2. Dentre os oito usuários voluntários, identificamos seu perfil digital através de dados coletados com o pré-teste. Assim, identificamos que quatro usuários tinham grandes habilidades e destreza com recursos computacionais e quatro usuários eram vistos como usuários iniciantes, ou seja, utilizam computadores apenas para situações esporádicas ou acessam uma plataforma específica. A partir dessas informações, o perfil semiótico está sendo gerado para cada participante e apresentado a seguir.

5.2.1 Participante V1

Com o processo de etiquetagem iniciado, **o participante V1**, que tinha o perfil definido como usuário iniciante, demorou mais de 3 minutos para identificar e realizar a tarefa proposta pela etapa número 1. Nessa primeira interface, identificada como T1, o usuário cita as expressões **Por que não funciona** e a

expressão **Cadê** por 2 vezes. Sabemos que a expressão **Cadê** demonstra uma dúvida, visto que o usuário sabe o que precisa ser feito, mas não encontra de imediato esse símbolo na interface. A expressão **Por que não funciona** é aplicada. O uso dessa expressão, segundo a metodologia, os quais foram apresentados nos quadros 2 e 3, ocorre quando a operação realizada não produz o resultado esperado. Essa ruptura ocorreu quando os encaixes dos blocos de comando não produziram o resultado esperado pelo usuário em ação. Por fim, depois de 3 minutos de contato com a interface, o usuário consegue avançar para a segunda atividade, denominada T2. Nesse segundo contato, o usuário não demonstra dúvida e avança com facilidade. Na terceira atividade, T3, a expressão **Epa**, aparece por 3 vezes consecutivas. A expressão **Epa** de acordo com a metodologia, representa uma ruptura temporária, aquela em que o usuário ainda está tentando entender a lógica de funcionamento. Por fim, o usuário avança para a Interface 4; nesse momento não ocorre dificuldade, porém um aviso é exibido, demonstrando que o usuário poderia ter cumprido as atividades de um modo mais otimizado. O usuário fecha o aviso e prossegue. Nesse momento, as etiquetas de **Não, obrigado** e **Vai de Outro Jeito** são vinculadas às tarefas. Expressões desse tipo demonstram uma ruptura parcial de comunicação, não impedem o avanço e entusiasmo do usuário. Isso ocorre quando o sistema sugere um caminho, mas o usuário prefere seguir por outro meio e avança para a próxima fase. A interface 5 é vencida com facilidade, apenas com a utilização de uma expressão **Epa**. A interface 6 aparece e com ela surge um novo desafio, os blocos de repetição. Nesse momento, o usuário tem dificuldade de entender os blocos de repetição e as expressões de dúvidas tais como **Cadê**, **Por que não funciona** e **E agora** ocorrem com frequência. A ruptura de comunicação temporária ocorre com intensidade e o usuário demora mais de 5 minutos para avançar de fase. Mesmo assim, esse avanço se dá de outro modo e uma mensagem de dica do sistema aparece. Essa dica é descartada pelo usuário e a expressão **não obrigado** e **vai de outro jeito** volta a se repetir. O usuário continua avançando, mesmo tendo dificuldades de entender os blocos de repetição. O usuário chega à interface 7 e, logo em seguida, à interface 8. As expressões encontradas nessas duas interfaces são **Epa**, mas o usuário consegue avançar. Na interface 9, o usuário se vê novamente diante das expressões **Por que não funciona**, **E agora**, **Ué, o que houve**, **Assim não dá**, que demonstram falhas temporárias em que o usuário não sabe o que fazer, mas tenta descobrir. A expressão **Assim não dá** indica que o

usuário insiste em repetir encaixes e ações, mas percebe que esse caminho não é o correto. Mais uma vez ocorre uma falha temporária de comunicação. Mesmo assim, o usuário consegue avançar e chega à atividade 10. As tarefas 10, 11 e 12 são cumpridas sem dificuldades, porém as expressões **Não obrigado** e **Vai de outro jeito** são associadas, visto que essas etapas são vencidas, mas não como a plataforma esperava. Na tela 13, o usuário está mais à vontade na plataforma e consegue entender os elementos com mais facilidade. Passa pela atividade 13 sem dificuldades. A atividade 14 aparece e novos elementos como estruturas condicionais e blocos de repetição aparecem. Mais uma vez, o usuário apresenta dificuldades em entender esses novos elementos. Consegue avançar até a atividade 17, porém nas atividades 14, 15 e 16 as expressões **Assim não dá**, **Epa**, **Por que não funciona**, **Vai de outro jeito** e **Não, obrigado** aparecem em todas elas. Por fim, o usuário, na atividade 17, tenta por inúmeras vezes, mas utiliza a expressão **desisto**, ou seja, a expressão indica que o usuário não consegue cumprir a tarefa e desiste, ocorrendo, nesse momento, a ruptura completa. Ainda dentro da metodologia apurada, podemos perceber que falhas parciais, temporárias e completas ocorreram no processo de comunicação com esse usuário.

5.2.2 Participante V2

Com o processo de etiquetagem iniciado, **o participante V2**, que tinha o perfil definido como usuário avançado, demorou mais de 2 minutos para identificar e realizar a tarefa proposta pela etapa número 1. Nessa primeira interface, identificada como T1, o usuário cita as expressões **Por que não funciona** e a expressão **Epa** por 2 vezes cada. Sabemos que a expressão **Epa** demonstra uma ruptura temporária, ocorrendo quando o usuário realiza uma tarefa indesejada e tenta retornar. A expressão **Por que não funciona** é aplicada também nessa primeira interface. O uso dessa expressão, segundo a metodologia, ocorre quando a operação realizada não produz o resultado esperado. Essa ruptura ocorreu quando os encaixes dos blocos de comando não produziram o resultado esperado pelo usuário em ação. Por fim, depois de 2 minutos de contato com a interface, o usuário consegue avançar para a segunda atividade, denominada T2. Nesse segundo contato, o usuário não demonstra dúvida e avança com facilidade, porém a expressão

Epa volta a ser rotulada. Na terceira e na quarta atividades, T3 e T4, o usuário avança sem dificuldade e já apresenta uma facilidade de identificação dos elementos e seus significados. O usuário avança e a interface 5 é exibida, a expressão **Assim não dá** e **Vai de outro jeito** são etiquetadas. O usuário avança para a interface 6 e surge um novo desafio, os blocos de repetição. Nesse momento, o usuário não tem dificuldade para entender os blocos de repetição e passa sem dificuldades pela atividade. O usuário avança sem apresentar dificuldades e o mesmo entende a lógica e as estruturas de repetição. O usuário avança sem dificuldades até a atividade 16, na qual surge mais uma novidade, os blocos condicionais. A ruptura de comunicação temporária ocorre com intensidade e o usuário demora mais de 5 minutos para avançar de fase. Mesmo assim, esse avanço se dá de outro modo e uma mensagem de dica do sistema aparece. Essa dica é descartada pelo usuário e as expressões **Não, obrigado** e **Vai de outro jeito** voltam a se repetir. O usuário continua avançando, mesmo tendo dificuldades de entender os blocos condicionais. O usuário chega à interface 18 e tenta avançar por 9 vezes seguidas; as expressões **Assim não dá**, **Epa**, **Por que não funciona**, **Vai de outro jeito** e **Não, obrigado** aparecem por inúmeras vezes. A expressão **Desisto** quase foi utilizada, mas, finalmente, o usuário venceu o desafio e prosseguiu com as atividades. Por fim, após a atividade 18, as atividades 19 e 20 foram superadas sem nenhuma dificuldade por parte do usuário. Assim, com base na metodologia aplicada, podemos perceber que falhas parciais e temporárias ocorreram no processo de comunicação com esse usuário.

5.2.3 Participante V3

Com o processo de etiquetagem iniciado, **o participante V3**, que tinha o perfil definido como usuário iniciante, demorou mais de 5 minutos para identificar e realizar a tarefa proposta pela etapa número 1. Nessa primeira interface, identificada como T1, o usuário cita as expressões **Por que não funciona** e a expressão **Socorro** por 2 vezes cada. Sabemos que a expressão **Socorro** demonstra uma ruptura temporária, visto que o usuário não sabe que atitude tomar. A expressão **Por que não funciona** é aplicada também nessa primeira interface. O uso dessa

expressão, segundo a metodologia, ocorre quando a operação realizada não produz o resultado esperado. Essa ruptura ocorreu quando os encaixes dos blocos de comando não produziram o resultado esperado pelo usuário em ação, além de um tempo considerado para que o mesmo percebesse a necessidade de encaixe dos blocos. Por fim, após 5 minutos de contato com a interface, o usuário conseguiu avançar para a segunda atividade, denominada T2. A interface 2 segue com facilidade e o usuário já entende e manipula os blocos com facilidade. O usuário avança sem maiores dificuldades até a interface 5. Em seguida, o usuário avança para a interface 6 e surge um novo desafio, os blocos de repetição. Nesse momento, o usuário tem dificuldade de entender os blocos de repetição, mas, apesar disso, continua a atividade. O uso da expressão **Assim não dá** é percebida por 3 vezes consecutivas, porém o usuário avança depois de entender a lógica e as estruturas de repetição. O usuário avança sem dificuldades até a atividade 18, na qual o grau de dificuldade é maior. A ruptura de comunicação temporária ocorre com intensidade e o usuário demora mais de 5 minutos para avançar de fase. O usuário chega à interface 18, na qual tenta avançar por 7 vezes seguidas, e as expressões **Assim não dá**, **Epa**, **Por que não funciona**, **Vai de outro jeito** e **Não, obrigado** aparecem por inúmeras vezes; a expressão **Desisto** quase foi utilizada, mas finalmente o usuário venceu o desafio e prosseguiu com as atividades. Por fim, as atividades 19 e 20 são superadas sem nenhuma dificuldade por parte do usuário. Assim, com base na metodologia aplicada, podemos perceber que falhas parciais e temporárias ocorreram no processo de comunicação com esse usuário que se destacou pela motivação e expressões de desafio e entusiasmo com as atividades e os desafios apresentados pela plataforma.

5.2.4 Participante V4

Com o processo de etiquetagem iniciado, **o participante V4**, que tinha o perfil definido como usuário iniciante, não demorou mais de 1 minuto para identificar e realizar a tarefa proposta pela etapa número 1. O usuário avança para a interface 2, não demonstrando dúvida e avançando com facilidade. Na terceira atividade, T3, a expressão **Epa** aparece por 4 vezes consecutivas. A expressão **Epa**, de acordo com

a metodologia, representa uma ruptura temporária, aquela na qual o usuário ainda está tentando entender a lógica de funcionamento. Por fim, o usuário avança para a interface 4, sem dificuldade, porém um aviso é exibido, demonstrando que o usuário poderia ter cumprido as atividades de um modo mais otimizado. O usuário fecha o aviso e prossegue, mas, nesse momento, a etiqueta de **Não, obrigado** e **Vai de outro jeito** são vinculadas às tarefas. Expressões desse tipo demonstram uma ruptura parcial de comunicação, mas não impedem o avanço e entusiasmo do usuário. Isso ocorre quando o sistema sugere um caminho, mas o usuário prefere seguir por outro meio e avança para a próxima fase. O usuário avança sem dificuldades até a atividade 9, na qual se vê novamente diante das expressões **Por que não funciona, E agora, Ué, o que houve, Assim não dá** que demonstram falhas temporárias, em que o usuário não sabe o que fazer mas tenta descobrir. A expressão **Assim não dá** indica que o usuário insiste em repetir encaixes e ações, mas percebe que esse caminho não é o correto. Mais uma vez, ocorre uma falha temporária de comunicação. Mesmo assim, o usuário consegue avançar e chega à atividade 10. As tarefas 10, 11 e 12 são cumpridas sem dificuldades, porém as expressões **Não obrigado** e **Vai de outro jeito** são associadas, visto que essas etapas são vencidas, porém não como a plataforma esperava. Na tela 13, o usuário está mais à vontade na plataforma e consegue entender os elementos com mais facilidade. Passa pela atividade 13 sem dificuldades. O usuário consegue avançar mesmo com dificuldade até a atividade 16. Por fim, o usuário chega à atividade 17, na qual tenta por nove vezes consecutivas e erra todos os desafios. O usuário se irrita e utiliza a expressão **Desisto**, ou seja, a expressão indica que o usuário não conseguiu cumprir a tarefa, ocorrendo a ruptura completa. Ainda dentro da metodologia apurada, podemos perceber que falhas parciais, temporárias e completas ocorreram no processo de comunicação com este usuário.

5.2.5 Participante V5

Com o processo de etiquetagem iniciado, **o participante V5**, que tinha o perfil definido como usuário iniciante, demorou menos de 1 minuto para identificar e realizar a tarefa proposta na etapa número 1. Ao longo de seu trajeto pelas

interfaces apresentadas, a expressão **Cadê** foi utilizada nas telas apresentadas pelas tarefas das telas T1, T2 e T3. Sabemos que essa expressão demonstra uma dúvida, visto que o usuário sabe o que precisa ser feito, mas não encontra de imediato esse símbolo na interface. A expressão **Por que não funciona** foi utilizada a partir da atividade 6, em que os blocos de repetição aparecem. O uso dessa expressão, segundo a metodologia, ocorre quando a operação realizada não produz o resultado esperado. Essa ruptura ocorreu quando os encaixes dos blocos de comando não produziram o resultado esperado pelo usuário em ação. O usuário avança sem apresentar dificuldades e o mesmo entende a lógica e as estruturas de repetição. O usuário avança sem dificuldades até a atividade 16, quando surge mais uma novidade, os blocos condicionais. O usuário demora um tempo de aproximadamente 3 minutos e os meios alternativos de passar pela atividade são tentados, caracterizando o uso da expressão **Vai de outro jeito**. O usuário avança pelas demais atividades sem apresentar nenhum tipo de dificuldade. Esses avanços seguidos e sem rupturas completas nos levam a perceber que o usuário já tinha uma familiaridade com a plataforma, o que foi confirmado posteriormente através de uma entrevista informal pós-teste.

5.2.6 Participante V6

Com o processo de etiquetagem iniciado, **o participante V6**, que tinha o perfil definido como usuário avançado, demorou mais de 2 minutos para identificar e realizar a tarefa proposta pela etapa número 1. Nessa primeira interface, identificada como T1, o usuário cita a expressão **Cadê**, por 2 vezes. Sabemos que a expressão **Cadê** demonstra uma dúvida, visto que o usuário sabe o que precisa ser feito, mas não encontra de imediato esse símbolo na interface. Por fim, depois de 2 minutos de contato com a interface, o usuário consegue avançar para a segunda atividade, denominada T2. Nesse segundo contato, o usuário não demonstra dúvida e avança com facilidade. Na terceira atividade, T3, a expressão **Epa**, aparece por 3 vezes consecutivas. A expressão **Epa**, de acordo com a metodologia, representa uma ruptura temporária, aquela em que o usuário ainda está tentando entender a lógica de funcionamento. As atividades 4 e 5 são vencidas com facilidade, apenas com a

utilização de uma expressão **Epa**. A interface 6 aparece e com ela surge um novo desafio: os blocos de repetição. Nesse momento, o usuário tem dificuldade de entender os blocos de repetição e as expressões de dúvidas, tais como **Cadê**, **Por que não funciona** e **E agora**, ocorrem com frequência. A ruptura de comunicação temporária ocorre com intensidade e o usuário demora mais de 5 minutos para avançar de fase. Mesmo assim, esse avanço se dá de outro modo e uma mensagem de dica do sistema aparece. Essa dica é descartada pelo usuário e a expressão **Não, obrigado** e **Vai de outro jeito** volta a se repetir. O usuário continua avançando, mesmo tendo dificuldades de entender os blocos de repetição. O usuário chega à interface 7 e, logo em seguida, à interface 8. As expressões encontradas nessas duas interfaces são **Epa**, mas o usuário consegue avançar. Na interface 9, o usuário se vê novamente diante das expressões **Por que não funciona**, **E agora**, **Ué, o que houve** e **Assim não dá**. Essas expressões demonstram falhas temporárias, em que o usuário não sabe o que fazer, mas tenta descobrir. A expressão **Assim não dá** indica que o usuário insiste em repetir encaixes e ações, mas percebe que esse caminho não é o correto. Mais uma vez, ocorre uma falha temporária de comunicação. Mesmo assim, o usuário consegue avançar e chega à atividade 10. As tarefas 10, 11, 12, 13 e 14 são cumpridas sem dificuldades, porém as expressões **Não, obrigado** e **Vai de outro jeito** são associadas, visto que essas etapas são vencidas, porém não como a plataforma esperava. O usuário consegue avançar até a atividade 17 e tenta por 7 vezes, mas utiliza a expressão **Desisto**, ou seja, a expressão indica que o usuário não consegue cumprir a tarefa, ocorrendo, nesse momento a ruptura completa. Ainda dentro da metodologia apurada, podemos perceber que falhas parciais, temporárias e completas ocorreram no processo de comunicação com esse usuário.

5.2.7 Participante V7

Com o processo de etiquetagem iniciado, **o participante V7**, que tinha o perfil definido como usuário iniciante, demorou mais de 3 minutos para identificar e realizar a tarefa proposta na etapa número 1. Nessa primeira interface, identificada como T1, o usuário cita a expressão **Socorro** por 2 vezes cada. Sabemos que a

expressão **Socorro** demonstra uma ruptura temporária, visto que o usuário não sabe que atitude tomar. A expressão **Por que não funciona** é aplicada também nessa primeira interface. O uso dessa expressão, segundo a metodologia, ocorre quando a operação realizada não produz o resultado esperado. Essa ruptura ocorreu quando os encaixes dos blocos de comando não produziram o resultado esperado pelo usuário em ação, além de um tempo considerado para que o mesmo percebesse a necessidade de encaixe dos blocos. O usuário avança sem maiores dificuldades até a Interface 5. Em seguida, o usuário avança para a interface 6, quando surge um novo desafio: os blocos de repetição. Nesse momento, o usuário tem dificuldade de entender os blocos de repetição, mas continua a atividade. Ele não consegue avançar e, na atividade 7, depois de 7 tentativas frustradas, desiste da operação e a expressão **Desisto** é aplicada, ou seja, a expressão indica que o usuário não consegue cumprir a tarefa, ocorrendo a ruptura completa. Ainda dentro da metodologia apurada, podemos perceber que falhas parciais, temporárias e completas ocorreram no processo de comunicação com esse usuário. Um fato que chamou a atenção, além da pouca evolução das interfaces desse usuário, é que ele se encontra dentro do limite inferior de idade, ou seja, acabou de completar 8 anos de idade. Isso pode ter contribuído para a pouca evolução e entendimento das atividades propostas pela interface.

5.2.8 Participante V8

Enfim, chegamos ao nosso voluntário final e iniciamos o processo de etiquetagem com **o participante V8** que tinha o perfil definido como usuário iniciante. Ele não demorou mais de 3 minutos para identificar e realizar a tarefa proposta pela etapa número 1. O usuário avançou para a interface 2, não demonstrando dúvida e avançando com facilidade. Na terceira atividade, T3, a expressão **Epa** apareceu por 4 vezes consecutivas. A expressão **Epa**, de acordo com a metodologia, representa uma ruptura temporária, aquela em que o usuário ainda está tentando entender a lógica de funcionamento. Por fim, o usuário avança para a Interface 4, sem dificuldade, porém um aviso é exibido, demonstrando que o usuário poderia ter cumprido as atividades de um modo mais otimizado. O usuário fecha o aviso e prossegue, e as etiquetas **Não, obrigado** e **Vai de outro jeito** são

vinculadas às tarefas. Expressões desse tipo demonstram uma ruptura parcial de comunicação, mas não impedem o avanço e o entusiasmo do usuário. Isso ocorre quando o sistema sugere um caminho, mas o usuário prefere seguir por outro meio e avança para a próxima fase. O usuário avança sem dificuldades até que chega à atividade 10. As tarefas 11 e 12 são cumpridas sem dificuldades, porém as expressões **Não, obrigado** e **Vai de outro jeito** são associadas, visto que essas etapas são vencidas, porém não como a plataforma esperava. Na tela 13, o usuário está mais à vontade na plataforma e consegue entender os elementos com mais facilidade. O usuário passa pela atividade 13 sem dificuldades, consegue avançar, mesmo com dificuldade, até a atividade 16 e, por fim, chega à atividade 17, tentando por quatro vezes consecutivas e avançando por caminhos alternativos. O usuário continua avançando, mesmo tendo dificuldades de entender os blocos condicionais. O usuário chega à interface 18, tenta avançar por 9 vezes seguidas e as expressões **Assim não dá**, **Epa**, **Por que não funciona**, **Vai de outro jeito** e **Não, obrigado** aparecem por inúmeras vezes. Por fim, as atividades 19 e 20 são superadas sem nenhuma dificuldade por parte do usuário. Assim, com base na metodologia aplicada, podemos perceber que falhas parciais e temporárias ocorreram no processo de comunicação com esse usuário.

Enfim, com a apresentação do perfil semiótico individualizado devidamente tratado, inúmeras conclusões são apresentadas na próxima seção.

6 CONSIDERAÇÕES FINAIS

Diante deste trabalho investigativo e através do levantamento do perfil semiótico individualizado, uma série de considerações merecem destaque. Nesta dissertação, identificamos a ruptura de comunicação através da aplicação do Método de Avaliação da Comunicabilidade, com base nos pressupostos teóricos propostos pela metodologia. A investigação de rupturas de comunicação entre usuários finais e aplicação demonstra como a metodologia apura o perfil semiótico de cada usuário e qual é sua interpretação pessoal da comunicabilidade apresentada pela aplicação.

Métodos de avaliação de interfaces envolvem testes aplicados ao público-alvo específico, em especial em nosso trabalho, crianças com a faixa etária de 8 a 12 anos de idade. A aplicação em um grupo focal visa ouvir um grupo de pessoas para falar de problemas que interessam ao pesquisador.

É importante ressaltar que, apesar das dificuldades investigativas, a aplicação direta de testes de interface com o público-alvo nos leva a muitas respostas e situações muitas vezes inesperadas pelo avaliador ou pelo *designer* da aplicação. Essas reações do público-alvo motivaram nosso trabalho e justificaram a aplicação do Método de Avaliação da Comunicabilidade. Em virtude dessa investigação destinada ao grupo focal, percebeu-se que a ruptura de comunicação ocorreu em todos os usuários, porém em quatro deles a ruptura norteou as três categorias definidas por De Souza e Leitão (2009) no Quadro 3, ou seja, apresentou rupturas de comunicação parcial, temporária e completa. Uma ligação direta pode ser indicada quando quatro usuários apresentaram todas as rupturas de comunicação: eram usuários considerados iniciantes no uso de computadores.

A ruptura de comunicação do tipo parcial ocorreu em todos os voluntários, sendo que a predominância das expressões: **Não, obrigado** e **Vai de outro jeito** foram as mais utilizadas pelos usuários. Essa etiquetagem foi marcada pelo avaliador quando dicas eram exibidas ao entrar na tarefa, mas o usuário simplesmente fechava a tela, aplicando, assim, o uso da expressão **Não, obrigado**, sem levar a dica sugerida pela aplicação em consideração. Esse tipo de ação era etiquetado pelo avaliador como **Não, obrigado**, por se enquadrar na situação: O usuário faz a opção de um caminho alternativo, mesmo o sistema sugerindo um caminho a seguir, definido pela metodologia. Outro aspecto considerado era o uso da expressão **Vai de outro jeito**, que ocorria quando o usuário, mesmo sendo orientado pela aplicação a utilizar, por exemplo, blocos condicionais e de repetição, preferiam tentar blocos sequenciais e com um pensamento lógico padrão.

Em relação à ruptura de comunicação, a **do tipo temporária** ocorreu em todos voluntários, sendo que a expressão **Assim não dá!** e **Por que não funciona?** as mais sinalizadas pelos usuários. Desse modo, essa ruptura nos leva a definir o cenário de dúvida, que ocorre quando o usuário sabe o que deve fazer, porém o resultado produzido não está em conformidade com o resultado esperado, além do cenário de questionamento. Esse tipo de ruptura ocorreu em usuários do tipo iniciante e em avançados.

Por fim, a ruptura de comunicação **do tipo completa** foi identificada em quatro voluntários, que utilizaram a expressão **Desisto**, ou seja, deixaram de percorrer até o final das 20 etapas. A tarefa que teve maior incidência sobre a ruptura completa ocorreu na atividade 17, que envolve uma mudança de raciocínio lógico sobre a aplicação.

Do ponto de vista global, podemos dizer que a emissão das mensagens pela aplicação foram, em sua maioria, identificadas e devidamente interpretadas pelos usuários finais. A ruptura de comunicação também foi percebida, porém, por se tratar de um ambiente e de uma disciplina ainda pouco explorada, tais como o raciocínio lógico e a linguagem de programação, essas rupturas podem servir como alertas para o desenvolvimento de melhorias no processo de comunicação e relação de transformação de ensino-aprendizagem. Outro ponto que merece destaque é que

essa alta incidência de ruptura completa pode indicar a necessidade de algum instrutor para continuidade das atividades e da linha de raciocínio dos usuários finais.

Assim, nossa proposta de avaliar o perfil semiótico dos usuários e de identificar rupturas de comunicação da plataforma Scratch com os usuários pode ser observada e avaliada. Os usuários, após a aplicação dos questionários pós-testes foram unânimes em responder positivamente à questão “Você gostaria de aprender linguagem de programação”. Outros pontos foram detectados como positivos, tais como:

- Ambiente lúdico mostrou gratas supresas aos usuários finais – uma correlação direta com as três fases da experiência peirceana.
- Para sete dos oito voluntários, a plataforma estava sendo acessada e conhecida pela primeira vez.
- Todos os voluntários ficaram curiosos e instigados a aprofundar e conhecer um pouco mais sobre o potencial das linguagens de programação.

Sabemos que essa pesquisa tem uma limitação de escopo e que toda pesquisa que envolve interação homem-máquina tem caráter subjetivo. Apesar desse caráter subjetivo e limitando-se ao nosso grupo focal, conclusões foram detectadas, mas, mesmo assim, sabemos que as conclusões não podem ser generalizadas e tomadas como verdades únicas.

Portanto, esperamos que este trabalho de pesquisa possa contribuir para um embasamento teórico-prático sobre a Engenharia Semiótica e a comunicabilidade avaliada pela mesma sobre esse cenário específico.

Como trabalhos futuros, a comparação de mais de uma plataforma e a reaplicação desse estudo em diferentes grupos socioeconômicos contribuirão de maneira positiva para os desenvolvedores de interfaces de comunicação.

REFERÊNCIAS

- ALMEIDA, C. C. de. *Peirce e a Organização da Informação: contribuição teórica da Semiótica e do Pragmatismo*. 2009. 418f. Tese (Doutorado). Marília: Universidade Estadual Paulista – UNESP. Campus de Marília. Faculdade de Filosofia e Ciências, São Paulo.
- D'ALVIELLA, C. G. *A migração dos símbolos*. Tradução de Hebe Way Ramos e Newton Roberval Eichenberg. São Paulo: Pensamento, 1995.
- DALL'AGNOL, C. M.; TRENCH, M. H. Grupos focais como estratégia metodológicas em pesquisa na enfermagem. *Rev. Gaúcha Enf*, Porto Alegre, V. 20, n. 1, p. 5-25, 1999.
- DAVENPORT, T. H.; PRUSAK, L. *Conhecimento empresarial: como as organizações gerenciam seu capital intelectual*. Rio de Janeiro: Campus, 1998.
- DEBUS M. *Manual para excelência en la investigación mediante grupos focales*. Washington: Academy for Educational Development, 1997.
- DERVIN, B. User as research inventions: how research categories perpetuate inequities. *Journal of Communication*, v. 39, n. 3, p. 216-32, 1989.
- DE SOUZA, Clarisse Sieckenius. *The Semiotic Engineering of Human-Computer Interaction*. Cambridge: The MIT Press, 2005.
- DE SOUZA, C. S. de; LEITÃO, C. F. *The Semiotic Engineering Methods for Scientific Research in HCI*. Califórnia: Morgan and Claypool Publishers, 2009.
- DIAS, Cláudia Augusto. Portal corporativo: conceitos e características. *Revista Ciência da Informação*, v. 30, n. 1, p. 5060, jan./abr. 2001.
- DOHERTY, N. F., COOMBS, C. R.; LOAN-CLARKE, J. A re-conceptualization of the interpretive flexibility of information technologies: redressing the balance between the social and technical. *European Journal of Information Systems*, vol. 15(6), p. 569-582, 2006.
- EBOLI, Marisa. *Coletânea Universidades Corporativas: educação para as empresas do século XXI*. São Paulo: Schumkler Editores, 1999.
- ECO, U. *A theory of semiotics*. *Bloomington*: Indiana University Press, 1976.
- ECO, Umberto. *Conceito de Texto*. São Paulo: Editora Edusp, 1984.
- ECO, Umberto. *Tratado Geral de Semiótica*. São Paulo: Editora Perspectiva, 1997.
- FARRER, H. *Pascal estruturado*. Rio de Janeiro: LTC, 1999.
- HUIZINGA, J. *Homo Ludens – vom Unprung der Kultur im Spiel*. Tradução: João Paulo Monteiro. Revisão Mary Amazonas Leite de Barros. Produção: Ricardo W. Neves e Adriana Garcia. São Paulo: Editora Perspectiva, 2000.

IERVOLINO, S. A.; PELICIONE, M. C. A utilização do grupo focal como metodologia qualitativa na promoção da saúde. *Rev. Esc. Enf. USP*. São Paulo, v. 35, n. 2, p. 115-121, jun., 2001.

JAKOBSON, Roman. *Lingüística e comunicação*. São Paulo: Cultrix, 1973.

KLEIN, H. K.; MYERS, M. D. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*, p. 67-93, 1999.

LAUDON, Kenneth C., LAUDON, Jane Price. *Sistemas de informação: com Internet*. Tradução de Dalton Conde de Alencar. 4. ed. Rio de Janeiro: LTC, 1999.

LEITE, L. L. *Colaboração entre pares para melhor compreensão da mensagem do designer via sistema de ajuda*. 2012. 117f. Tese (Doutorado). Fac. de Informática, PUCRS, Porto Alegre.

LIN, A.; SILVA, L. The social and political construction of technological frames. *European Journal of Information System*, v. 14(1), p. 49-59, 2005.

MALHOTRA, N. K. *Pesquisa de marketing: uma orientação aplicada*. 4. ed. Porto Alegre: Bookman, 2006.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. *Estudo dirigido de algoritmos*. 3. ed. São Paulo: Érica, 1997.

MATTOS, B. A. M. *Uma Extensão do Método de Avaliação de Comunicabilidade para sistemas colaborativos*. 2010. 170f. Dissertação (Mestrado). DCC/UFMG, Belo Horizonte.

MEIER, M. J.; KUDLOWIEZ, S. Grupo Focal: Uma experiência singular. *Texto e Contexto Enf*, Florianópolis, v. 12, n. 3, p. 394-399, 2003.

MICROSOFT – Kodu. Disponível em: <<http://www.kodugamelab.com/>>. Acesso em: 09 out. 2014.

MURRAY, P.; MYERS, A. The facts about knowledge, *Information Strategy*, 2(7), p. 29-33. Murugesan. S. Boston, MA: Butterworth-Heinemann, 1997.

NANDHAKUMAR, J.; JONES, M: Development gain? Participant observation management information systems research. *Qualitative Research*, 2(3), p. 323-341, 2002.

NIELSEN, J.; LORANGER, H. *Usabilidade na Web: Projetando Websites com qualidade*. Rio de Janeiro: Editora Campus, 2006.

NETTO, J. Teixeira Coelho. *Semiótica, informação e comunicação*. São Paulo: perspectiva, 2003.

NONAKA, Ikujiro; TAKEUCHI, Hirotaka. *Criação de conhecimento na empresa*. 13ª reimpressão. Rio de Janeiro: Elsevier, 1997.

NONAKA, I.; TAKEUCHI, H. *Criação de conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação*. 4. ed. Rio de Janeiro: Campus, 1997. 358 p.

- NONAKA, I. *Gestão do conhecimento*. Porto Alegre: Bookman, 2008.
- NONAKA, I.; TAKEUCHI, H. *The knowledge-creating company*. Oxford: Oxford University Press, 1995.
- O'BRIEN, James A. *Sistemas de informação e as decisões gerenciais na era da Internet*. Tradução de Cid Knipel Moreira. 9. ed. São Paulo: Saraiva, 2002.
- OLIVEIRA, V. G. de; PORROZZI, R. Possibilidades e limitações da informática na educação. *Revista Práxis*, v. 1, n. 1, 1999.
- ORLIKOWSKI, W. J.; GASH, D. C. Technological frames: making sense of information technology in organizations. *Organization science*, v. 11, n. 4, p. 404-428, 2000.
- PEIRCE, C. S. *Collected papers of Charles S. Peirce*. Edited by Hartshorne, Weiss and Burks. Cambridge, MA: Harvard University Press, 1958, 1974. v. 1-8.
- PEIRCE, C. S. *Semiótica*. Trad. José Teixeira Coelho Neto. São Paulo: Perspectiva, 1977.
- PEIRCE, C. S. *Semiótica*. 2. ed. São Paulo: Perspectiva, 1995.
- PEIRCE, C. S. *Semiótica*. 3. ed. São Paulo: Editora Perspectiva, 2005.
- PEIRCE, C. S. *The new elements of mathematics*. Edited by Carolyn Eisele. The Hague: Mouton, 1976. v. 1-4.
- PRATES, R. O.; BARBOSA, S. D. J. *Introdução a teoria e a prática da interação humano computador fundamentada na engenharia semiótica*. 2007. Disponível em <http://www2.serg.inf.puc-rio.br/docs/JAI2007_PratesBarbosa_EngSem.pdf>. Acesso em: 10 ago. 2014.
- PRATES, Raquel Oliveira; De SOUZA, Clarisse Sieckenous; BARBOSA, Simone Diniz Junqueira. *A method for evaluating the communicability of user interfaces*. New York: ACM Interactions, 2000.
- PREECE, Jenny; ROGERS, Yvonne; SHARP, Helen. *Interaction Design: Beyond Human-computer Interaction*. New York: John Wiley & Sons, 2002.
- PROJETO CODE ORG. Disponível em: <<https://code.org>>. Acesso em: 22 maio 2015.
- SALMAZO, P. R. *Um estudo em gestão do conhecimento para uma empresa de tecnologia da informação*. Porto Alegre: Programa de Pós-Graduação, 2004.
- SALVETTI, D. D.; BARBOSA, L. M. *Algoritmos*. São Paulo: Pearson Education/Makron Books, 1999.
- SANTAELLA, L. *O que é Semiótica*. Coleção primeiros passos: 103. São Paulo: Editora Brasiliense, 1999.
- SCRATCH. Disponível em: <<http://scratch.mit.edu/>>. Acesso em: 21 jul. 2015.

SILVEIRA, Milene Selbach; DE SOUZA, Clarisse Sieckenius; BARBOSA, Simone. Um Método da Engenharia Semiótica para a Construção de Sistema de Ajuda Online. In: LATIN AMERICAN CONFERENCE ON HUMAN-COMPUTER INTERACTION. Rio de Janeiro, 17 ago. 2003.

SVEIBY, Karl Erik. *A Nova Riqueza das Organizações – gerando e avaliando patrimônios de conhecimento*. Rio de Janeiro: Campus, 1998.

TAKEUCHI, H.; NONAKA, I. Criação e dialética do conhecimento. In: TAKEUCHI, H.; NONAKA, I. *Gestão do conhecimento*. Porto Alegre: Bookman, 2008. p. 17-38.

VERGARA, Sylvia Constant. *Projetos e relatórios de pesquisa em administração*. 5. ed. São Paulo: Atlas, 2004.

WALSH, J. P. Managerial and organizational cognition: notes from a trip down memory lane. *Organization Science*, v. 6, n. 3, p. 280-321, 1995.

WALSHAM, G. Doing interpretive research. *European Journal of Information Systems*. v. 15, n. 3, p. 320-330, 2006.

WILSON, T. D. Recent Trends in user studies: action research and qualitative methods. *Information Research*, v. 5, n. 3, s. p., Apr 2000.

W3C. Disponível em: <<http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-I.html>>. Acesso em: 24 out. 2015.

YIN, R. K. *Case study research: design and methods*. London: Thousand Oaks, 2009.

YOUNG, R. *Knowledge Management Tools and Techniques Manual*. Tokyo: Asian Productivity Organization, 2010.

APÊNDICE A – DOCUMENTOS E TERMOS UTILIZADOS PARA OS TESTES

Termo de autorização de participação

Título: Avaliação da comunicabilidade do ambiente de programação Scratch

Usuário:

Data de Nascimento:

Idade atual:

Nome do pai ou responsável:

Pesquisador responsável pela aplicação do teste:

Rodrigo de Matos Vargas

Por meio do aceite e assinatura deste termo, os pais ou representantes legais autorizam seu filho, menor de idade, com faixa etária entre 08 e 12 anos, a realizarem os testes de comunicabilidade para avaliação do ambiente de programação Scratch.

Para entendimento dos pais ou responsáveis, o usuário devidamente autorizado irá acessar o ambiente de programação Scratch, disponível no *site* www.scratch.com, e realizará as tarefas relacionadas a raciocínio lógico e linguagem de programação propostas pelo ambiente estudado. Todo o acesso aos equipamentos e ao conteúdo acessado serão de responsabilidade do pesquisador.

Os dados coletados durante os testes serão utilizados na avaliação da comunicabilidade através do Método de Avaliação da Comunicabilidade (MAC). Todos os usuários são voluntários e estão cientes de que não haverá qualquer tipo de premiação ou remuneração de qualquer espécie por sua participação.

Caso deseje maiores informações e/ou esclarecimentos, fineza enviar um e-mail para rmatos@rmatos.com.br ou contatar através do telefone: (31) 9977-6860.

Estando de comum acordo e ciente das atividades, autorizo a participação de meu(minha) filho(a) nessa atividade.

Assinatura do responsável legal:

Questionário Pré-teste

Nome do usuário:

Data de Nascimento:

Sexo: () Masculino () Feminino

1- Você sabe o que é linguagem de programação?

() Sim () Não

2- Você sabe o que é um software?

() Sim () Não

3- Você utiliza computadores ou notebooks?

() Sim () Não

4- Você utiliza tablets ou videogames?

() Sim () Não

5- Você utiliza computadores com qual objetivo? Pode marcar mais de uma opção:

() Tenho meu próprio perfil no facebook e o uso com frequência.

() Envio e recebo e-mails pelo computador.

() Utilizo o computador apenas para jogos.

() Tenho meu próprio canal no youtube.

Questionário Pós-teste

Nome do usuário:

Data de Nascimento:

Sexo: () Masculino () Feminino

1- Você sabe o que é linguagem de programação?

() Sim () Não

2- Você sabe o que é um software?

() Sim () Não

3- Você acha que esse ambiente foi divertido e de fácil utilização?

() Sim () Não

4- Você gostou das atividades propostas pelo ambiente?

() Sim () Não

5- Você tem interesse em aprender mais sobre linguagens de programação?

() Sim () Não

APÊNDICE B – RESUMO DO PERFIL DOS VOLUNTÁRIOS

Identificação	Definição do perfil tecnológico	Ruptura completa	Idade
V1	Iniciante	Na atividade 17	11 anos
V2	Avançado	Não ocorre	11 anos
V3	Iniciante	Não ocorre	12 anos
V4	Iniciante	Na atividade 17	11 anos
V5	Iniciante	Não ocorre	12 anos
V6	Avançado	Na atividade 17	10 anos
V7	Iniciante	Na atividade 7	8 anos
V8	Iniciante	Não ocorre	12 anos

Amostra de 08 voluntários, sendo 06 usuários do tipo Iniciante e dois do tipo avançado.

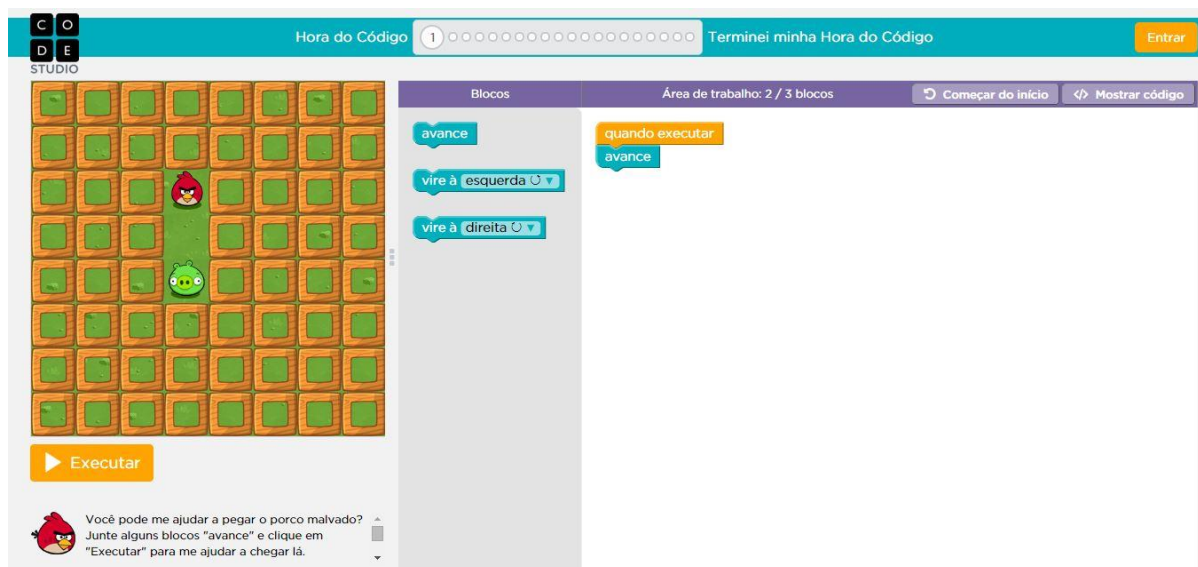
Em quatro voluntários, ocorre ruptura completa.

Em quatro voluntários, não ocorre ruptura completa.

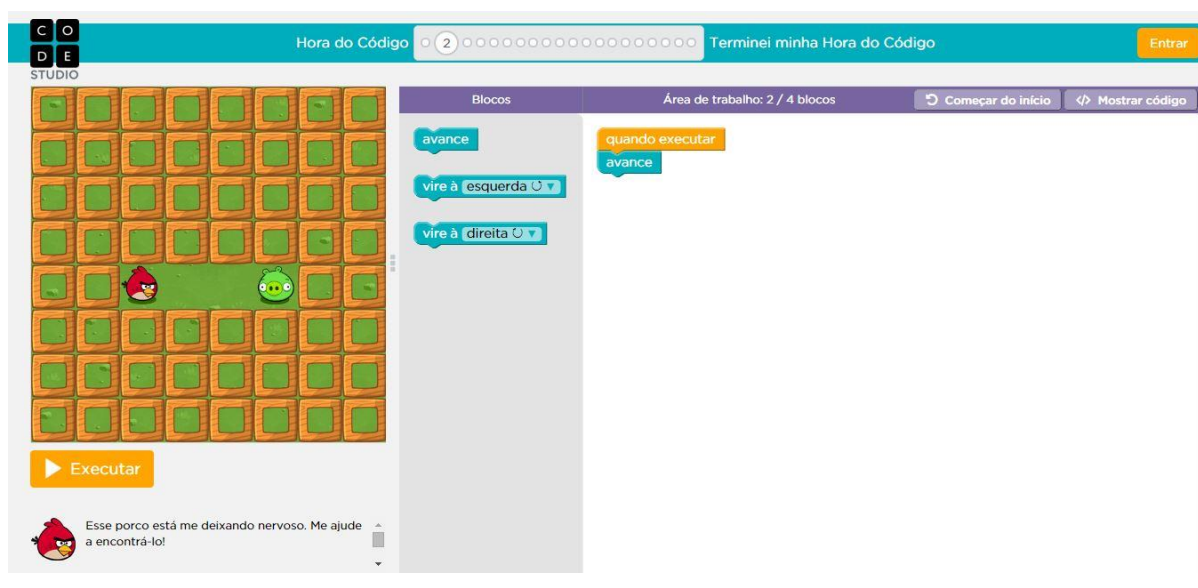
Todos os voluntários estão dentro da faixa etária proposta pela plataforma.

ANEXO A – TELAS DE ACESSO À PLATAFORMA

Tela 1 – <https://studio.code.org/hoc/1>



Tela 2 – <https://studio.code.org/hoc/2>



Tela 3 – <https://studio.code.org/hoc/3>

STUDIO

Hora do Código 3 Terminar minha Hora do Código Entrar

Blocos Área de trabalho: 2 / 5 blocos Começar do início Mostrar código

avance

vire à esquerda

vire à direita

quando executar

avance

▶ Executar

Desenhe o caminho e leve-me até aquele porco bobalhão. Não passe pelas dinamites, senão penas vão voar!

Tela 4 – <https://studio.code.org/hoc/4>

STUDIO

Hora do Código 4 Terminar minha Hora do Código Entrar

Blocos Área de trabalho: 2 / 6 blocos Começar do início Mostrar código

avance

vire à esquerda

vire à direita

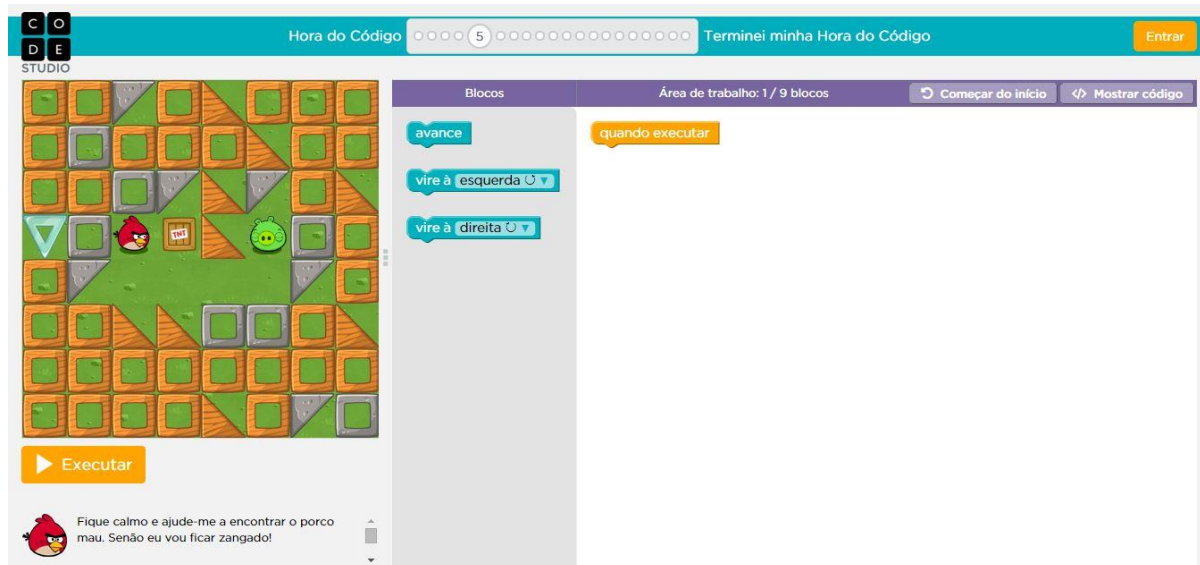
quando executar

avance

▶ Executar

Leve-me até a maldade verde! (Cuidado com a dinamite)

Tela 5 – <https://studio.code.org/hoc/5>



C O
D E

Hora do Código 5 Terminei minha Hora do Código Entrar

STUDIO

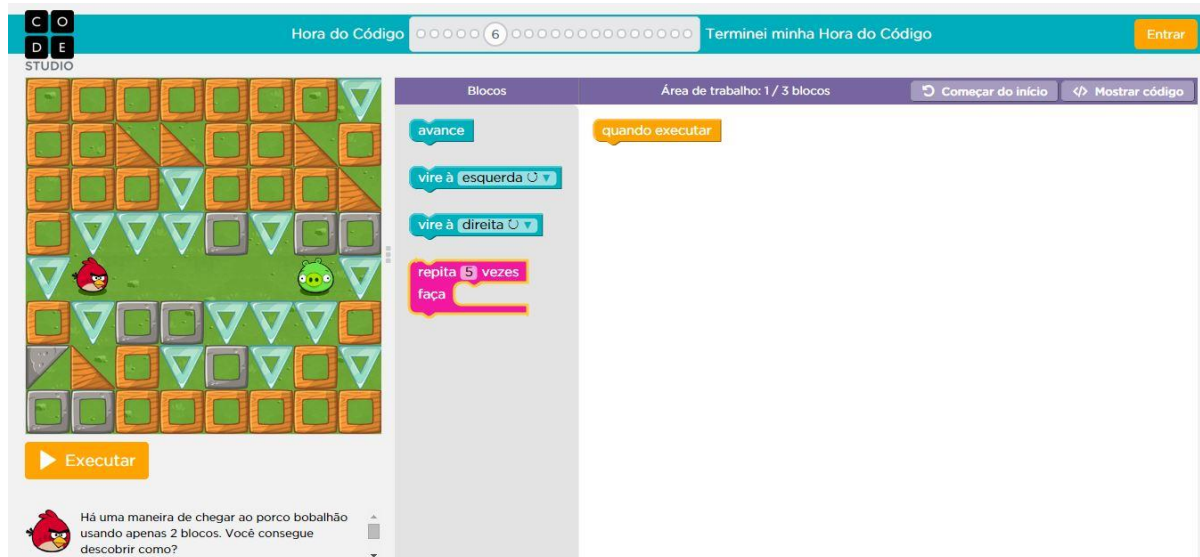
Blocos Área de trabalho: 1 / 9 blocos Começar do início Mostrar código

avance
quando executar
vire à esquerda
vire à direita

▶ Executar

Fique calmo e ajude-me a encontrar o porco mau. Senão eu vou ficar zangado!

Tela 6 – <https://studio.code.org/hoc/6>



C O
D E

Hora do Código 6 Terminei minha Hora do Código Entrar

STUDIO

Blocos Área de trabalho: 1 / 3 blocos Começar do início Mostrar código

avance
quando executar
vire à esquerda
vire à direita
repita 5 vezes
faça

▶ Executar

Há uma maneira de chegar ao porco bobalhão usando apenas 2 blocos. Você consegue descobrir como?

Tela 7 – <https://studio.code.org/hoc/7>

The screenshot shows the Code.org Hour of Code interface for level 7. At the top, the text "Hora do Código" is followed by a progress bar with 10 circles, the 7th of which is filled. To the right, it says "Terminei minha Hora do Código" and has an "Entrar" button. Below this is the "STÚDIO" header. The main area is divided into two panels. The left panel shows a 10x10 grid of blocks: green squares, brown squares, and grey triangles. A red bird is at the top center, and a green pig is at the bottom center. Below the grid is an "Executar" button. The right panel is the "Blocos" (Blocks) area, showing a "quando executar" (when executed) block connected to a script containing: "avance" (move forward), "vire à esquerda" (turn left), "vire à direita" (turn right), and "repita 5 vezes faça" (repeat 5 times do). Below the grid, a small instruction reads: "Tente me levar até o intruso verde usando apenas três blocos."

Tela 8 – <https://studio.code.org/hoc/8>

The screenshot shows the Code.org Hour of Code interface for level 8. At the top, the text "Hora do Código" is followed by a progress bar with 10 circles, the 8th of which is filled. To the right, it says "Terminei minha Hora do Código" and has an "Entrar" button. Below this is the "STÚDIO" header. The main area is divided into two panels. The left panel shows a 10x10 grid of blocks: green squares, brown squares, and grey triangles. A green pig is at the top center, and a red bird is at the bottom center. Below the grid is an "Executar" button. The right panel is the "Blocos" (Blocks) area, showing a "quando executar" (when executed) block connected to a script containing: "avance" (move forward), "vire à esquerda" (turn left), "vire à direita" (turn right), and "repita 5 vezes faça" (repeat 5 times do). Below the grid, a small instruction reads: "Ajude-me a banir esse porquinho mau usando o menor número de blocos possível. Tente usar mais de um bloco 'repita'."

Tela 9 – <https://studio.code.org/hoc/9>

STUDIO

Hora do Código 9 Terminei minha Hora do Código Entrar

Blocos Área de trabalho: 2 / 5 blocos Começar do início Mostrar código

avance

vire à esquerda

vire à direita

repita 3 vezes

faça

quando executar

repita 3 vezes

faça

Os blocos cinza não podem ser excluídos. Você consegue resolver o desafio assim mesmo?

▶ Executar

Quando um bloco estiver cinza, significa que você não pode excluí-lo. Resolva esse desafio usando o bloco "repita" que faz 3 repetições.

Tela 10 – <https://studio.code.org/hoc/10>

STUDIO

Hora do Código 10 Terminei minha Hora do Código Entrar

Blocos Área de trabalho: 1 / 3 blocos Começar do início Mostrar código

avance

vire à esquerda

vire à direita

repita até

faça

quando executar

repita até

faça

Ok, use o novo bloco "repita até" - ele vai se repetir até que eu chegue naquele porco irritante.

▶ Executar

Tela 11 – <https://studio.code.org/hoc/11>

C O
D E

Hora do Código 11 Terminei minha Hora do Código Entrar

STUDIO

Blocos Área de trabalho: 1 / 5 blocos Começar do início Mostrar código

avance quando executar

vire à esquerda

vire à direita

repita até

faça

▶ Executar

Ok, pratique mais uma vez - você pode resolver isso usando apenas 4 blocos?

Tela 12 – <https://studio.code.org/hoc/12>

C O
D E

Hora do Código 12 Terminei minha Hora do Código Entrar

STUDIO

Blocos Área de trabalho: 1 / 6 blocos Começar do início Mostrar código

avance quando executar

vire à esquerda

vire à direita

repita até

faça

▶ Executar

Pessoa querida. Mim zumbi. Mim com fome. Devo... chegar... no girassol... Você pode me ajudar a chegar lá com apenas 5 blocos?

Tela 13 – <https://studio.code.org/hoc/13>

STUDIO

Hora do Código 13 Terminei minha Hora do Código Entrar

Blocos Área de trabalho: 1 / 6 blocos Começar do início Mostrar código

avance
vire à esquerda
vire à direita
repita até
faça

quando executar

▶ Executar

Ok, é parecido, mas um pouco diferente. Você pode usar apenas 5 blocos?

Tela 14 – <https://studio.code.org/hoc/14>

STUDIO

Hora do Código 14 Terminei minha Hora do Código Entrar

Blocos Área de trabalho: 4 / 5 blocos Começar do início Mostrar código

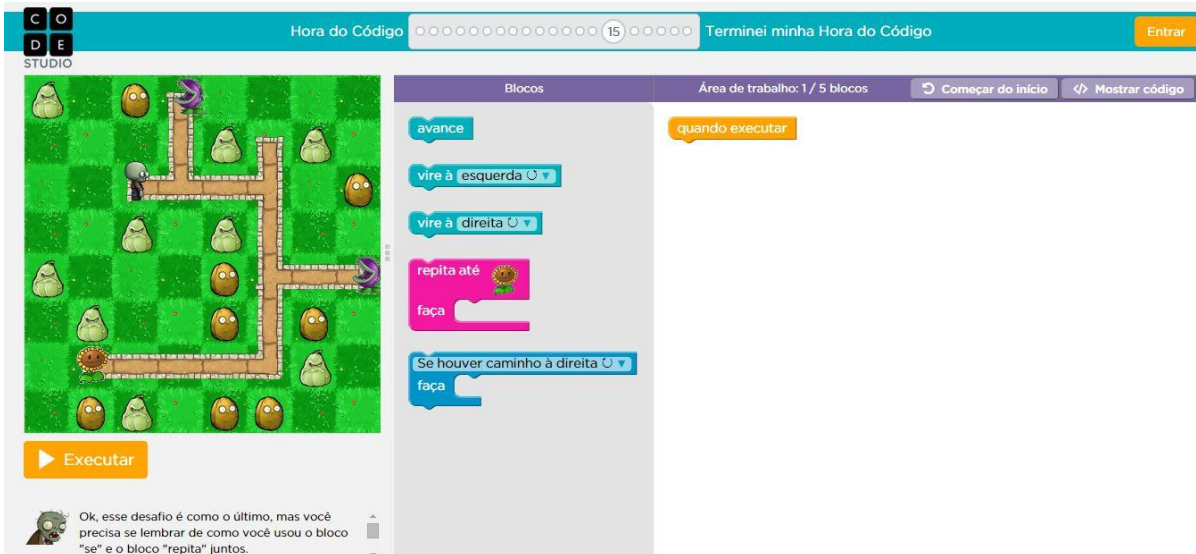
avance
vire à esquerda
vire à direita
repita até
faça

quando executar
repita até
faça
 avance
 Se houver caminho à esquerda
faça

Se houver caminho à esquerda
faça

▶ Executar

Use o novo bloco "se" para me deixar decidir quando virar. Dica: você só precisa de mais um bloco, mas veja como o configuramos, então

Tela 15 – <https://studio.code.org/hoc/15>

C O
D E

Hora do Código 15 Terminei minha Hora do Código Entrar

STUDIO

Blocos Área de trabalho: 1 / 5 blocos Começar do início Mostrar código

quando executar

avance

vire à esquerda

vire à direita

repita até

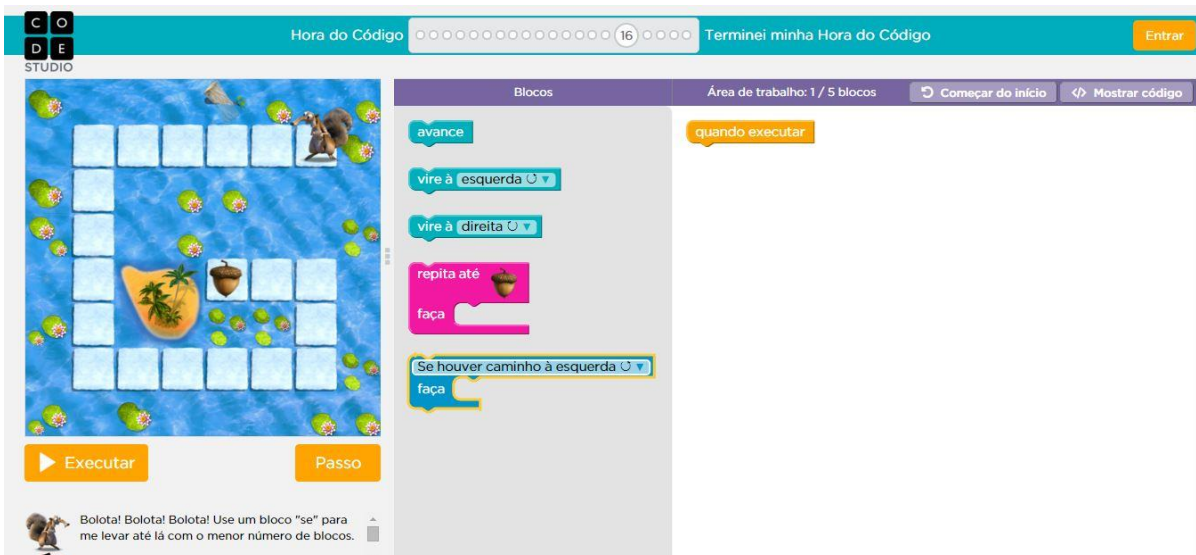
faça

Se houver caminho à direita

faça

▶ Executar

Ok, esse desafio é como o último, mas você precisa se lembrar de como você usou o bloco "se" e o bloco "repita" juntos.

Tela 16 – <https://studio.code.org/hoc/16>

C O
D E

Hora do Código 16 Terminei minha Hora do Código Entrar

STUDIO

Blocos Área de trabalho: 1 / 5 blocos Começar do início Mostrar código

quando executar

avance

vire à esquerda

vire à direita

repita até

faça

Se houver caminho à esquerda

faça

▶ Executar Passo

Bolota! Bolota! Bolota! Use um bloco "se" para me levar até lá com o menor número de blocos.

Tela 17 – <https://studio.code.org/hoc/17>

C O
 D E

Hora do Código 17 Terminei minha Hora do Código Entrar

STUDIO

Blocos Área de trabalho: 1 / 5 blocos ↺ Começar do início ↻ Mostrar código

avance
 vire à esquerda
 vire à direita
 repita até
 faça
 se houver caminho à esquerda
 faça

quando executar

▶ Executar Passo

Ok, vamos fazer novamente para pegar a prática - este não é muito diferente, mas cuidado com a parede!

Tela 18 – <https://studio.code.org/hoc/18>

C O
 D E

Hora do Código 18 Terminei minha Hora do Código Entrar

STUDIO

Blocos Área de trabalho: 1 / 5 blocos ↺ Começar do início ↻ Mostrar código

avance
 vire à esquerda
 vire à direita
 repita até
 faça
 se houver caminho à frente
 faça
 se não

quando executar

▶ Executar Passo

Os blocos "se-se-não" verificam uma condição e, em seguida, fazem uma coisa OU outra. Para que eu chegue até a bolota, tente usar este novo

Tela 19 – <https://studio.code.org/hoc/19>

STUDIO

Hora do Código 19 Terminei minha Hora do Código Entrar

Blocs

Área de trabalho: 1 / 5 blocos

Começar do início Mostrar código

quando executar

avance

vire à esquerda

vire à direita

repita até

faça

se houver caminho à frente

faça

se não

Executar Passo

Vamos praticar o uso do bloco "se-senão" mais uma vez. Você consegue acertar na primeira tentativa?

Tela 20 – <https://studio.code.org/hoc/20>

STUDIO

Hora do Código 20 Terminei minha Hora do Código Entrar

Blocs

Área de trabalho: 4 / 7 blocos

Começar do início Mostrar código

quando executar

repita até

faça

se houver caminho à frente

faça

se não

se houver caminho à direita

faça

se não

avance

vire à esquerda

vire à direita

repita até

faça

se houver caminho à frente

faça

se não

Executar Passo

Você consegue adicionar somente 3 blocos para me ajudar a resolver um labirinto mais complexo? Se você fizer isso corretamente, eu...