UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

NATHALIA DA CRUZ ALVES

DESENVOLVIMENTO DE UMA UNIDADE INSTRUCIONAL INTERDISCIPLINAR PARA ENSINAR COMPUTAÇÃO NO ENSINO FUNDAMENTAL

FLORIANÓPOLIS
2016

NATHALIA DA CRUZ ALVES

DESENVOLVIMENTO DE UMA UNIDADE INSTRUCIONAL INTERDISCIPLINAR PARA ENSINAR COMPUTAÇÃO NO ENSINO FUNDAMENTAL

Trabalho de Conclusão do Curso de Graduação em Ciências da Computação, do Departamento de Informática e Estatística, do Centro Tecnológico da Universidade Federal de Santa Catarina, requisito parcial à obtenção do título de Bacharel em Ciências da Computação.

Orientadora: Prof.ª, Dr.ª, rer., nat., Christiane Gresse von Wangenheim.

FLORIANÓPOLIS
2016

NATHALIA DA CRUZ ALVES

DESENVOLVIMENTO DE UMA UNIDADE INSTRUCIONAL INTERDISCIPLINAR PARA ENSINAR COMPUTAÇÃO NO ENSINO FUNDAMENTAL

Trabalho de conclusão de curso submetido ao Departamento de Informática e Estatística da Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharelado em Ciências da Computação.

Orientadora:

_____

Prof.ª, Dr.ª, rer., nat., Christiane Gresse von Wangenheim
Universidade Federal de Santa Catarina

Banca Examinadora:

_____

Prof., Dr., Jean Carlo Rossa Hauck
Universidade Federal de Santa Catarina

_____

Prof., Msc., Pedro Eurico Rodrigues
Universidade de São Paulo

FLORIANÓPOLIS
2016

RESUMO

A computação está se tornando cada vez mais onipresente na vida do ser humano. A cada dia que passa, novos dispositivos digitais passam a fazer parte do dia-a-dia das pessoas. É indiscutível a importância da computação atualmente. Apesar disso, no Brasil, seus fundamentos só são ensinados em cursos específicos do ensino superior. Os conceitos da computação são valiosos em várias áreas, de forma que o ensino da computação auxilia não só na resolução de problemas em geral, mas também na estruturação do pensamento. Nesse contexto, este trabalho visa o desenvolvimento sistemático de uma unidade instrucional interdisciplinar alinhada às diretrizes de currículo para computação e para o Ensino de História no Ensino Fundamental 2. A unidade instrucional tem como foco o ensino de conceitos básicos de computação, tais como a concepção do pensamento lógico e computacional e programação. Seguindo o método de *design* instrucional conhecido como ADDIE são projetadas atividades relativas ao ensino da unidade e desenvolvidos recursos didáticos que são utilizados durante a aplicação da unidade. A aplicação da unidade é realizada em duas etapas com diferentes turmas de escolas de Ensino Fundamental. Os resultados apresentam que a unidade é adequada pois os alunos atingem a maioria dos objetivos de aprendizagem por meio da unidade instrucional, e motivadora pois desperta interesse nos alunos acerca da computação.

Palavras-chave: Computação Interdisciplinar, Unidade Instrucional, Ensino Fundamental, Ensino de História.

ABSTRACT

Computing is becoming increasingly ubiquitous in human life. Every day, new digital devices become part of day-to-day lives. There is no doubt of the importance of computing today. Nevertheless, in Brazil, computing concepts are only taught in specific courses of higher education. The computer science concepts are valuable in many areas; therefore, teaching computing not only assists in solving problems in general, but also in the structure of thought. In this context, this project consists of the systematic development of an interdisciplinary instructional unit aligned to the curriculum guidelines of computer science and History in Elementary/Middle School. The instructional unit focuses on teaching the basics of computing, such as the design of logical and computational thinking and programming. Following the instructional design method known as ADDIE activities are designed for the instructional unit and learning resources are developed to be used during application of the unit. The application of the unit is performed in two stages with different classes of primary schools. The results show that the unit is suitable, as the students achieve most of the learning objectives through the instructional unit, and motivational, as it raises interest in students about computing.

Keywords: Computer Interdisciplinary, Instructional Unit, Elementary/Middle School, History Teaching.

LISTA DE FIGURAS

Figura 1 — Áreas de conhecimento para o ensino de computação no Ensino Fundamental (CSTA, 2011). 12
Figura 2 — Método de pesquisa. 16
Figura 3 — Diagrama com as cinco fases do modelo ADDIE (BRANCH, 2009). 21
Figura 4 — Níveis de ensino de computação (CSTA, 2011) 26
Figura 5 — Ambiente de programação Scratch off-line em português (SCRATCH, 2016). 34
Figura 6 — Percentual de brasileiros de 10 a 14 anos que utilizaram a Internet, no período de referência dos últimos três meses, por Grandes Regiões - 2014 (IBGE, 2014). 51
Figura 7 — Sistema operacional dos computadores da escola pública (PROINFO; MEC, 2016). 51
Figura 8 — Comparação entre o número de concluintes de cursos de matemática e português com informática (INEP, 2010 a 2014). 52
Figura 9 — Alunos dos quintos e sétimos anos da escola Autonomia durante a aplicação da UNIfICA v1.0 piloto. 56
Figura 10 — Extrato de jogos desenvolvidos pelos alunos durante a unidade instrucional UNIfICA v1.0 piloto. 57
Figura 11 — Dados de antes/depois da aplicação da UNIfICA v1.0 piloto sobre habilidade de fazer programas de computador. 59
Figura 12 — Dados depois da aplicação da UNIfICA v1.0 piloto sobre dificuldade das aulas e de fazer programas de computador. 59
Figura 13 — Dados após a aplicação da UNIfICA v1.0 piloto. 60
Figura 14 — Alunos do sexto ano do colégio Reinaldo Weingartner durante a aplicação da UNIfICA v2.0. 72
Figura 15 — Extrato de jogos desenvolvidos pelos alunos durante a UNIfICA v2.0. 73
Figura 16 — Gênero dos jogos criados pelos alunos durante a aplicação da UNIfICA v2.0. 74
Figura 17 — Capacidade de ensinar alguém a criar um programa de computador antes e depois da UNIfICA v2.0. 75
Figura 18 — Capacidade de criar um programa de computador antes e depois da UNIfICA v2.0. 75
Figura 19 — Análise dos jogos via Dr.Scratch. 77
Figura 20 — Atendimento dos objetivos de história nos jogos desenvolvidos durante a UNIfICA v2.0. 78
Figura 21 — Percepção da facilidade das aulas pelos alunos durante a UNIfICA v2.0. 78
Figura 22 — Percepção dos alunos em relação à facilidade de fazer programas de computador antes e depois da UNIfICA v2.0. 79
Figura 23 — Experiência das aulas durante a UNIfICA v2.0. 80
Figura 24 — Satisfação em mostrar o jogo a outras pessoas durante a UNIfICA v2.0. 80
Figura 25 — Percepção dos alunos sobre diversão em criar um programa de computador antes e depois da UNIfICA v2.0. 81
Figura 26 — Motivação em relação à computação depois da UNIfICA v2.0. 82

LISTA DE TABELAS

Tabela 1 — Categorias do domínio cognitivo da Taxonomia de Bloom (BLOOM, 1956). 23
Tabela 2 — Categorias do domínio afetivo da Taxonomia de Bloom (BLOOM, 1956).....23
Tabela 3 — Categorias do domínio psicomotor por Simpson (SIMPSON, 1972). 24
Tabela 4 — Objetivos de aprendizagem para os níveis 1B e 2 (CSTA, 2011). 28
Tabela 5 — Elementos comuns de um jogo (WANGENHEIM e WANGENHEIM, 2012).....32
Tabela 6 — Gênero de jogos (adaptado de HERZ, 1997). 33
Tabela 7 — Categorias de blocos do Scratch. 35
Tabela 8 — Terminologia para o ambiente Scratch.....36
Tabela 9 — Pontuação das áreas de análise pelo Dr.Scratch (PROGRAMAMOS; DR.SCRATCH). 37
Tabela 10 — Objetivos gerais de História (MEC; PCN 5.2, 1998). 38
Tabela 11 — Ciclos e eixos temáticos de História no Ensino Fundamental (MEC; PCN Volume 1, 1998).....39
Tabela 12 — Objetivos de História no ciclo 2 (MEC; PCN Volume 5.1, p. 45, 1998). 39
Tabela 13 — Objetivos de História no ciclo 3 (MEC; PCN Volume 5.2, 1998, p. 54). 40
Tabela 14 — Critérios de exclusão e de inclusão da RSL. 42
Tabela 15 — Termos de busca. 43
Tabela 16 — Trabalhos encontrados na primeira iteração.....44
Tabela 17 — Trabalhos encontrados na segunda iteração.....44
Tabela 18 — Dados extraídos de trabalhos no estado da arte relacionados ao contexto. 45
Tabela 19 — Dados extraídos de trabalhos no estado da arte sobre unidades instrucionais. 46
Tabela 20 — Dados extraídos de trabalhos no estado da arte sobre avaliação de UI.....47
Tabela 21 — Média de alunos por turma no Ensino Fundamental (INEP, 2015). 52
Tabela 22 — Sequenciamento de conteúdo da UNIfICA v1.0 piloto. 54
Tabela 23 — Materiais desenvolvidos para a UNIfICA v1.0 piloto. 54
Tabela 24 — Quantidade de alunos.....56
Tabela 25 — Atendimento dos objetivos da UNIfICA v1.0 piloto.....58
Tabela 26 — Contabilização da utilização de comandos/conceitos.....58
Tabela 27 — Objetivos de aprendizagem específicos da unidade instrucional UNIfICA v2.0. 61
Tabela 28 — Sequenciamento da unidade instrucional UNIfICA v2.0. 63
Tabela 29 — Materiais instrucionais da unidade UNIfICA v2.0. 67
Tabela 30 — Detalhamento do planejamento da avaliação da UNIfICA v2.0. 70
Tabela 31 — Pontos levantados pelos alunos durante a UNIfICA v2.0. 81
Tabela 32 — Recomendações para a aplicação da UNIfICA v2.0. 83

LISTA DE ABREVIATURAS

ACM – *Association for Computing Machinery*

ADDIE – *Analyze, Design, Develop, Implement, Evaluate*

BNCC – Base Nacional Comum Curricular

CEPSH – Comitê de Ética em Pesquisa com Seres Humanos

CSTA – *Computer Science Teachers Association*

IEEE – Instituto de Engenheiros Eletricistas e Eletrônicos

INEP – Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

ISD – *Instructional System Development*

LDB – Lei de Diretrizes e Bases da Educação Nacional

MEC – Ministério da Educação

MIT – *Massachusetts Institute of Technology*

PCN – Parâmetros Curriculares Nacionais

RSL – Revisão Sistemática da Literatura

SBC – Sociedade Brasileira de Computação

TI – Tecnologia da Informação

UI – Unidade Instrucional

UNIfICA – UNidade Instrucional Interdisciplinar de Computação e HistóriA

SUMÁRIO

LISTA DE FIGURAS 6
LISTA DE TABELAS 7
LISTA DE ABREVIATURAS 8
1. INTRODUÇÃO 11
1.1 CONTEXTUALIZAÇÃO 11
1.2 OBJETIVOS 14
1.3 MÉTODO DE PESQUISA 15
1.4 ESTRUTURA DO TRABALHO 18
2. FUNDAMENTAÇÃO TEÓRICA 19
2.1 O PROCESSO DE ENSINO E APRENDIZAGEM 19
2.2 ENSINO DE COMPUTAÇÃO NO ENSINO FUNDAMENTAL 24
2.3 APRENDER COM O DESENVOLVIMENTO DE JOGOS 31
2.4 AMBIENTE E LINGUAGEM DE PROGRAMAÇÃO SCRATCH 34
2.4.1 Blocos de programação do ambiente Scratch 35
2.4.2 Terminologia do ambiente Scratch 36
2.5 ENSINO DE HISTÓRIA NO ENSINO FUNDAMENTAL 38
3. ESTADO DA ARTE 42
3.1 DEFINIÇÃO DA REVISÃO 42
3.2 EXECUÇÃO DA BUSCA 43
3.3 EXTRAÇÃO DE INFORMAÇÃO 45
3.4 DISCUSSÃO 47
3.4.1 Ameaças à validade 49
4. PESQUISA EXPLORATÓRIA 50
4.1 ANÁLISE DE CONTEXTO 50
4.2 DESIGN INSTRUCIONAL UNIfICA v1.0 PILOTO 53
4.3 DESENVOLVIMENTO DE MATERIAIS PILOTO 54
4.4 APLICAÇÃO PILOTO 55
4.5 ANÁLISE DOS DADOS COLETADOS 58
5. UNIDADE INSTRUCIONAL UNIfICA v2.0 61
5.1 DESIGN INSTRUCIONAL 61
5.2 DESENVOLVIMENTO DE MATERIAIS 67
6. AVALIAÇÃO 70
6.1 PLANEJAMENTO DA AVALIAÇÃO 70

6.2 EXECUÇÃO DA AVALIAÇÃO 71
6.3 ANÁLISE DOS DADOS COLETADOS 74
6.3.1 Análise do atingimento dos objetivos de aprendizagem da unidade 74
6.3.2 Análise do grau de facilidade de aprendizagem da unidade 78
6.3.3 Análise da experiência de aprendizagem da unidade 79
6.3.4 Análise da percepção do aluno em relação à computação (antes e depois) 81
6.4 DISCUSSÃO 83
7. CONCLUSÃO 84
REFERÊNCIAS 85
ANEXO A – APROVAÇÃO PREFEITURA DE PALHOÇA 89
ANEXO B – ARTIGO 90

1. INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A computação tem revolucionado o mundo moderno. O advento dos computadores, e a aquisição dos mesmos posteriormente pela população em geral, foi imprescindível para o seu avanço. A computação ganhou status de ciência no final da década de 1950, onde o termo *computer science* (pt: ciência da computação) apareceu em um artigo da revista *Communications of the ACM* (FINE, 1959). Nesse artigo, argumenta-se que a ciência da computação é aplicada e interdisciplinar por natureza, apesar de ter as características típicas de uma disciplina acadêmica (FINE, 1959).

Atualmente, é fundamental conhecer os conceitos básicos da ciência da computação, os quais podem ser aplicados na resolução de problemas de diferentes áreas do conhecimento, ainda que elas não estejam diretamente relacionadas à computação (CSTA, 2011). Entre seus conceitos básicos estão: o **pensamento computacional** (pensar de forma algorítmica, isto é, passo a passo, de forma sistemática) e a **programação de computadores** (uso de uma linguagem de programação para dar comandos ao computador). Ambos fazem uso, principalmente, do raciocínio lógico e algorítmico, os quais auxiliam na estruturação do pensamento (CSTA, 2011).

É evidente que os conceitos da computação são valiosos em todos os contextos, ou seja, aprendê-los, prepara melhor os cidadãos para o cenário atual da tecnologia, tornando-os fluente digitalmente (BLIKSTEIN, 2008). Contudo, pode-se dizer que de todas as habilidades mais requeridas do indivíduo no século XXI, aquela que talvez seja a mais importante e a menos compreendida é o pensamento computacional (BLIKSTEIN, 2008). A fluência digital não envolve apenas saber como se usam novas ferramentas/aplicações digitais. Ela também inclui saber como realizar coisas significativas e desenvolver novas formas de pensar baseando-se no uso destas ferramentas (LIN, 2002). Por exemplo, uma pessoa pode saber utilizar um buscador *on-line*, mas na fluência digital isto não basta. Esta pessoa deve ser capaz de realizar pesquisas e encontrar resultados significativos que contém novas informações e assim, adquirir novos conhecimentos por meio do raciocínio, inferência e reflexão sobre estas informações.

Um dos problemas relacionados à fluência digital trata-se da escassez de profissionais

de TI (Tecnologia da Informação). Se nada for feito a respeito disso, estima-se que em 2020 o déficit de profissionais de TI, no Brasil, pode chegar a 408 mil (SOFTEX, 2013). Além disso, as taxas de evasão de cursos superiores relacionados à computação estão entre as maiores do país segundo Filho:

> [...] a área de Ciências, Matemática e Computação tem uma taxa em torno dos 28%, consideravelmente acima, portanto, da média nacional, abaixo apenas da evasão anual média registrada nos cursos de Serviços. (FILHO *et al.*, 2007, p. 653)

Isso evidencia que poucas pessoas possuem proficiência e conhecimento em computação. Uma das formas de contornar esse problema, seria **inserir o ensino da computação no Ensino Fundamental**, popularizando-a e motivando os alunos a seguirem carreira na área.

Contudo, muitos fatores influenciam o fato da computação não ser ensinada no Ensino Fundamental, entre eles, a ausência de meios, tanto na parte de materiais quanto de recursos humanos qualificados para tal (INEP, 2010 a 2014). Também, a própria cultura de ensino do Brasil não se encontra totalmente envolvida nesse contexto. Atualmente, encontra-se em desenvolvimento no Brasil a **BNCC** (Base Nacional Comum Curricular, 2016), a qual ainda foca no uso da tecnologia digital como um tema integrador, ao invés da produção de tecnologia em si (BNCC, 2016).

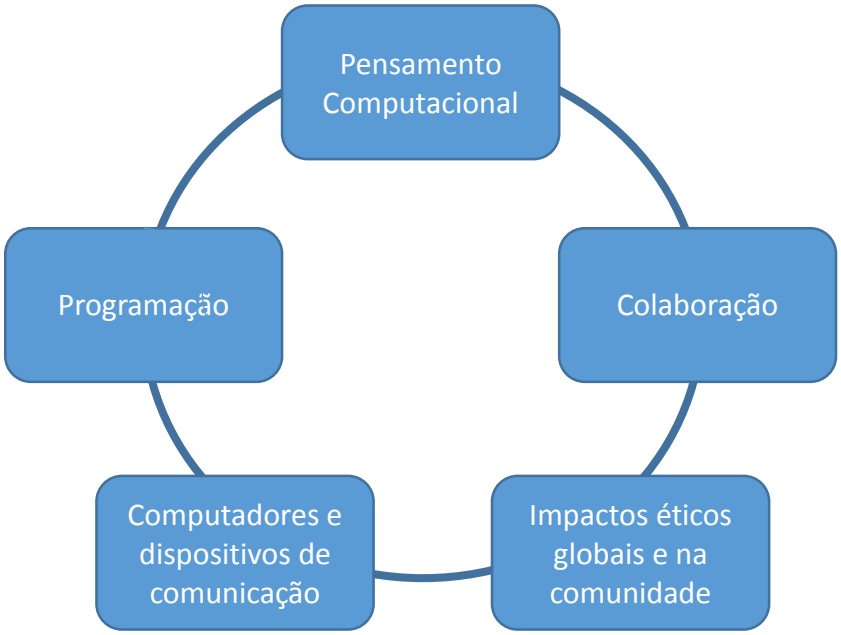


Figura 1 — Áreas de conhecimento para o ensino de computação no Ensino Fundamental (CSTA, 2011).

Idealmente, o ensino de computação no Ensino Fundamental deve abordar várias áreas de conhecimento conforme apresentado na Figura 1. O pensamento computacional, a programação, a colaboração, os computadores e dispositivos de comunicação e os impactos éticos, globais e na comunidade fazem parte de um conjunto de diretrizes para o ensino de computação (CSTA, 2011).

O pensamento computacional é uma forma de raciocínio utilizada para resolver problemas. Trata-se de um conjunto de conceitos tais como abstração, recursão, análise de dados, algoritmos, entre outros (CSTA, 2011). A programação é essencial na computação, pois garante a competência de criar programas de softwares e dá ao indivíduo a capacidade de aprender a projetar, desenvolver e publicar produtos (CSTA, 2011). Computadores e dispositivos de comunicação são de grande importância atualmente, sendo ferramentas essenciais para realizar de forma eficiente várias atividades em campos variados do conhecimento (CSTA, 2011). Impactos éticos, globais e na comunidade, tratam-se de questões de privacidade, segurança de rede, licenças de direito autoral, entre outros fatores, positivos ou negativos que possam vir a afetar a sociedade, é importante ensinar os alunos a serem cidadãos responsáveis (CSTA, 2011). A colaboração diz respeito ao trabalho em equipe, haja vista raramente o progresso significativo em computação é feito por uma única pessoa, geralmente são grandes equipes trabalhando em conjunto (CSTA, 2011).

Atualmente, já existem iniciativas ao redor do mundo para ajudar na popularização da computação entre crianças, adolescentes e leigos, como, por exemplo, Code.org (CODE, 2013) e Scratch (SCRATCH; MIT, 2013). Estas são ferramentas para ensinar conceitos básicos de programação. No Brasil, também existem iniciativas deste caráter, como, por exemplo, Computação na Escola (CNE, 2013) e Scratch Brasil (SCRATCH BRASIL, 2014). Também já foram aplicadas oficinas para o público em geral (ANGELO, HENNO e CAMPOS, 2013) e já foram realizados pilotos em escolas (WANGENHEIM, NUNES e DOS SANTOS, 2014). No entanto, foram encontradas poucas aplicações desse tipo. É necessário que estas iniciativas atinjam um público maior.

O ideal seria que os alunos aprendessem computação já no Ensino Fundamental, mas, atualmente, a grade curricular encontra-se lotada, isto é, com base nas horas-aulas recomendadas para cada disciplina, não há tempo disponível para a inserção da computação como uma disciplina (LDB, 1996). Também não foram encontradas **UI** (Unidades Instrucionais) interdisciplinares prontas, já com um guia de professor para aqueles que não são formados na

área, para ensinar computação no Ensino Fundamental.

Desta maneira, este trabalho pretende vincular o ensino da computação ao conteúdo de uma disciplina do Ensino Fundamental. Isto é feito por meio de uma Unidade Instrucional a qual abre a possibilidade do ensino da computação para todos os alunos do Ensino Fundamental. Quanto se estuda História, depara-se com o que os homens foram e fizeram, e isto auxilia na compreensão do que se pode ser e fazer. A História é a ciência do passado e do presente, no entanto, o estudo do passado e a compreensão do presente não acontecem de forma perfeita, já que não há como voltar ao passado e o mesmo não se repete. Já a computação nos permite criar mundos (referindo-se a mundos digitais, factíveis por meio da programação de computadores e afins), os quais, de certa forma, podem ser uma recriação daquilo que um dia já existiu. Assim, a UI vincula os conteúdos de computação e História permitindo aos alunos o aprendizado de conceitos de computação, e ao mesmo tempo, construção de conhecimento histórico.

Este trabalho visa o *design* das atividades e materiais didáticos relativos ao ensino da computação no Ensino Fundamental por meio de uma unidade instrucional, incluindo a definição de atividades e desenvolvimento de todos os materiais necessários. Esta unidade, desenvolvida durante o trabalho, estará alinhada com as diretrizes do currículo de referência da ACM/IEEE/CSTA para Ensino Fundamental e seguirá o método de Design Instrucional conhecido como ADDIE (BRANCH, 2009).

Uma das principais dificuldades é a falta de professores formados nesta área, assim, para minimizar este problema e possibilitar a aplicação da Unidade Instrucional, esta unidade tem um guia para o professor, viabilizando a implantação da computação na prática de modo independente, isto é, pode ser aplicado por qualquer professor na disciplina de História que tenha interesse. Espera-se assim motivar tanto os professores na parte do ensino da computação, quanto os alunos a seguirem esta área no futuro.

1.2 OBJETIVOS

Objetivo geral

Desenvolver sistematicamente uma unidade instrucional de 14 horas/aula para ensinar computação, por meio da programação de um jogo com Scratch, de forma interdisciplinar vinculada a disciplinas de História em escolas brasileiras em nível de Ensino Fundamental.

Objetivos específicos

O1. Analisar a fundamentação teórica sobre ensino, aprendizagem, computação e a linguagem Scratch.

O2. Analisar o estado da arte sobre unidades instrucionais de ensino de computação integrada em disciplinas de História no Ensino Fundamental.

O3. Projetar uma unidade instrucional para ensinar computação integrada em disciplinas de História no Ensino Fundamental.

O4. Aplicar a unidade instrucional criada na prática.

O5. Avaliar a unidade instrucional.

Premissas e restrições

A unidade tem duração de 14 horas/aulas, de forma que permite uma familiarização inicial com conceitos de computação e programação pelos alunos, os quais posteriormente criam um jogo digital com tema de História. Esta restrição de duração também atenta-se ao fato de não se prolongar muito, de maneira que seja facilmente aplicada dentro de um bimestre letivo (dois meses de aula).

O trabalho é realizado de acordo com o regulamento vigente do Departamento de Informática e Estatística (INE - UFSC) em relação aos Trabalhos de Conclusão de Curso. A unidade instrucional tem como foco o *design* instrucional de uma unidade para o ensino integrado às disciplinas de História somente para o Ensino Fundamental.

1.3 MÉTODO DE PESQUISA

O método de pesquisa utilizado neste trabalho é dividido em seis etapas (veja Figura 2).

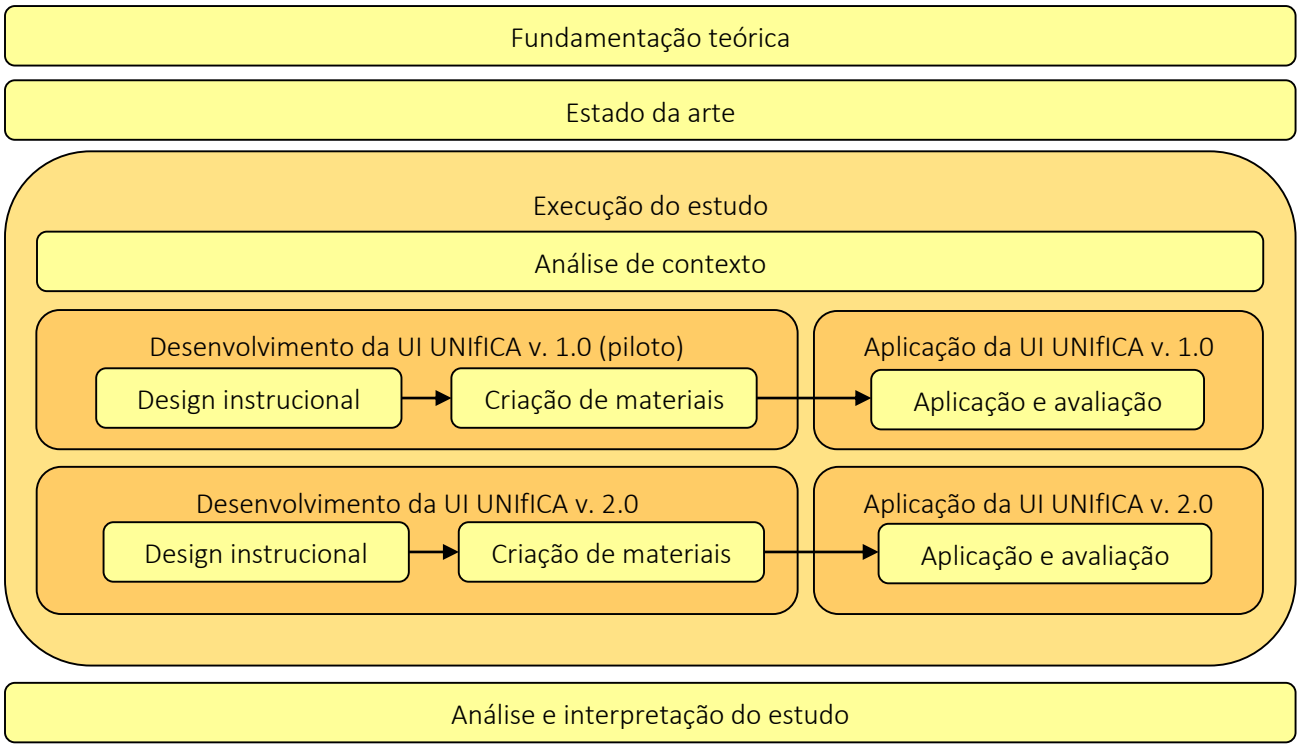


Figura 2 — Método de pesquisa.

A seguir é apresentada cada etapa de forma detalhada:

Etapa 1 – Fundamentação teórica

Nesta etapa é realizada uma análise teórica sobre conceitos relacionados ao escopo deste trabalho. Esta primeira grande etapa é dividida em cinco atividades:

A1.1 – Análise teórica sobre ensino e aprendizagem.

A1.2 – Análise teórica sobre *design* instrucional.

A1.3 – Análise teórica sobre computação no Ensino Fundamental.

A1.4 – Análise teórica sobre a ferramenta Scratch.

A1.5 – Análise teórica sobre a História no Ensino Fundamental.

Etapa 2 – Estado da arte

Nesta etapa é realizada uma revisão sistemática da literatura segundo o processo proposto por Kitchenham (KITCHENHAM, 2004) para identificar e analisar unidades instrucionais/estratégias de ensino atualmente sendo utilizadas e voltadas ao ensino de computação interdisciplinar. Esta segunda grande etapa é dividida em quatro atividades:

A2.1 – Definição da revisão sistemática.

A2.2 – Execução da busca.

A2.3 – Extração e análise da informação.

A2.4 – Discussão.

Etapa 3 – Execução do estudo: Análise de contexto

Nesta etapa é realizada uma análise de contexto referente ao público alvo e ambiente de aplicação de uma unidade instrucional para se ensinar computação de forma interdisciplinar com História. Esta terceira etapa é dividida em duas partes:

A3.1 – Análise do público alvo.

A3.2 – Análise do ambiente.

Etapa 4 – Execução do estudo: Desenvolvimento e aplicação da UNIfICA v1.0 piloto

Nesta etapa é feito o *design* instrucional, seguindo o método de *design* instrucional conhecido como ADDIE (BRANCH, 2009), da UNIfICA – **UN**idade **I**nstrucional **I**nterdisciplinar de **C**omputação e Histórí**A**. Após isto, são criados materiais instrucionais e então realizada a aplicação da unidade por meio de um estudo de caso, conforme os procedimentos propostos por Yin (YIN, 2001), em uma escola privada que oferece Ensino Fundamental. Esta pesquisa permite a familiarização na prática com o perfil do aluno, com a ementa da disciplina de História e com o tipo de infraestrutura que a escola tipicamente provê. Também fornece *feedback* acerca da aprendizagem de conceitos de computação. Esta quarta etapa tem sete partes:

A4.1 – *Design* da UNIfICA v1.0 piloto (plano de ensino).

A4.2 – Definição da avaliação (rubrica).

A4.3 – Desenvolvimento do guia do professor (roteiro).

A4.4 – Desenvolvimento de jogos com tema de história.

A4.5 – Desenvolvimento de questionários.

A4.6 – Planejamento do estudo.

A4.7 – Execução do estudo.

Etapa 5 – Execução do estudo: Desenvolvimento e aplicação da UNIfICA v2.0

Nesta etapa é definido o *design* da unidade instrucional (BRANCH, 2009) a partir de melhoramentos identificados pelo *feedback* da aplicação da UNIfICA v1.0 piloto. Também são criados novos materiais e rubricas para se avaliar a unidade UNIfICA v2.0. A unidade instrucional UNIfICA v2.0 é colocada em prática por meio de um estudo de caso (YIN, 2001)

realizado numa escola pública de Ensino Fundamental para ensinar computação com História. Esta etapa é dividida em onze atividades:

A5.1 – *Design* da UNIfICA v2.0 (plano de ensino).

A5.2 – Definição da avaliação (rubrica).

A5.3 – Desenvolvimento do material de apresentação (*slides*).

A5.4 – Desenvolvimento de atividades, lista de grupos e *worksheet*.

A5.5 – Desenvolvimento de um novo guia do professor (roteiro).

A5.6 – Desenvolvimento da prova final e gabarito.

A5.7 – Desenvolvimento de informativo para pais.

A5.8 – Desenvolvimento de novos jogos com tema de história.

A5.9 – Melhoramento dos questionários.

A5.10 – Planejamento do estudo.

A5.11 – Execução do estudo.

Etapa 6 – Análise e interpretação do estudo

Nesta etapa são analisados os dados das aplicações da UNIfICA versão piloto (v1.0) e versão final (v2.0). Os dados são analisados por meio de análises qualitativas e quantitativas utilizando estatística descritiva. Esta etapa final é dividida em duas atividades:

A6.1 – Análise dos dados da aplicação da UNIfICA v1.0.

A6.2 – Análise dos dados da aplicação da UNIfICA v2.0.

1.4 ESTRUTURA DO TRABALHO

No capítulo 2 é realizada uma fundamentação teórica sobre os conceitos necessários para o desenvolvimento da unidade instrucional. No capítulo 3 é realizada uma revisão sistemática da literatura sobre o ensino de computação de forma interdisciplinar, no qual os alunos desenvolvem jogo como objeto de aprendizagem. No capítulo 4 é apresentada uma análise de contexto, o *design* e materiais da UNIfICA v1.0 piloto e a análise dos dados obtidos por meio uma pesquisa exploratória. No capítulo 5 é apresentada a unidade instrucional UNIfICA v2.0 para ensinar computação de forma interdisciplinar com História, incluindo *design* e um extrato da versão final dos materiais. No capítulo 6 são apresentados e analisados os dados da aplicação da unidade UNIfICA v2.0. No capítulo 7 é apresentada uma discussão sobre o resultado do presente trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

São apresentados neste capítulo conceitos referentes à teoria de ensino e aprendizagem e ao *design* instrucional. Também são abordados tópicos sobre aprender com o desenvolvimento de jogos, ensino de computação no Ensino Fundamental e uma visão geral sobre o ambiente e linguagem de programação Scratch. Por fim, questões sobre o Ensino de História no Ensino Fundamental.

2.1 O PROCESSO DE ENSINO E APRENDIZAGEM

A teoria de ensino e aprendizagem pode ser abordada sob diversas óticas do conhecimento. Na literatura são encontradas as mais variadas definições devido à complexidade envolta a este processo. Sendo assim, são abordados apenas conceitos relacionados ao contexto deste trabalho.

Apesar de dois temas distintos, o ensino e a aprendizagem estão fortemente conectados haja vista a complementação entre ambos. O ensino está relacionado à transmissão de conhecimentos e a aprendizagem à aquisição de conhecimentos (SPRINTHALL, 1993).

O processo em que há a justaposição do ensino e da aprendizagem é chamado de educação. A educação é um fenômeno intrínseco em toda e qualquer sociedade. É por meio dela que ocorre a transferência de conhecimento entre gerações. Formalmente, a educação é definida segundo o Dicionário Michaelis como:

> [...] formação consciente das novas gerações segundo os ideais de cultura de cada povo […]. Aperfeiçoamento das faculdades físicas intelectuais e morais do ser humano; disciplinamento, instrução, ensino. (MICHAELIS, 2000)

A educação trata-se de um fenômeno que sofre constantes mudanças de acordo com inúmeros aspectos, tais como, entre outros, fatores temporais e locais. Portanto a educação deve ser tratada como um processo, já que não pode ser vista como um pacote pronto e imutável (VENTURA, 2005).

Modelos de aprendizagem

Podem ser definidos como maneiras ou processos pelos quais as pessoas aprendem ou constroem conhecimentos. A seguir são apresentadas as definições para o modelo instrucional

e para o modelo construtivista.

- **Modelo instrucional**: é o modelo tradicional de aprendizagem. Consiste na absorção passiva do aluno sem levar em conta o seu perfil, o professor é a figura especialista no assunto e é aplicado normalmente em um ambiente que o acesso à mídias e informações é dificultado.
- **Modelo construtivista**: processo no qual o aluno deve construir o conhecimento. Um domínio dentro de uma área de conhecimento é especificada ao aluno, e este é encorajado a buscar novos domínios do conhecimento que sejam significantes dentro do contexto. Neste modelo não existem os objetivos de aprendizagem pré-definidos. Entretanto, alguns autores defendem que nem todo conhecimento pode ser construído pelo aluno. Deve haver um ponto de início para que a construção seja feita (WINN, 1991).

***Design* Instrucional**

O *design* instrucional ou engenharia pedagógica (BASQUE, 2010), trata-se de uma metodologia para fazer com que a aquisição de conhecimentos e habilidades seja eficiente, efetiva e motivadora (MERRILL *et al.*, 1996). O *design* instrucional faz uma adaptação à teoria construtivista. Consiste no processo do estudo do contexto sobre o perfil do aluno e do ambiente, e posteriormente, na definição objetivos finais e desenvolvimento de algum tipo de intervenção que auxilie nesta transição para o atingimento dos objetivos.

Constitui-se também, na realização de uma sequência de passos a qual serve para identificar necessidades de conhecimento e assim, encontrar meios para supri-las (QUINN, 2005). De acordo com Filatro, o *design* instrucional corresponde a:

> Ação intencional e sistemática de ensino, que envolve o planejamento, o desenvolvimento e a utilização de métodos, técnicas, atividades, materiais, eventos e produtos educacionais em situações didáticas específicas, a fim de facilitar a aprendizagem humana a partir dos princípios de aprendizagem e instrução conhecidos. (FILATRO, 2004, p. 65).

Desta forma, para se realizar esta ação, são utilizados modelos para o desenvolvimento de materiais instrucionais de forma sistemática. O termo **ISD** (*Instructional System Development*) é usado para fazer referência aos modelos que se baseiam no *design* instrucional (BRANCH, 2009). Um dos principais modelos ISD é o modelo **ADDIE** (BRANCH,

2009).

Modelo ADDIE

Trata-se de um processo iterativo que envolve múltiplos ciclos de *feedback* e várias atividades são realizadas simultaneamente (veja Figura 3). Possui uma abordagem de acordo com o ISD. O nome ADDIE é um acrônimo para as palavras: ***A**nalyze, **D**esign, **D**evelop, **I**mplement* e ***E**valuate* (pt: Análise, Projeto, Desenvolvimento, Implementação e Avaliação).

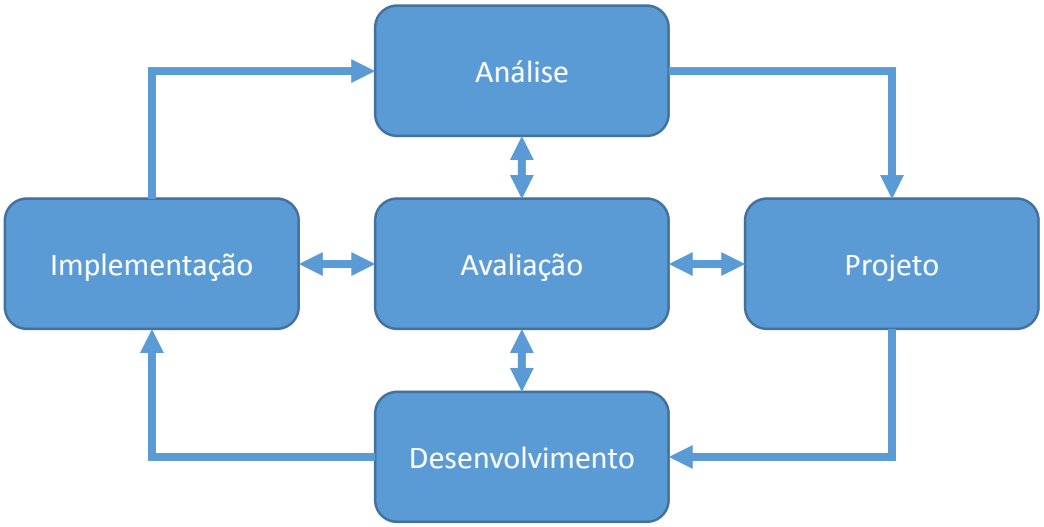


Figura 3 — Diagrama com as cinco fases do modelo ADDIE (BRANCH, 2009).

Dentro de cada fase, existem instruções claramente definidas. Estas instruções são explicadas mais detalhadamente a seguir (BRANCH, 2009):

- **Análise**: esta fase consiste na análise de componentes que serão utilizados para orientar o projeto de desenvolvimento da unidade instrucional. Devem ser analisadas as características do público-alvo, isto é, estudados o nível de habilidade que cada aprendiz apresenta ter, que conhecimentos ele possui e que conhecimentos ele deve ter após a aplicação da unidade instrucional. Também é analisado o contexto em que se insere a unidade instrucional e quais são as limitações de recursos humanos, técnicos, financeiro e de tempo.
- **Projeto**: nesta fase é feita a especificação detalhada do objetivo geral do projeto e de todos os objetivos de aprendizagem. Também são escolhidos quais recursos serão utilizados para o desenvolvimento da unidade instrucional, selecionados quais conteúdos deverão ser abordados na unidade instrucional e que métodos instrucionais serão utilizados. São

definidas quais os tipos de atividades que serão realizadas (colaborativas, interativas ou individuais) e quanto tempo será atribuído para cada uma delas. Também é feito o sequenciamento que definirá a ordem em que serão apresentados os conteúdos, a escolha de quais ferramentas serão utilizadas na medição de desempenho, os tipos de testes que serão feitos e a seleção da mídia de aprendizagem. Destina-se a explicar como a aprendizagem será adquirida.

- **Desenvolvimento**: esta fase destina-se à criação dos materiais a serem utilizados durante a unidade. Diferente das duas fases anteriores que exigiam planejamento e *brainstorming*, a fase de desenvolvimento tem a ver com colocar tudo isto em ação. São utilizadas várias ferramentas (papel, caneta, processadores de texto, editor de gráfico, software de programação, etc.), e envolve o desenvolvimento dos materiais projetados na fase anterior, podendo ser *slides*, folhas de exercícios, rubricas, guias, *worksheet*, entre outros.
- **Implementação**: é a fase em que a unidade instrucional desenvolvida é aplicada para direcionar os alunos à aprendizagem. Requer estabelecimento de uma infraestrutura organizacional e tecnológica. Permite testar todos os materiais para determinar se estes são funcionais e apropriados para a audiência analisada na primeira fase.
- **Avaliação**: esta fase serve para julgar as diferentes variáveis como qualidade, eficiência e eficácia da unidade instrucional com o intuito de melhorar ou fazer uma decisão sobre a aprovação ou reprovação para determinado ambiente. É avaliado se a unidade instrucional é efetiva, isto é, melhora a aprendizagem, motiva os aprendizes, etc. Determina de que modo será feito a coleta de dados, definindo as medições tanto para a avaliação da unidade instrucional como para a avaliação da aprendizagem do aluno, cujas observações incluem a confiabilidade e validade dos dados.

Dentro da fase de Projeto os objetivos de aprendizagem podem ser definidos utilizando-se a **Taxonomia de Bloom** (BLOOM, 1956). As especificações são feitas com base nas informações recolhidas junto a fase de Análise em conjunto com as teorias e modelos de *design* instrucional.

A Taxonomia de Bloom refere-se à **classificação** de diferentes objetivos de aprendizagem que foram definidos para alunos. É uma forma de se fazer distinção entre os aspectos fundamentais no contexto da educação. Dispõe do nome de Benjamin Bloom o qual presidiu a comissão de educadores que concebeu a taxonomia e editou o primeiro volume do

texto padrão (BLOOM, 1956).

É dividida em 3 domínios: **cognitivo**, **afetivo** e **psicomotor**. Dentro dos domínios existe uma hierarquia entre os níveis, de modo que a aprendizagem nos níveis mais altos é dependente de possuir conhecimento dos níveis mais baixos (ORLICH *et al.*, 2004).

Cognitivo: As competências deste domínio estão associadas ao conhecimento, compreensão e aplicação de um tema. É dividido em 6 níveis que se deslocam de ordem mais baixa para mais alta, são eles conhecimento, compreensão, aplicação, análise, síntese e avaliação (veja Tabela 1).

Tabela 1 — Categorias do domínio cognitivo da Taxonomia de Bloom (BLOOM, 1956).

1. Conhecimento	Memorização de materiais previamente aprendidos por fatos, termos, conceitos básicos e respostas.
2. Compreensão	Comparação demonstrativa, organizando, traduzindo e interpretando, fatos e ideias.
3. Aplicação	Aplicar o novo conhecimento. Conseguir resolver problemas de novas situações por meio da aplicação de um novo conhecimento, fatos e técnicas.
4. Análise	Examinar e dividir informações em partes, identificando causas. Fazer inferência e encontrar provas.
5. Síntese	Compilar informações juntas de uma maneira distinta.
6. Avaliação	Apresentar opinião e fazer julgamentos sobre informações e ideias.

Afetivo: neste domínio, há a descrição da reação emocional e capacidades de sentir alegria e fastio. É dividida em 5 níveis, são eles recepção, reação, valoração, organização e caracterização (veja Tabela 2).

Tabela 2 — Categorias do domínio afetivo da Taxonomia de Bloom (BLOOM, 1956).

1. Receber	O aluno participa com atenção passiva
2. Responder	O aluno participa ativamente do processo de aprendizado e reage de alguma forma expressando sua vontade de responder (motivação).
3. Valorizar	Quando o aluno atribui valor para objetivos, fenômenos ou informações.
4. Organizar	O aluno pode apresentar diferentes valores, informações e ideias organizando em seu próprio esquema, contrastando diferentes valores, resolvendo conflito entre eles, e criando um sistema único de valores pessoais.
5. Caracterizar	O aluno apresenta um determinado valor que exerce influência em seus comportamentos, até

	se tornar uma característica pessoal.

Psicomotor: aqui são descritas as habilidades de manipular fisicamente ferramentas ou instrumentos. Não foram criados níveis, mas alguns educadores criaram suas próprias taxonomias psicomotoras (CLARK, 2009). Os níveis propostos por Simpson são: percepção, prontidão, resposta guiada, mecanismo, resposta complexa, adaptação e originalidade (SIMPSON, 1972).

Tabela 3 — Categorias do domínio psicomotor por Simpson (SIMPSON, 1972).

1. Percepção	Habilidade de usar estímulos sensoriais para orientar as atividades, por exemplo, detectar os sinais de uma comunicação não-verbal.
2. Prontidão	Prontidão para agir, por exemplo, conhecer e agir em uma sequência de etapas em um processo de comunicação.
3. Resposta guiada	Estágios iniciais de uma aprendizagem de uma habilidade mais complexa, que inclui imitação, tentativa e erro. A adequação ao desempenho é alcançada por meio da prática, por exemplo, repetir um processo de comunicação demonstrando anteriormente.
4. Mecanismo	Estágio intermediário de uma habilidade complexa. Respostas aprendidas se tornam habituais, realizando movimentos com alguma confiança e proficiência.
5. Resposta complexa	Desempenho hábil, dos atos que envolvem padrões complexos e sem hesitação tendo desempenho automático.
6. Adaptação	As habilidades são bem desenvolvidas e o indivíduo pode modificar os padrões de movimento para atender as necessidades especiais, por exemplo, responder eficazmente às experiências inesperadas.
7. Originalidade	Criação de novos padrões de movimento para atender a uma determinada situação ou problema específico. Os resultados da aprendizagem enfatizam a criatividade com base em habilidades altamente desenvolvidas, como a construção de uma nova teoria de comunicação.

2.2 ENSINO DE COMPUTAÇÃO NO ENSINO FUNDAMENTAL

A Ciência da Computação estuda a fundamentação teórica das construções computacionais, bem como suas aplicações em dispositivos tecnológicos e sistemas de computação (MEC, 2003). Atualmente, o estudo da computação *per se* no Brasil é abordado somente em cursos do Ensino Superior.

Contudo, encontra-se em desenvolvimento a **BNCC** (BNCC, 2016) a qual apresenta

Direitos e Objetivos de Aprendizagem e Desenvolvimento que devem orientar a elaboração de currículos para as diferentes etapas de escolarização. A BNCC menciona o uso da tecnologia digital como um tema integrador dentro do **Tema Especial Culturas Digitais** (BNCC, 2016)**.** Mas, ainda na 2ª versão revista da BNCC a tecnologia é apresentada apenas como um **facilitador do processo** de aprendizagem e não como **seu fim**:

> [...] o Tema Especial culturas digitais e computação se relaciona à abordagem, nas diferentes etapas da Educação Básica e pelos diferentes componentes curriculares, do uso pedagógico das novas tecnologias da comunicação e da exploração dessas novas tecnologias para a compreensão do mundo e para a atuação nele.
>
> Numa perspectiva crítica, as tecnologias da informação e comunicação são instrumentos de mediação da aprendizagem e as escolas, especialmente os professores, devem contribuir para que o estudante aprenda a obter, transmitir, analisar e selecionar informações. (BNCC, 2016, p. 50).

Sendo assim, atualmente, no Brasil, a abordagem de conceitos da computação é restrita aos cursos de Ensino Superior. Para se promover o ensino de computação na educação básica foram feitos esforços, a partir de vários pesquisadores ao redor do mundo, para se criar recomendações de diretrizes de currículo (CSTA, 2011).

Computer Science Standards K-12 (CSTA, 2011)

As diretrizes de currículo CSTA K-12, estabelecem quais conhecimentos e competências acerca da ciência da computação os alunos devem ter para que possam ser capazes de desenvolver-se, adaptar-se e exercer a cidadania de forma efetiva no século XXI. Os principais tópicos são:

1. Introduzir os conceitos fundamentais da ciência da computação para todos os estudantes, a começar no Ensino Fundamental;
2. Apresentar a ciência da computação no nível do ensino secundário de uma maneira que possa ter referência à ciência da computação, matemática, ou qualquer outra graduação sobre uma área científica;
3. Incentivar as escolas a oferecerem cursos de ciência da computação adicionais de nível secundário, o que permitirá aos estudantes interessados a possibilidade de estudar aspectos da ciência da computação com mais profundidade e prepará-los para a entrada no mercado de trabalho ou faculdade;

4. Aumentar a disponibilidade de aprendizado da ciência da computação para todos os alunos, especialmente aqueles que pertencem à minorias.

Para o ensino da computação no Ensino Fundamental e Médio, as diretrizes CSTA K-12 são baseadas em um modelo com **3 níveis**. O nível 1 fornece os padrões de aprendizagem para os estudantes do ensino básico até o quinto ano. O nível 2 fornece os padrões de aprendizagem para estudantes do sexto ao nono ano. O nível 3 fornece os padrões de aprendizagem para os alunos do ensino médio. Este trabalho, é limitado apenas aos níveis 1 e 2 que são referentes ao Ensino Fundamental (veja Figura 4).

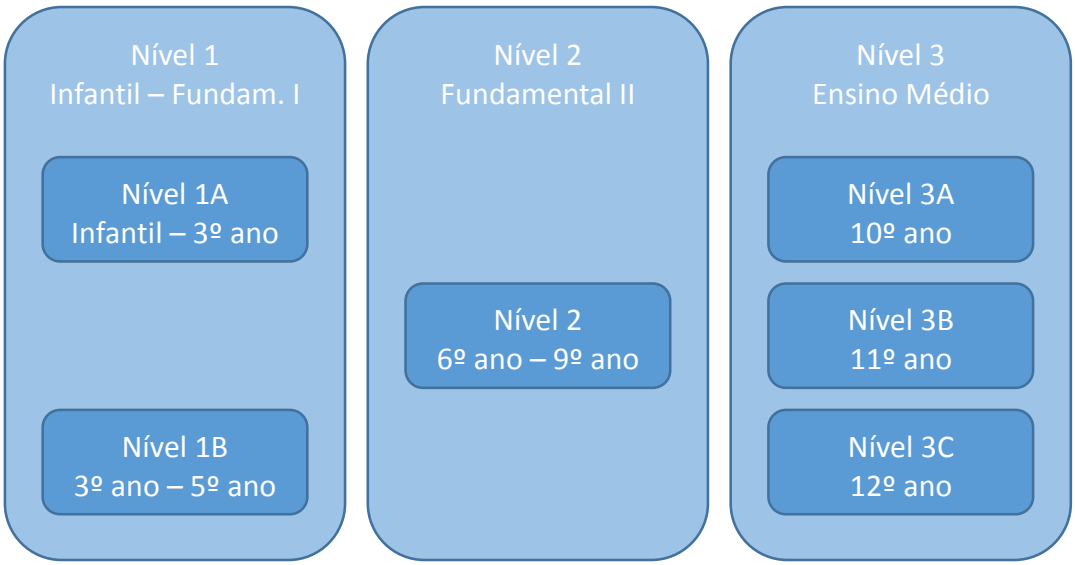


Figura 4 — Níveis de ensino de computação (CSTA, 2011)

Nível 1: os alunos do Ensino Fundamental são introduzidos aos conceitos fundamentais de ciência da computação pela integração de competências básicas em tecnologia com ideias simples sobre o pensamento computacional. As experiências de aprendizagem criadas devem ser inspiradoras e envolventes, ajudando os alunos a ver a computação como uma parte importante de seu mundo. Elas devem ser projetados com foco na aprendizagem ativa, criatividade e exploração e, muitas vezes, serem incorporadas dentro de outras áreas curriculares, tais como ciências sociais, língua, matemática e ciência (CSTA, 2011).

Nível 2: os alunos começam a usar o pensamento computacional como uma ferramenta para a resolução de problemas. Eles começam a apreciar a ubiquidade da computação e as formas na qual a computação facilita a comunicação e colaboração. Os alunos começam a

experimentar o pensamento computacional como um meio de abordar questões relevantes, não apenas para eles, mas para o mundo em torno deles. As experiências de aprendizagem criadas devem ser relevantes para os alunos e promover a sua percepção de si mesmos como solucionadores de problemas proativos e capacitados. Elas devem ser projetadas com foco na aprendizagem e exploração ativa. Elas podem ser ensinadas em disciplinas explícitas de ciência da computação ou incorporados em outras áreas curriculares, tais como ciências sociais, línguas, matemática e ciência (CSTA, 2011).

Dentro do ensino de computação para o Ensino Fundamental são abordadas 5 áreas que se complementam e são essenciais para todos os níveis de aprendizagem. Elas são: o pensamento computacional, a colaboração, a programação (prática da computação), os computadores e dispositivos de comunicação e os impactos éticos globais e na comunidade. A seguir é apresentado o detalhamento destas áreas (CSTA, 2011):

- **Pensamento computacional**: é uma abordagem para a resolução de problemas de forma que pode ser implementada com um computador. Os alunos tornam-se não somente utilizadores de ferramentas, mas construtores de ferramentas. Os alunos fazem uso de um conjunto de conceitos, tais como abstração, recursão e iteração, para processar e analisar dados e criar artefatos reais e virtuais. O pensamento computacional é uma metodologia de resolução de problemas que pode ser automatizada, transferida e aplicada para diferentes áreas. A relevância do pensamento computacional se dá pela facilidade de aplica-lo a qualquer outro tipo de raciocínio.
- **Colaboração**: a computação é uma disciplina intrinsecamente colaborativa. Progressos significativos raramente são feitos por uma pessoa trabalhando sozinha. Normalmente, os projetos de computação envolvem grandes equipes de profissionais de computação que trabalham em conjunto para projetar, codificar, testar, depurar, descrever e manter o software ao longo do tempo. Portanto, é importante que sejam desenvolvidas habilidades de colaboração, tais como trabalho em equipe, crítica construtiva e comunicação eficaz.
- **Programação (prática da computação)**: trata-se de uma parte essencial da computação, é a competência de criar programas de software. Alunos devem aprender a projetar, desenvolver e publicar produtos (*websites*, aplicações móveis, animações e jogos) utilizando recursos tecnológicos. Eles devem compreender o que são algoritmos e qual a sua aplicação prática. Como parte da prática, também devem implementar software utilizando uma linguagem de programação.

- **Computadores e dispositivos de comunicação**: os alunos devem compreender os elementos do computador moderno e de dispositivos e redes de comunicação. Os alunos devem usar a terminologia apropriada e precisa quando se comunicam acerca de tecnologia.
- **Impactos éticos globais e na comunidade**: princípios de privacidade, segurança de rede, licenças de software e direitos autorais devem ser ensinados a fim de preparar os alunos a se tornarem cidadãos responsáveis no mundo moderno. Os alunos também devem ser capazes de avaliar a confiabilidade e a precisão das informações na Internet. É essencial que os alunos compreendam o impacto dos computadores na comunicação internacional e devem aprender o comportamento apropriado em redes sociais. Os alunos também devem estar preparados para avaliar os diversos impactos positivos e negativos de computadores na sociedade e identificar até que ponto os problemas de acesso impactam nossas vidas.

Cada área possui uma gama de objetivos de aprendizagem específica para cada nível. A Tabela 4 apresenta os objetivos para os níveis 1B (3º ano – 5º ano) e 2 (6º ano – 9º ano).

Tabela 4 — Objetivos de aprendizagem para os níveis 1B e 2 (CSTA, 2011).

Nível 1B	Nível 2
Pensamento computacional	
[O1] Entender e usar os passos básicos para a solução de problemas algorítmicos (declaração e exploração do problema, identificação de exemplos, projeto, implementação e testes). **[O2]** Desenvolver um entendimento simples de um algoritmo (busca, sequência de eventos ou ordenação/classificação) usando exercícios sem o uso de computador. **[O3]** Demonstrar como uma cadeia de bits pode ser usada para representar informação alfanumérica. **[O4]** Descrever como uma simulação pode ser usada para resolver um problema. **[O5]** Fazer uma lista de subproblemas para serem considerados enquanto resolve um problema maior. **[O6]** Entender as conexões entre ciências da computação e outros campos.	**[O1]** Usar os passos básicos de algoritmos para a resolução de problemas ao projetar soluções (p.ex., declaração e exploração do problema, examinação de exemplos, design, implementação de uma solução, testes, avaliação). **[O2]** Descrever o processo de paralelização na forma que se refere à resolução de problemas. **[O3]** Definir um algoritmo, como sendo uma sequência de instruções que podem ser processadas por um computador. **[O4]** Avaliar formas em que algoritmos diferentes podem ser utilizados para resolver o mesmo problema. **[O5]** Dramatizar algoritmos de busca e ordenação. **[O6]** Descrever e analisar uma sequência de instruções a ser seguida (p.ex., descrever o comportamento de um personagem em um videogame, dirigido por regras e algoritmos). **[O7]** Representar dados em maneiras diferentes, incluindo texto, sons, imagens e números. **[O8]** Usar representações visuais de estados de problema, estruturas, e dados (p.ex., gráficos, tabelas, diagramas de rede, fluxogramas). **[O9]** Interagir com modelos específicos de conteúdo e simulações (p.ex., ecossistemas, epidemias, dinâmica molecular) para apoiar a aprendizagem e pesquisa.

	[O10] Avaliar que tipos de problemas podem ser resolvidos usando modelagem e simulação. **[O11]** Analisar o grau em que um modelo de computador representa, com precisão, o mundo real. **[O12]** Fazer uso da abstração para decompor um problema em subproblemas. **[O13]** Compreender a noção de hierarquia e abstração em computação, incluindo linguagens de alto-nível, tradução (p. ex., interpretar o mesmo problema de diferentes modos), conjunto de instruções, e circuitos lógicos. **[O14]** Examinar conexões entre elementos da matemática e ciência da computação, incluindo números binários, lógica, conjuntos e funções. **[O15]** Fornecer exemplos de aplicações interdisciplinares do pensamento computacional.
Colaboração	
[O7] Usar ferramentas tecnológicas de produtividade (p.ex., processador de texto, planilha, programa de apresentação) para atividades de publicação, comunicação e escrita colaborativa e individual. **[O8]** Usar recursos *on-line* (p.ex., e-mail, discussões *on-line*, ambientes de colaboração web) para participar em atividades de resolução colaborativa de problemas com o propósito de desenvolver soluções ou produtos. **[O9]** Identificar modos em que o trabalho em equipe e a colaboração podem apoiar a resolução de problemas e a inovação.	**[O16]** Aplicar ferramentas e periféricos de produtividade/multimídia para colaboração em grupo e para apoiar a aprendizagem ao longo do currículo. **[O17]** Colaborativamente criar, desenvolver, publicar e apresentar produtos (p.ex., vídeos, *podcasts*, sites), utilizando recursos tecnológicos que demonstram e comunicam conceitos do currículo. **[O18]** Colaborar com colegas, especialistas e outros utilizando práticas colaborativas como programação em pares, trabalho em equipes de projeto, e participação em atividades de aprendizagem ativa em grupo. **[O19]** Exibir disposições necessárias para colaboração: fornecer feedback útil e integrante, compreender e aceitar múltiplas perspectivas, socialização.
Programação (Prática da computação)	
[O10] Usar recursos tecnológicos (p.ex., calculadoras, sondas para coleta de dados, dispositivos móveis, vídeos, software educacional, e ferramentas web) para resolver problemas e aprender sozinho. **[O11]** Usar ferramentas e periféricos de produtividade e de propósito geral para apoiar a produtividade pessoal, remediar deficiências de habilidade e facilitar a aprendizagem. **[O12]** Usar ferramentas tecnológicas (tais como autoria de texto e multimídia, apresentação, ferramentas web, câmeras digitais e scanners) para escrita colaborativa e individual, comunicação e atividades de editoração/publicação. **[O13]** Obter e manipular dados usando uma variedade de ferramentas digitais. **[O14]** Construir um programa como um conjunto de instruções passo a passo para serem executadas (p.ex. a montagem de um sanduíche de pasta de amendoim e geleia). **[O15]** Implementar soluções de problemas usando uma linguagem de programação visual baseada em blocos. **[O16]** Usar dispositivos computacionais para	**[O20]** Selecionar ferramentas e recursos tecnológicos apropriados para realizar tarefas variadas e resolver problemas. **[O21]** Usar uma variedade de ferramentas e periféricos de multimídia para apoiar a produtividade e aprendizagem pessoal durante todo o currículo. **[O22]** Conceber, desenvolver, publicar e apresentar produtos (p.ex., páginas web, aplicações móveis, animações) usando recursos de tecnologia que demonstram e comunicam os conceitos do currículo. **[O23]** Demonstrar uma compreensão de algoritmos e a sua aplicação prática. **[O24]** Implementar soluções de problema utilizando uma linguagem de programação, incluindo: o comportamento de laços (sequências de instruções que se repetem), instruções condicionais, lógica, expressões, variáveis e funções. **[O25]** Demonstrar boas práticas na segurança da informação pessoal, usando senhas, encriptação e transações seguras. **[O26]** Identificar carreiras interdisciplinares que são abrangidas pela ciência da computação. **[O27]** Demonstrar receptiva disposição para resolver e programar problemas indeterminados (p.ex. conforto com complexidade, persistência, brainstorming, adaptabilidade, paciência, tendência a mexer, criatividade, aceitação de

acessar informação remota e comunicar-se com outros para apoio ao aprendizado independente e direto e ao atendimento de interesses pessoais. **[O17]** Navegar entre páginas web usando hiperlinks e realizar buscas simples usando motores de busca. **[O18]** Identificar uma ampla variedade de trabalhos que exigem conhecimento ou uso de computação. **[O19]** Obter e manipular dados usando uma variedade de ferramentas digitais	mudanças). **[O28]** Coletar e analisar dados que correspondem à saída de múltiplas execuções de um programa de computador.
Computadores e dispositivos de comunicação	
[O20] Demonstrar um nível apropriado de proficiência com teclados e outros dispositivos de entrada e saída. **[O21]** Entender a onipresença dos computadores e da computação no dia a dia (p.ex., mensagem por voz, baixar arquivos de áudio e de vídeo, fornos micro-ondas, termóstatos, internet sem fio, dispositivos de computação móveis, sistemas GPS). **[O22]** Pôr em prática estratégias para identificar problemas simples de hardware e de software que podem ocorrer durante o uso. **[O23]** Identificar que informações são trazidas para o computador de variadas fontes por meio da rede. **[O24]** Identificar fatores que distinguem humanos de máquinas. **[O25]** Reconhecer o comportamento inteligente dos modelos computacionais (perceptível em robótica, reconhecimento de fala e de linguagem e animação de computador).	**[O29]** Reconhecer que os computadores são equipamentos que executam programas. **[O30]** Identificar uma variedade de dispositivos eletrônicos que contêm processadores computacionais. **[O31]** Demonstrar compreensão sobre a relação entre hardware e software. **[O32]** Usar terminologia adequada ao desenvolvimento e, precisa na comunicação sobre tecnologia. **[O33]** Aplicar estratégias para identificar e resolver problemas de rotina de hardware que ocorrem no uso de computador diariamente. **[O34]** Descrever os principais componentes e funções de sistemas de computadores e redes. **[O35]** Descrever o que distingue os seres humanos de máquinas, dando um enfoque na inteligência humana contra a inteligência de máquina e, formas que podemos nos comunicar. **[O36]** Descrever maneiras em que os computadores usam modelos de comportamento inteligente (p.ex., movimento de robô, fala e compreensão da linguagem e, visão computacional).
Impactos éticos, globais e na comunidade	
[O26] Discutir questões básicas relacionadas ao uso responsável da tecnologia e da informação e as consequências do uso inadequado. **[O27]** Identificar o impacto da tecnologia (p.ex., redes sociais, *bullying* cibernético, comunicação e computação móvel, tecnologias web, segurança cibernética e virtualização) na vida pessoal e na sociedade. **[O28]** Avaliar a exatidão, a relevância, a propriedade, a qualidade de compreensão e a tendenciosidade que ocorre em fontes eletrônicas de informação. **[O29]** Entender questões éticas relacionadas aos computadores e às redes (p.ex., justiça de acesso, segurança, privacidade, direitos autorais e propriedade intelectual).	**[O37]** Apresentar comportamentos legais e éticos no uso de informação e tecnologia e, discutir as consequências do uso indevido. **[O38]** Demonstrar conhecimento das mudanças nas tecnologias de informação ao longo do tempo e os efeitos destas mudanças na educação, no local de trabalho e na sociedade. **[O39]** Analisar os impactos positivos e negativos da computação na cultura humana. **[O40]** Avaliar a precisão, relevância, adequação, abrangência, e viés de fontes de informação eletrônicas referentes a problemas do mundo real. **[O41]** Avaliar a precisão, relevância, adequação, abrangência, e viés de fontes de informação eletrônicas referentes a problemas do mundo real. **[O42]** Discutir como a distribuição desigual de recursos de computação em uma economia global levanta questões de equidade, acesso e poder.

Apesar de haver iniciativas para se ensinar computação logo no Ensino Fundamental, não existe horário disponível para inserção de mais uma disciplina separada. O tempo é

sempre colocado como um problema a ser enfrentado pela equipe escolar. As diretrizes e bases da educação nacional são reguladas pela Lei 9.394/96 (Lei de Diretrizes e Bases da Educação – LDB). Segundo o Artigo 31, Inciso II, da LDB, a carga horária mínima anual é de "oitocentas horas, distribuída por um mínimo de duzentos dias de trabalho educacional".

A grade é dividida em um núcleo comum (Língua Portuguesa, História, Geografia, Matemática, Ciências, Educação Física e Arte) e um núcleo diversificado, que inclui uma segunda língua moderna e demais disciplinas. A partir do 5º ano, a escola é obrigada a oferecer aulas de segunda língua (LDB, Artigo 26, § 5º). A maioria opta pelo inglês. O ensino religioso é facultativo (LDB, Artigo 33). As aulas tipicamente ocorrem de segunda à sexta com 5 aulas por dia, totalizando 25 aulas durante a semana.

Outra questão que complica a inserção da computação como disciplina no Ensino Fundamental é a falta de professores capacitados nesta área, uma das causas disto é o número de concluintes em cursos de formação de professor de computação ser muito pequeno para a demanda existente (INEP, 2010 a 2014). Embora muitas escolas possuam tecnologias, as mesmas muitas vezes não são utilizadas no potencial que deveriam, ficando os laboratórios/salas de informática trancadas por falta de profissionais habilitados para atuarem na gerência, supervisão, orientação e manutenção das máquinas, assim como por falta de capacitação de alguns docentes para o conhecimento das potencialidades que este recurso pode agregar nas atividades em sala de aula.

Cabe ressaltar que em algumas escolas é ensinado o uso do computador, abordando tópicos como: instalação de programas, criação e uso de e-mail, formatação de trabalhos, entre outros. Porém, não são ensinados conceitos referente à computação em si. Trata-se apenas de uma informatização do processo de ensino.

2.3 APRENDER COM O DESENVOLVIMENTO DE JOGOS

Uma das formas de se ensinar computação é por meio do uso de jogos como objeto de aprendizagem a ser desenvolvido pelos alunos. Desenvolver jogos, antes de mais nada, é bem diferente de jogar um jogo. O desenvolvimento de um jogo envolve uma série de etapas como a concepção do jogo, incluindo: quais são os objetivos do jogo, qual a condição de vitória/derrota, quais são as regras, entre outros (veja Tabela 5); a programação do jogo em si fazendo uso de conceitos de computação, os testes a serem feitos inúmeras vezes, etc.

Tabela 5 — Elementos comuns de um jogo (WANGENHEIM e WANGENHEIM, 2012).

Elemento	Descrição
Desafio/competição/ conflito	Uma proposta que envolve fazer algo difícil, ou complexo, podendo ser em forma de competição entre duas ou mais pessoas.
Interação/cooperação	É quando ocorre envolvimento de dois ou mais jogadores empenhados a trabalhar juntos cuja ação de um provoca uma reação no outro.
Narrativa	Refere-se à estória do jogo. Alguns jogos não possuem nenhum tipo de narrativa.
Objetivos	Está relacionado aos objetivos que um jogador tem ao jogar um jogo. Por exemplo, finalizar no menor tempo ou como primeiro, alcançar o nível mais elevando de proficiência, ou, ser simplesmente o melhor entre vários competidores.
Regras & restrições	As regras definem o que pode e o que não pode ser feito dentro de um jogo. As restrições podem definir que ações um jogador pode executar, quando certas condições forem cumpridas.
Resultados, recompensas e *feedback*	As recompensas são dadas em forma de prover aos jogadores o acesso a novos recursos ou permitir, que os jogadores façam coisas que antes não eram possíveis. O *feedback* é um elemento que aparece com frequência nos jogos, mostrando o resultado obtido após uma ação qualquer de um jogador.

Segundo Kafai, a maioria das crianças gosta de jogar jogos de acordo com regras estabelecidas tanto quanto elas também gostam de modificar e criar suas próprias regras:

> Turkle apontou um paralelo interessante entre as atrações de jogar jogos de computadores e programá-los. Ela viu a programação como uma maneira para as crianças construírem seus próprios mundos. Dentro deste contexto, as crianças podiam determinar as regras e limites que regem o mundo do jogo e se tornarem os criadores e os jogadores de seus próprios jogos. Em contraste, quando as crianças jogam um jogo de computador, elas estão sempre a jogar um jogo programado por outra pessoa; elas estão sempre a explorar o mundo de outra pessoa e decifrar mistérios de outra pessoa. Turkle viu que o que ela chamou de poder de fixação de jogar jogos poderia também ser aplicado para a criação ou a programação de jogos. (KAFAI, 2001, p. 3).

O jogo pode ser definido como uma atividade física ou mental, a qual é organizada por um sistema de regras (HAIDT, 2001). No jogo tem-se a figura do jogador (aquele que pratica) e regras específicas para um ambiente. Segundo Huizinga a ação de jogar é definida como:

> Uma atividade voluntária exercida dentro de certos e determinados limites de tempo e espaço, segundo regras livremente consentidas, mas absolutamente obrigatórias, dotado de um fim em si mesmo, acompanhado de um sentimento de tensão e alegria e de uma consciência de ser diferente de vida cotidiana. (HUIZINGA, 1993, p. 33).

Os jogos são elementos presentes em todas as culturas e estão muito ligados ao desenvolvimento humano. O surgimentos e desenvolvimento de jogos surgiram de acordo com

o potencial intelectual e subjetivo do homem (HUIZINGA, 1993). Os jogos eletrônicos ou digitais são uma das formas mais comum de jogos no século XXI. Entre os elementos típicos de um jogo digital tem-se diversos fatores como *game play*, estilo de arte, interatividade, gênero, entre outros.

Os gêneros são utilizados para classificar os jogos baseados em sua jogabilidade, contudo, não há uma padronização aceita de taxonomias de gêneros de jogos. A questão do gênero específico de um jogo é aberta à interpretação pessoal, sendo que um jogo pode pertencer a vários gêneros ao mesmo tempo, além disso, a cada novo jogo tendo pequenas mudanças, os gêneros tendem a se transformar lentamente ou até mesmo radicalmente (KONZACK, 2014). Desta forma, a Tabela 6 apresenta uma descrição baseada em Herz (HERZ, 1997) com algumas adaptações. O sistema de Herz é semelhante ao utilizado pela indústria de jogos, distinguindo jogos de ação, jogos de aventura, jogos de *role-playing*, entre outros.

Tabela 6 — Gênero de jogos (adaptado de HERZ, 1997).

Gênero	Descrição
Ação	Um jogo que exigem que os jogadores utilizem reflexos rápidos, com precisão para superar obstáculos, resolver desafios.
Aventura	Um jogo em que o jogador assume o papel de protagonista da estória e interage por meio da exploração e a solução de desafios. O jogador também pode acompanhar uma história por meio de textos, músicas e imagens.
Estratégia	Um jogo que enfatiza habilidades de pensamento e planejamento para alcançar a vitória.
Quiz	Um jogo em que o jogador precisa responder perguntas para uma determinada área de conhecimento.
RPG	Um jogo em que o jogador controla ações de um protagonista e com este personagem vive imerso em um mundo fictício. No jogo de RPG (*role-playing-game*) os personagens interagem com este mundo e ficam mais fortes.
Simulação	Um jogo desenvolvido para colocar o jogador no controle de um determinado ambiente ou atividade, o qual busca ser o mais realista possível.

Assim, desenvolver um jogo é um método efetivo de aprender conceitos de computação e aplicá-los por meio do uso de uma linguagem de programação (KAFAI, 2001). Na criação de um jogo, todas as decisões de desenvolvimento devem ser feitas pelo aluno, isto faz com que se comece a desenvolver a fluência digital (en: *digital fluency)* (KAFAI, 2001).

2.4 AMBIENTE E LINGUAGEM DE PROGRAMAÇÃO SCRATCH

Usar recursos tecnológicos para o ensino, não significa usar quaisquer ferramentas. Para que haja uma garantia de aprendizagem sobre o conteúdo é fundamental que exista um ambiente de aprendizagem no qual os alunos possam ter iniciativas, problemas a resolver, possibilidade de testar e verificar os erros e criar soluções pessoais (MEC; PCN, 1998).

Scratch é uma linguagem gráfica de programação criada no MIT (*Massachusetts Institute of Technology*). É inspirada em linguagens de programação para jovens como: *LOGO* e *Squeak Etoys* (RESNICK, 2007). Apesar de ser baseado em linguagens direcionadas a crianças e jovens, o Scratch foi concebido para ser diferente de outros ambientes, isto é, mais simples, com maior facilidade na utilização e mais intuitivo (GUZDIAL, 2004).

O público-alvo do Scratch é composto por adultos e principalmente crianças que não possuem conhecimento prévio em outras linguagens e conceitos de programação (RUSK, RESNICK e MALONEY, 2006). O objetivo primário do Scratch é ajudar as crianças (já a partir dos 8 anos de idade) a desenvolver competências essenciais de aprendizagem do século XXI (RUSK, RESNICK e MALONEY, 2006). O ambiente Scratch permite que possam ser programados jogos, animações, entre outros programas interativos (veja Figura 5).

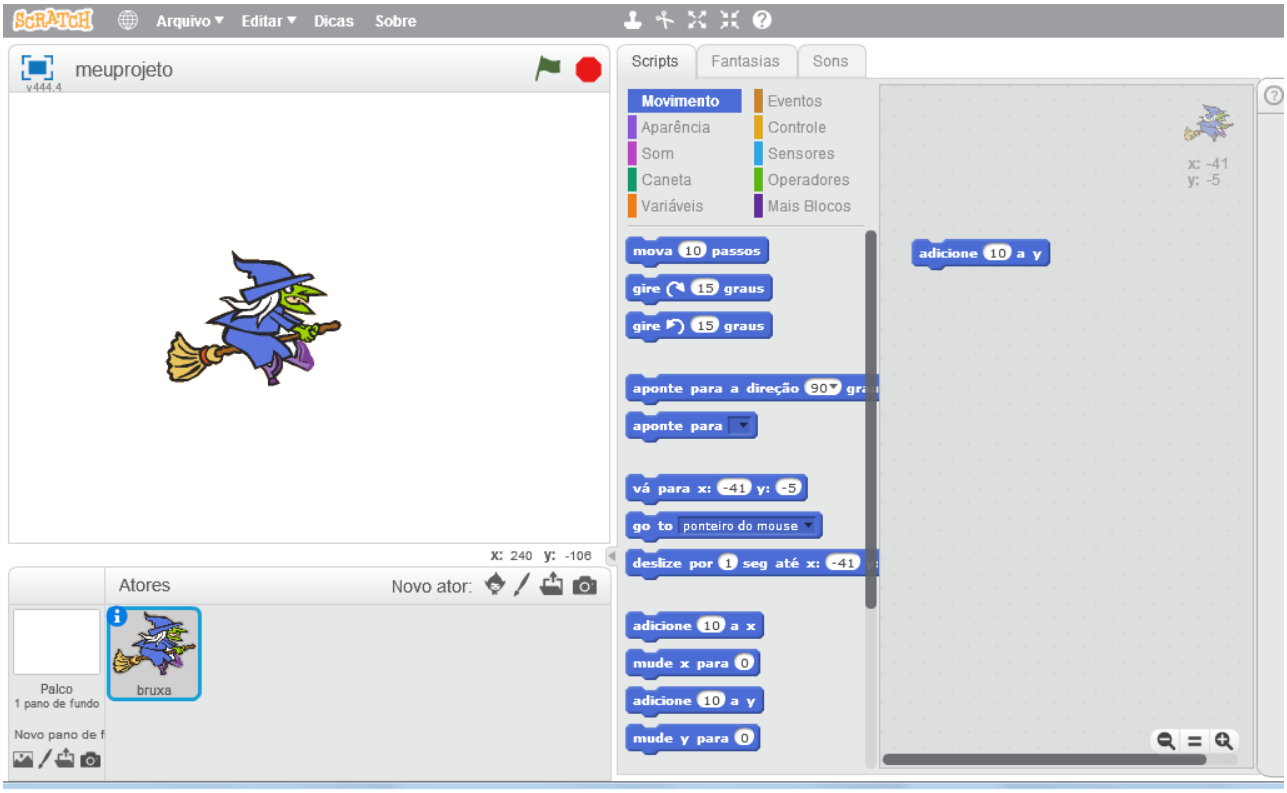


Figura 5 — Ambiente de programação Scratch off-line em português (SCRATCH, 2016).

O ambiente Scratch pode ser acessado de forma *on-line* ou *off-line*. O acesso *on-line* se dá por meio do próprio site[1] do Scratch utilizando-se um navegador *web* atualizado com o *plug-in* Adobe Flash Player. Para acessá-lo de forma *off-line*, é necessário o *download* e instalação previamente do programa Adobe Air e do Scratch Editor *Off-line*[2]. O programa *off-line*, na sua versão 2.0, é suportado pelos sistemas operacionais: Linux em algumas versões de 32 bits, Windows, e Mac nas versões 10 ou acima.

Terminologia do Scratch

Ao programar dentro do ambiente Scratch são usados alguns conceitos/elementos que não são claramente entendidos por aqueles que são novos neste ambiente. Isto acontece pois seus significados denotam características específicas do Scratch. As tabelas 7 e 8 apresentam uma lista destes conceitos com seus significados definidos para o contexto do ambiente Scratch.

2.4.1 Blocos de programação do ambiente Scratch

Os blocos utilizados na programação em Scratch são agrupados de acordo com as suas categorias: Movimento, Aparência, Som, Caneta, Dados, Eventos, Controle, Sensores, Operadores e Mais blocos. A Tabela 7 apresenta todos os blocos presentes em cada uma das categorias.

Tabela 7 — Categorias de blocos do Scratch.

Categoria	Blocos
Movimento	Blocos que modificam a posição e rotação do ator, etc. Exemplo: mova 10 passos.
Aparência	Blocos que modificam a fantasia, efeitos, tamanho, falas e controle de aparecimento/desaparecimento do ator, etc. Exemplo: diga “olá” por 2 segundos.
Som	Blocos para controlar quais sons devem tocar, em que velocidade e volume, etc. Exemplo: toque o som “meow”.
Caneta	Blocos para carimbar atores, riscar tudo que um ator faz na tela, etc. Exemplo: carimbe.
Variáveis	Comandos para criar variáveis ou listas. Contém blocos para fazer operações com elas, etc. Exemplo: adicione a “x” 1.
Eventos	Blocos para tratar eventos de iniciar, teclas do teclado pressionadas, cliques em atores, etc. Exemplo: quando este ator for clicado.
Controle	Blocos para controle de fluxo como laços, condicionais, etc. Exemplo: se então, senão.

[1] https://scratch.mit.edu/projects/editor/
[2] https://scratch.mit.edu/scratch2download/

Sensores	Blocos para uso de sensores de cor, posição do mouse, ruído, perguntas etc. Exemplo: pergunte “olá, tudo bem?” e espere a resposta.
Operadores	Blocos da lógica booleana, operações da aritmética, números aleatórios, operação modular, etc. Exemplo: escolha um número entre 1 e 10.
Mais blocos	Permite que o usuário crie seus próprios blocos. Podem ser passados parâmetros.

2.4.2 Terminologia do ambiente Scratch

Também são usados alguns termos que possuem significado específico dentro do contexto do ambiente Scratch. A Tabela 8 apresenta esses termos com seus respectivos significados.

Tabela 8 — Terminologia para o ambiente Scratch.

Termo	Descrição
Ator	Um objeto individual que pode mudar a aparência de acordo com as **fantasias**, emitir **sons** e executar outras ações em um **palco**.
Clone	Uma cópia de si mesmo criada por um **ator** em tempo de execução. Todos os **scripts** do **ator** original fazem parte do clone criado pelo mesmo.
Fantasias	Uma coleção de imagens que um **ator** ou **palco** pode assumir para modificar sua aparência.
Sons	Uma coleção de sons que um **ator** ou **palco** pode emitir.
Palco ou Cenário	Uma coleção de **roteiros**, **fantasias** e **sons** que fornecem o plano de fundo em que os **atores** executam.
Script ou Roteiro	Um ou mais **blocos** que descrevem a aparência, os **sons** e o comportamento de um **ator**, **palco** ou variável de um **projeto**.
Bloco	Um comando individual ou instrução usada dentro de um **roteiro**.
Centro ou Ponto de rotação	Ponto em que um **ator** permanece na mesma posição quando ele é rotacionado. Os valores “x” e “y” de um **ator** são determinados pelas coordenadas “x” e “y” no centro de rotação.
Projeto ou Programa	Uma coleção de **roteiros**, **atores**, **sons**, **palco**, e outras imagens, a qual é salva em um único arquivo e usa todos os seus elementos juntos.
Mochila	Local onde se pode copiar **roteiros** de outros ou do próprio **projeto** para usos futuros.
Compartilhar	Tornar público um projeto no site www.scratch.mit.edu.
Remix	Uma versão modificada e **compartilhada** de um projeto previamente compartilhado.
Estúdio	Lugar onde os usuários podem colocar/remover vários **projetos** em um grupo acessível.

O ambiente Scratch provê uma biblioteca padrão para atores, fantasias e sons, além de fornecer um “*Paint Editor*” e um “Som *editor*” ambos dentro do próprio ambiente para que o usuário também possa desenhar sua própria fantasia ou gravar e editar seu próprio som. No portal Scratch é possível compartilhar ideias além de projetos criados, isto é, pode-se publicar diretamente um projeto na *web*, copiar/guardar roteiros de outros projetos na mochila, comentar em outros projetos, remixar outros projetos e criar estúdios temáticos. Os remixes,

e todos os demais projetos, estão sempre sob a licença "*Creative Commons Attribution-Share Alike License*".

ScratchEd

ScratchEd[3] é um portal com o objetivo de auxiliar o uso do Scratch para ensinar computação. No portal podem ser encontrados relatos, materiais didáticos, ideias de experimentos, entre outros. No ScratchEd os educadores podem compartilhar histórias, trocar informações/recursos, fazer perguntas e encontrar outros educadores *on-line*.

Dr.Scratch

Dr.Scratch[4] é uma ferramenta de análise que avalia projetos feitos com Scratch em várias áreas da computação. A análise é feita de forma *on-line* no próprio *site*, sendo necessário apenas inserir a *url* do jogo ou carregar o jogo a partir de um arquivo no próprio computador. O Dr.Scratch classifica os jogos de acordo com as áreas: Lógica, Paralelismo, Interatividade com o usuário, Representação de dados, Controle de fluxo, Sincronização e Abstração. A Tabela 9 apresenta como é realizada a pontuação para cada uma destas áreas.

Tabela 9 — Pontuação das áreas de análise pelo Dr.Scratch (PROGRAMAMOS; DR.SCRATCH).

Área de análise	1 ponto	2 pontos	3 pontos
Lógica	Utilização do comando "Se, então".	Utilização do comando "Se, então; senão".	Utilização de comandos de operações lógicas.
Paralelismo	2 scripts iniciando com "bandeira verde".	2 scripts com o comando "quando a tecla for pressionada" utilizando a mesma tecla ou 2 scripts com o comando "quando este ator for clicado" utilizando o mesmo ator.	2 scripts de recebimento de mensagens ou criação de clones ou 2 scripts de sensores ou 2 scripts de mudança de pano de fundo.
Interatividade com o usuário	Utilização do comando da "bandeira verde".	Utilização de comandos de teclas/atores pressionadas, perguntas ao usuário, e botão do mouse.	Utilização de comandos de sensor de câmera e vídeo.
Representação dos dados	Modificação de propriedades dos atores como coordenadas, tamanho e aparência.	Operações com variáveis.	Operações com listas.
Controle de fluxo	Programação de uma sequência de blocos.	Utilização dos comandos "repita x vezes" e "sempre".	Utilização do comando "repita até que".
Sincronização	Utilização do comando "espere".	Utilização de comandos de envio/recebimento de	Utilização de comandos de "espere até" ou programação

[3] http://scratched.gse.harvard.edu/
[4] http://drscratch.programamos.es/

		mensagens.	de ações quando ocorre mudança de pano de fundo.
Abstração	Programação de mais de 1 script.	Utilização e programação de funções por meio do comando “defina”.	Utilização de clones.

2.5 ENSINO DE HISTÓRIA NO ENSINO FUNDAMENTAL

O ensino de História no Ensino Fundamental é fundamentado pela **LDB** (Lei de Diretrizes e Bases da Educação – Lei 9.394/96) e baseado nos **PCN** (Parâmetros Curriculares Nacionais criados pelo Ministério da Educação, 1998). Os PCN não são uma coleção de regras obrigatórias, mas sim uma referência de quais conteúdos devem ser abordados nas escolas de Ensino Fundamental. A LDB expressa o que se considera necessário transmitir aos alunos dentro da disciplina de História:

> O documento reitera a ênfase no estudo da história do Brasil, por meio da tríade: "as matrizes indígena, africana e europeia na formação do povo brasileiro", conforme exposto no Parágrafo 4º do Artigo 26 da LDB. (SILVA e FONSECA, 2010).

Espera-se que ao longo do Ensino Fundamental os alunos gradativamente possam ampliar a compreensão de sua realidade, especialmente confrontando-a e relacionando-a com outras realidades históricas, e, assim, possam fazer suas escolhas e estabelecer critérios para orientar suas ações (MEC; PCN 5.1, 1998).

Tabela 10 — Objetivos gerais de História (MEC; PCN 5.2, 1998).

Objetivos Gerais de História	
[O1]	Identificar relações sociais no seu próprio grupo de convívio, na localidade, na região e no país, e outras manifestações estabelecidas em outros tempos e espaços;
[O2]	Situar acontecimentos históricos e localizá-los em uma multiplicidade de tempos;
[O3]	Reconhecer que o conhecimento histórico é parte de um conhecimento interdisciplinar;
[O4]	Compreender que as histórias individuais são partes integrantes de histórias coletivas;
[O5]	Conhecer e respeitar o modo de vida de diferentes grupos, em diversos tempos e espaços, em suas manifestações culturais, econômicas, políticas e sociais, reconhecendo semelhanças e diferenças entre eles, continuidades e descontinuidades, conflitos e contradições sociais;
[O6]	Questionar sua realidade, identificando problemas e possíveis soluções, conhecendo formas político-institucionais e organizações da sociedade civil que possibilitem modos de atuação;
[O7]	Dominar procedimentos de pesquisa escolar e de produção de texto, aprendendo a observar e colher informações de diferentes paisagens e registros escritos, iconográficos, sonoros e materiais;
[O8]	Valorizar o patrimônio sociocultural e respeitar a diversidade social, considerando critérios éticos;
[O9]	Valorizar o direito de cidadania dos indivíduos, dos grupos e dos povos como condição de efetivo fortalecimento da democracia, mantendo-se o respeito às diferenças e a luta contra as desigualdades.

Antes de 2006, o Ensino Fundamental tinha duração de oito anos. O Governo Federal alterou a LDB, por meio da Lei 11.274/2006, ampliando a duração para nove anos. O Ensino Fundamental foi organizado e tratado em duas fases: 1º ao 5º ano e 6º ao 9º ano. Os PCN de História são organizados em 4 ciclos. O primeiro e segundo ciclos são direcionados aos 1º e 5º anos do Ensino Fundamental. O terceiro e quarto ciclos são direcionados aos 6º e 9º anos. Os conteúdos selecionados para a disciplina de História nos PCN, estão organizados por eixos temáticos (veja Tabela 11).

Tabela 11 — Ciclos e eixos temáticos de História no Ensino Fundamental (MEC; PCN Volume 1, 1998).

Ensino Fundamental	Ciclo	Eixo Temático
1º ao 5º ano	1º ciclo	**História local e do cotidiano**, subdividida em: a localidade e comunidade indígena.
	2º ciclo	**História das organizações populacionais**, subdividida em: deslocamentos populacionais, organizações e lutas de grupos sociais e étnicos, organizações políticas e administrações urbanas, e organização histórica e temporal.
6º ao 9º ano	3º ciclo	**História das relações sociais, da cultura e do trabalho**, subdividida em: as relações sociais, a natureza e a terra e as relações de trabalho.
	4º ciclo	**História das representações e das relações de poder**, subdividida em: nações, povos, lutas, guerras e revoluções e cidadania e cultura no mundo contemporâneo.

O ensino de História, inicialmente, é voltado para atividades aonde os estudantes possam compreender diferenças e semelhanças assim como permanências e transformações no estilo de vida social, cultural e econômico. Neste sentido, deve ser levado em conta a localidade onde o aluno está inserido, tanto no presente como no passado, mediante a leitura de diferentes obras humanas (MEC; PCN 5.1, 1998). Os objetivos detalhados para o segundo ciclo são apresentados na Tabela 12.

Tabela 12 — Objetivos de História no ciclo 2 (MEC; PCN Volume 5.1, p. 45, 1998).

Objetivos de História para o 2º ciclo	
[O1]	Reconhecer algumas relações sociais, econômicas, políticas e culturais que a sua coletividade estabelece ou estabeleceu com outras localidades, no presente e no passado;
[O2]	Identificar as ascendências e descendências das pessoas que pertencem à sua localidade, quanto à nacionalidade, etnia, língua, religião e costumes, contextualizando seus deslocamentos e confrontos culturais e étnicos, em diversos momentos históricos nacionais;
[O3]	Identificar as relações de poder estabelecidas entre a sua localidade e os demais centros políticos, econômicos e culturais, em diferentes tempos;
[O4]	Utilizar diferentes fontes de informação para leituras críticas;
[O5]	Valorizar as ações coletivas que repercutem na melhoria das condições de vida das localidades.

Os conteúdos de História para o segundo ciclo enfocam as diferentes histórias que compõem as relações estabelecidas entre a coletividade local e outras coletividades de outros tempos e espaços (MEC; PCN 5.1, 1998). Por exemplo, em Santa Catarina é estudada a cultura de populações indígenas que aqui viviam e os acontecimentos históricos referentes à colonização do estado. No segundo ciclo são contemplados diálogos entre presente e passado e os espaços locais, nacionais e mundiais:

> Como no primeiro ciclo, os questionamentos são realizados a partir do entorno do aluno, com o objetivo levantar dados, coletar entrevistas, visitar locais públicos, incluindo os que mantêm acervos de informações, como bibliotecas e museus.
>
> Valorizando os procedimentos que tiveram início no primeiro ciclo, a preocupação de ensino e aprendizagem no segundo ciclo envolve um trabalho mais específico com leitura de obras com conteúdo histórico, como reportagem de jornais, mitos e lendas, textos de livros didáticos, documentários em vídeo, telejornais. (MEC; PCN 5.1, 1998, p. 45).

Após o primeiro e segundo ciclos, os alunos já dominam algumas noções temporais e conhecem o calendário atual. Deve-se identificar os conhecimentos dos alunos e desenvolver trabalhos mais aprofundados sobre padrões de medida de tempo e respectivas histórias, para que possam, de modo autônomo, localizar fatos e sujeitos nas devidas épocas e, dessa forma, ao longo da escolaridade, aprenderem a discerni-los por critérios de anterioridade, posterioridade e simultaneidade (MEC; PCN 5.2, 1998).

Tabela 13 — Objetivos de História no ciclo 3 (MEC; PCN Volume 5.2, 1998, p. 54).

Objetivos de História para o 3º ciclo	
[O1]	Conhecer realidades históricas singulares, distinguindo diferentes modos de convivência nelas existentes;
[O2]	Caracterizar e distinguir relações sociais da cultura com a natureza em diferentes realidades históricas;
[O3]	Caracterizar e distinguir relações sociais de trabalho em diferentes realidades históricas;
[O4]	Refletir sobre as transformações tecnológicas e as modificações que elas geram no modo de vida das populações e nas relações de trabalho;
[O5]	Localizar acontecimentos no tempo, dominando padrões de medida e noções para distingui-los por critérios de anterioridade, posterioridade e simultaneidade;
[O6]	Utilizar fontes históricas em suas pesquisas escolares;
[O7]	Ter iniciativas e autonomia na realização de trabalhos individuais e coletivos.

Para o terceiro ciclo está é proposto o eixo temático **história das relações sociais, da cultura e do trabalho**. O eixo temático remete para o estudo de questões sociais relacionadas à realidade dos alunos; acontecimentos históricos e suas relações e durações no tempo;

discernimento de sujeitos históricos como agentes de transformações e/ou permanências sociais; abordagens históricas e suas aproximações e diferenças; e conceitos históricos e seus contextos.

Solicitam, por sua vez, atividades e situações didáticas que favoreçam a aprendizagem de procedimentos de pesquisa, observação, identificação, confrontação, distinção e reflexão; e de atitudes de comprometimento, envolvimento, respeito, ética, colaboração e amadurecimento moral e intelectual:

> No terceiro ciclo, os alunos já adquiriram tanto na escolaridade anterior quanto no convívio social um conjunto de informações e reflexões de caráter histórico. Assim, no processo de ensino e de aprendizagem, os professores devem considerar a importância de investigar o que é de domínio dos alunos e quais são as suas hipóteses explicativas para os temas estudados. (MEC; PCN 5.2, 1998, p. 53)

3. ESTADO DA ARTE

Neste capítulo é apresentado o estado da arte para unidades instrucionais interdisciplinares para ensinar computação no Ensino Fundamental. Para isto, é realizada uma Revisão Sistemática da Literatura (KITCHENHAM, 2004).

3.1 DEFINIÇÃO DA REVISÃO

A Revisão Sistemática da Literatura (**RSL**) tem como objetivo identificar, avaliar e interpretar pesquisas disponíveis por meio de critérios de qualificação claros e reproduzíveis em relação ao tema deste trabalho (KITCHENHAM, 2004). Esta RSL tem como questão central de pesquisa a seguinte pergunta: "**Existem unidades instrucionais para ensinar computação de forma interdisciplinar para crianças e jovens?**". Para responder essa pergunta, foram definidas as seguintes questões de pesquisa:

QP1. Quais são as outras áreas trabalhadas junto com a computação?

QP2. Quais são os objetivos de aprendizagem?

QP3. Qual o tipo de material didático?

QP4. Qual o objetivo da avaliação?

QP5. Como é feita a avaliação?

Critérios de inclusão e exclusão

Para responder a pergunta e questões de pesquisa foram levantados critérios de inclusão e exclusão para consideração de trabalhos existentes (veja Tabela 14).

Tabela 14 — Critérios de exclusão e de inclusão da RSL.

Critérios de inclusão	Critérios de exclusão
Tem como objeto de aprendizagem conceitos da computação incluindo a programação; Apresenta informações detalhadas de como foi realizado o estudo; e Tem como aprendizes crianças ou jovens no Ensino Fundamental.	Artigos duplicados; Artigos que abordam o processo de ensino-aprendizagem fazendo-se de uso de jogos educativos prontos; Artigos resumidos; e Artigos não relevantes (título, resumo ou palavras-chave não relacionadas aos objetivos desta RSL).

Fontes de dados

As pesquisas foram realizadas na web via a ferramenta de busca RESuLT (SALAZAR, 2015) para as bases acadêmicas e no site portal ScratchED[5] para relatos ou propostas de educadores. As bases pesquisadas foram IEEE Xplore[6], a ACM Digital Library[7], a Science Direct[8], a Scopus[9] e a SpringerLink[10].

Termos de busca

Com base na pergunta de pesquisa "Existem **unidades instrucionais (Id.1)** para **ensinar (Id.2) computação (Id.3)** de forma **interdisciplinar (Id.4)** para **crianças** e **jovens (Id.5)?**" foram derivados os termos unidade instrucional, ensino, computação, interdisciplinar e crianças. Com base no critério de inclusão "Deve ter como objeto de aprendizagem conceitos da computação incluindo a **programação (Id.6)"** foi derivado o termo programação. Também foram adicionados os sinônimos para os termos em português e traduções para inglês (veja Tabela 15).

Tabela 15 — Termos de busca.

Id	Termos	Sinônimos	Tradução (Inglês)
1	unidade instrucional	unidade didática	instructional unit, teaching unit
2	ensino	educação, aprendizagem	teaching, learning
3	computação	ciência da computação, cs	computer science, computing
4	interdisciplinar	pluridisciplinar	interdisciplinary
5	crianças	jovens	children
6	programação	-	computer programming, programming

3.2 EXECUÇÃO DA BUSCA

Primeira iteração

A primeira iteração da busca foi realizada com todos os termos apresentados na Tabela 15. Pela ferramenta RESuLT, a busca retornou em média 73 resultados por base. Foi lido o título, resumo, palavras-chave e procurado por informações referentes aos materiais de

[5] http://scratched.gse.harvard.edu
[6] http://ieeexplore.ieee.org
[7] http://dl.acm.org
[8] http://www.sciencedirect.com
[9] http://www.scopus.com
[10] http://www.springerlink.com

unidade, aplicação e tipo de avaliação realizada dos 50 primeiros resultados de cada base de dados entre 2005 e 2015. Verificou-se que nenhum destes atendia a todos os critérios de inclusão definidos.

Os resultados retornados mais próximos do que se esperava apresentavam relatos sobre o uso de jogos prontos para o ensino de outras áreas. Estes trabalhos não foram incluídos na revisão pela razão de estarem fora do escopo deste trabalho, aonde o jogo é um objeto a ser desenvolvido pelo aluno.

Já no portal ScratchED, a busca retornou 14 resultados, dos quais 2 foram escolhidos seguindo os critérios de inclusão/exclusão (veja Tabela 16). Os demais resultados apresentavam recursos e histórias relacionadas à computação, contudo, não proviam informações substanciais para uma análise mais detalhada.

Tabela 16 — Trabalhos encontrados na primeira iteração.

Id	Título do relato	Referência
1	*Embedding Scratch in US History/Geography*	Relato por Karen Randall (SCRATCHED; RANDALL, 2009)
2	*Introduction to Programming*	Relato por Amanda Wilson (SCRATCHED; WILSON, 2011)

Segunda iteração

Para a segunda iteração, foram retirados os termos referentes à unidade instrucional (Id.1) juntamente com seu sinônimo e traduções. A busca retornou em média 300 resultados por base. Nesta iteração, as buscas foram realizadas somente nas bases acadêmicas, haja vista trabalhos semelhantes já tinham sido encontrados no portal ScratchED durante a primeira iteração. Foi lido o resumo e procurado por informações referentes aos materiais de unidade, aplicação e tipo de avaliação dos 100 primeiros resultados, seguindo os critérios de inclusão foram escolhidos para análise 3 trabalhos (veja Tabela 17).

Tabela 17 — Trabalhos encontrados na segunda iteração.

ID	Título do trabalho	Referência
3	*An interdisciplinary approach to injecting computer science into the K-12 classroom*	(GOLDSCHMIDT, MACDONALD, *et al.*, 2011)
4	*Using App Inventor & History as a Gateway to Engage African American Students in Computer Science*	(JIMENEZ e GARDNER-MCCUNE, 2015)
5	*Animal tlatoque: attracting middle school students to computing through culturally-relevant themes*	(FRANKLIN, CONRAD, *et al.*, 2011)

3.3 EXTRAÇÃO DE INFORMAÇÃO

Nesta seção é apresentada a extração de informações de 5 resultados recuperados durante a busca descrita na seção anterior. Todos estes resultados atendem aos requisitos da seção 3.1. As informações levantadas sobre cada trabalho consistem em 3 tópicos principais: o **contexto**, a **unidade instrucional** e a **avaliação** da mesma. Cada tópico possui um conjunto de itens que descrevem objetivamente os trabalhos. A Tabela 18 apresenta a extração de informações referente ao contexto, com os seguintes itens:

- **ID**: identificador do artigo.
- **Ensino**: o que se quer ensinar, isto é, o que está sendo proposto em termos gerais.
- **País**: em qual país o estudo foi realizado.
- **Idade/Série**: que idade ou em que ano escolar encontram-se os alunos em que foi realizado o estudo.
- **Interdisciplinaridade**: com qual disciplina o estudo propõe a interdisciplinaridade.

Tabela 18 — Dados extraídos de trabalhos no estado da arte relacionados ao contexto.

ID	Ensino	País	Idade/Série	Interdisciplinaridade
1	Programar uma animação interativa sobre um tema referente a estudo sociais.	Estados Unidos	5ª série	História e Geografia
2	Ensinar conceitos de programação de computadores junto com alguns conceitos de matemática.	Escócia	8 anos	Matemática
3	Ensinar conceitos de programação de computadores junto com alguns conceitos de matemática.	Estados Unidos	3ª série	Matemática
4	Utilizar aspectos do pensamento computacional alinhado ao pensamento histórico para introduzir os alunos à ciência da computação dentro da disciplina de História.	Estados Unidos	Não informado	História
5	Usar a linguagem de programação Scratch para envolver os alunos na criação de animações sobre animais e cultura maia, permitindo-lhes uma experiência interdisciplinar que combina programação, cultura, biologia, arte e contagem de histórias.	Estados Unidos	Acima da 7ª série	Biologia, Arte e História.

A Tabela 19 apresenta a extração de informações referente a unidade instrucional, com os seguintes itens:

- **ID**: identificador do artigo.
- **Objetivos de aprendizagem**: que objetivos de aprendizagem são definidos para cada disciplina.
- **Material didático**: que tipo de material didático é utilizado, por exemplo: livros, slides.

- **Linguagem de programação**: que linguagem/ambiente de programação são utilizados para as atividade de programação, por exemplo: Scratch, AppInventor, Logo, etc.

Tabela 19 — Dados extraídos de trabalhos no estado da arte sobre unidades instrucionais.

ID	Objetivos de aprendizagem	Material didático	Linguagem de programação
1	**História** O1. O aluno compreenderá que grandes e diversas nações indígenas norte-americanas eram os habitantes originais da América do Norte; O2. O aluno irá demonstrar conhecimento da exploração europeia sobre o continente norte-americano e a interação resultante com nações indígenas americanas; O3. O aluno irá demonstrar conhecimento das colônias e dos fatores que moldaram a América do Norte colonial. **Geografia** O1. O aluno irá distinguir diferenças entre o uso e limitações dos diferentes tipos de mapas temáticos usados para descrever o desenvolvimento dos Estados Unidos; O2. O aluno irá utilizar a terminologia básica descrevendo as características físicas e culturais básicos de continentes estudados. **Computação** O1. Aprender conceitos básicos de programação usando o Scratch.	Livro de História e projetos modelos no Scratch	Scratch
2	**Matemática** O1. O aluno poderá comparar, descrever e mostrar as relações numéricas, usando vocabulário apropriado e os símbolos para igual, não igual, menor ou maior do que. **Computação** O1. O aluno explorará o software e usará o que aprender para resolver problemas e apresentar ideias, pensamentos ou informações; O2. Por meio da descoberta, curiosidade natural e imaginação, o aluno poderá explorar maneiras de construir modelos ou resolver problemas; O3. O aluno desenvolverá habilidades de estratégias, de navegação e de coordenação de resolução de problemas, brincando e aprendendo com jogos eletrônicos, controle remoto ou brinquedos programáveis.	Cartas Scratch	Scratch
3	**Matemática** O1. O aluno compreenderá o sentido da contagem e numeração. **Computação** O1. O aluno compreenderá conceitos de programação por meio do software Scratch	Não informado	Scratch
4	**História** O1. O aluno deve conhecer e fazer uso do pensamento histórico; O2. O aluno deve saber como montar um quadro sequencial temporal (*storyboard*). **Computação** O1. O aluno deve conhecer e fazer uso do pensamento computacional; O2. O aluno deve conhecer conceitos e praticar a programação.	Não informado	AppInventor
5	**Computação** O1. Prover uma experiência positiva sobre a exposição da ciência da computação; O2. Melhorar a habilidade dos participantes e autoconfiança dentro da computação; O3. Aumentar a possibilidade dos participantes escolherem uma carreira em ciência da computação.	Livros relacionados ao tema	Scratch

A Tabela 20 apresenta a extração de informações referente a unidade instrucional, com os seguintes itens:

- **ID**: identificador do artigo.
- **Objetivo da avaliação**: qual o objetivo principal da avaliação, por exemplo: avaliar a aprendizagem dos alunos, avaliar a motivação dos alunos antes e depois da aplicação sobre a computação ou avaliar a eficácia da unidade instrucional.
- **Tipo de estudo empírico**: que tipo de estudo não-experimental foi realizado, por exemplo: um pós-teste ou um pré-teste e pós-teste.
- **Número de participantes**: Qual a quantidade de alunos participou da aplicação do estudo.
- **Coleta de dados**: que instrumentos foram utilizados para coletar os dados, por exemplo: questionários, rubrica, entre outros.

Tabela 20 — Dados extraídos de trabalhos no estado da arte sobre avaliação de UI.

ID	Objetivo da avaliação	Tipo de estudo empírico	Número de participantes	Coleta de dados
1	Avaliar se os alunos conseguiram programar corretamente os projetos de cada aula.	Estudo de caso Pós-teste	Não informado	Análise do código produzido pelos alunos.
2	Avaliar a compreensão dos alunos sobre rotas, usando conceitos da matemática como graus e direções, maior e menor.	Estudo de caso Pós-teste	Não informado	Análise do código produzido pelos alunos.
3	Não informado	Não informado	Não informado	Não informado
4	Medir o grau em que se pode aproveitar o interesse dos alunos em tecnologia móvel para promover seu interesse na criação de aplicativos móveis, alinhando o pensamento computacional e histórico.	Estudo de caso Pré-teste e Pós-teste	30	Questionário pré-aplicação Questionário pós-aplicação.
5	Medição da autopercepção dos alunos sobre a sua aprendizagem e seus interesse na atividade em geral durante as aulas (no acampamento).	Estudo de caso Pré-teste e Pós-teste	34	Questionário pré-aplicação Questionário pós-aplicação.

3.4 DISCUSSÃO

Após a definição, realização da busca e extração de informações sobre os resultados encontrados, percebeu-se que apesar de existir vários trabalhos relacionados ao ensino de computação de forma interdisciplinar, poucos apresentam uma unidade instrucional pronta, já com todos os materiais instrucionais desenvolvidos para ser aplicada.

QP1. Quais são as outras áreas trabalhadas junto com a computação?

As áreas trabalhadas variam entre ciências exatas (Id.2 e Id.3 trabalham com Matemática), ciências biológicas (Id.5 trabalha com Biologia), ciências humanas (Id.1, Id.4 e Id.5 trabalham com História) e arte (Id.5 trabalha com Arte).

QP2. Quais são os objetivos de aprendizagem?

Todos os estudos encontrados possuem os objetivos de aprendizagem de computação. A maioria dos estudos possui também objetivos de aprendizagem de outra disciplina (Id.1, Id.2, Id.3 e Id.4 possuem objetivos para outra disciplina além da computação).

QP3. Qual o tipo de material didático?

Entre os materiais didáticos utilizados estão o livro da disciplina que se está trabalhando junto com computação, e “cartas Scratch” ou “projetos modelos” que apresentam comandos de como realizar determinada ação. Importante salientar que o estudo apresentado no artigo Id.4 não utiliza o Scratch como linguagem de programação e sim o **AppInventor**. Isso demonstra que existem outras alternativas de linguagens de programação para ensinar crianças e jovens.

QP4. Qual o objetivo da avaliação?

Foram identificados dois objetivos gerais de avaliação nos estudos. A maioria realizou a avaliação dos objetivos de aprendizagem, verificando se os objetivos foram atingidos pelos alunos (Id.1, Id.2, Id.4 e Id.5). Além disso, os artigos Id.4 e Id.5 também apresentam a avaliação da unidade em si, verificando se a mesma é motivadora em relação à computação, isto é, o objetivo da avaliação foi medir, também, se a aplicação do estudo teve um efeito motivador ou despertou interesse por parte dos alunos sobre a área da computação.

QP5. Como é feita a avaliação?

A maioria dos estudos foi realizada por meio de um estudo de caso com uma avaliação após a aplicação da unidade instrucional. Os estudos apresentados nos artigos Id.4 e Id.5 relatam que também foi realizada uma avaliação antes da aplicação, assim pôde-se realizar uma comparação entre os conhecimentos dos alunos antes e depois.

Outras considerações

Cabe salientar que os artigos Id.3 e Id.5 afirmam que abordar a computação de forma interdisciplinar motiva os alunos para a realização de atividades e os fazem considerar a computação como uma possível área para se investir futuramente. Notou-se também em todos os estudos que antes de se apresentar conceitos de computação aos alunos os temas interdisciplinares já haviam sido introduzidos previamente. Deste modo, quando chegado o momento em que os alunos devem programar, eles já possuem conhecimento suficiente para realizar as atividades interdisciplinares.

Conclusão

Esta RSL permite concluir que apesar de existirem unidades instrucionais para se ensinar computação no ensino fundamental, não foram encontradas unidades na língua portuguesa para o contexto do Brasil.

3.4.1 Ameaças à validade

As principais ameaças à validade da RSL são o tipo de seleção de publicações, cujos resultados positivos são mais prováveis de serem publicados do que resultados negativos (KITCHENHAM, 2004). Também é possível existir uma imprecisão nos dados extraídos, já que alguns dados tiveram que ser inferidos em consequência de não estarem claramente descritos.

A limitação de uma breve análise dos 50 primeiros resultados na primeira iteração e dos 100 primeiros na segunda iteração pode ter ocasionado uma perda de resultados relevantes. Para a segunda iteração a pesquisa foi realizada com menos termos, o que pode ter excluído resultados durante a recuperação de informação.

É difícil garantir que todos os estudos relevantes tenham sido coletados, há o risco de que algum estudo possa ter sido omitido também devido aos termos de busca utilizados. Para minimizar este risco, as pesquisas foram realizadas empregando recursos eletrônicos e fazendo busca em várias bases de dados, utilizando-se pesquisas experimentais e sinônimos. Deste modo, podem existir outros estudos relevantes, os quais não foram recuperados, já que podem estar publicados em outros locais diferentes dos pesquisados, por exemplo.

4. PESQUISA EXPLORATÓRIA

A pesquisa exploratória é muito utilizada para realizar um estudo preliminar do principal objetivo da pesquisa que será realizada. Ela permite a familiarização com o fenômeno que está sendo investigado, assim a pesquisa subsequente pode ser concebida com uma maior compreensão e precisão (YIN, 2001). A execução da pesquisa exploratória "investiga um fenômeno contemporâneo dentro do seu contexto da vida real, especialmente quando os limites entre o fenômeno e o contexto não estão claramente definidos" (YIN, 2001, p. 32).

Os objetivos desta pesquisa exploratória são: a análise, o desenvolvimento, a aplicação e a avaliação de uma unidade instrucional piloto para o ensino de computação de forma interdisciplinar no Ensino Fundamental. Para atingir este objetivo é realizado um estudo de caso exploratório para compreender os fenômenos observados durante a aplicação da unidade instrucional e identificar direcionamentos para melhorias na unidade. O resultado desta pesquisa proporciona *feedback* referente ao projeto e organização da unidade em si.

4.1 ANÁLISE DE CONTEXTO

De acordo com o design instrucional, são analisados o público alvo e o ambiente, para então se definir os objetivos de aprendizagem.

Análise do público alvo

Consiste em alunos do Ensino Fundamental de idade entre 8 e 14 anos. Tipicamente, a maioria dos alunos acima de 10 anos já possui conhecimentos e habilidades no uso de dispositivos eletrônicos devido ao acesso à internet (veja Figura 6). Os alunos tipicamente também já sabem usar computadores e dispositivos periféricos por meio do uso de dispositivos eletrônicos (celulares, computadores, *tablets*) em casa e também por meio de aulas de informática nas escolas com foco na utilização de TI (*IT literacy*).

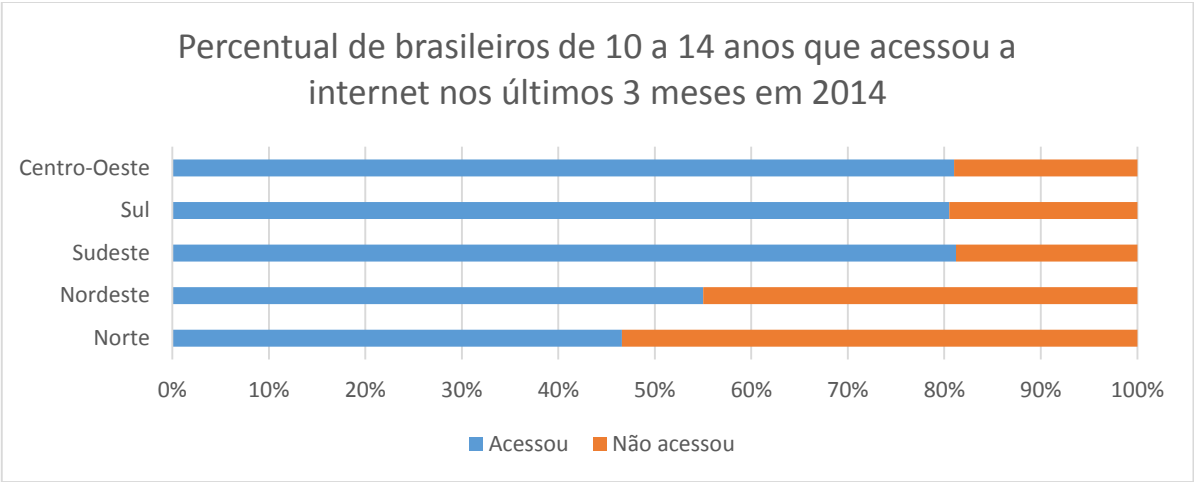


Figura 6 — Percentual de brasileiros de 10 a 14 anos que utilizaram a Internet, no período de referência dos últimos três meses, por Grandes Regiões - 2014 (IBGE, 2014).

Em termos de conhecimento geral, os alunos do Ensino Fundamental 2 já são alfabetizados e, nas escolas que optam pela língua estrangeira inglesa, os alunos também possuem domínio inicial em Inglês (LDB, 1996). Em relação à História, os alunos já dominam algumas noções temporais e têm conhecimento sobre a história local (MEC; PCN 5.2, 1998).

Análise do ambiente

A maioria das escolas possui computadores ou notebooks. As máquinas de escola pública normalmente possuem instalado o sistema operacional Linux Educacional numa escala maior do que o Microsoft Windows (veja Figura 7).

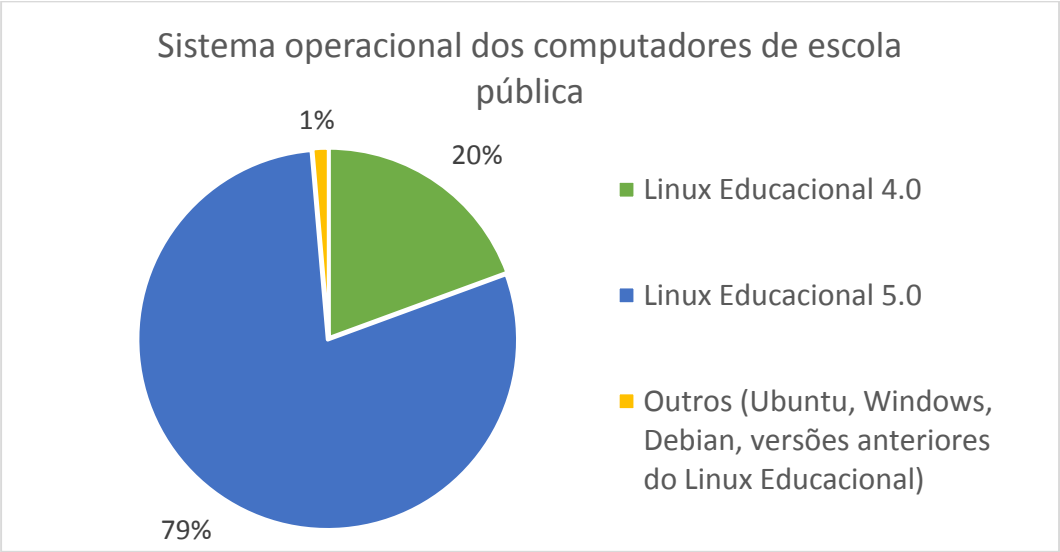


Figura 7 — Sistema operacional dos computadores da escola pública (PROINFO; MEC, 2016).

A maioria das escolas não possui professores de computação. Uma das causas é o número de concluintes em cursos de Formação de professor de computação (informática) ser muito menor do que, por exemplo, em cursos de formação de professores de matérias específicas (INEP, 2010 a 2014).

A Figura 8 apresenta uma comparação do número de concluintes de cursos de formação de professor de matemática e português com o de cursos formação de professor de computação (informática) do ensino superior no período 2010 a 2014. Vale ressaltar que se está considerando apenas cursos de licenciatura em computação, excluindo-se os cursos de bacharelado relacionados à computação como: Ciência da Computação e Sistemas de Informação.

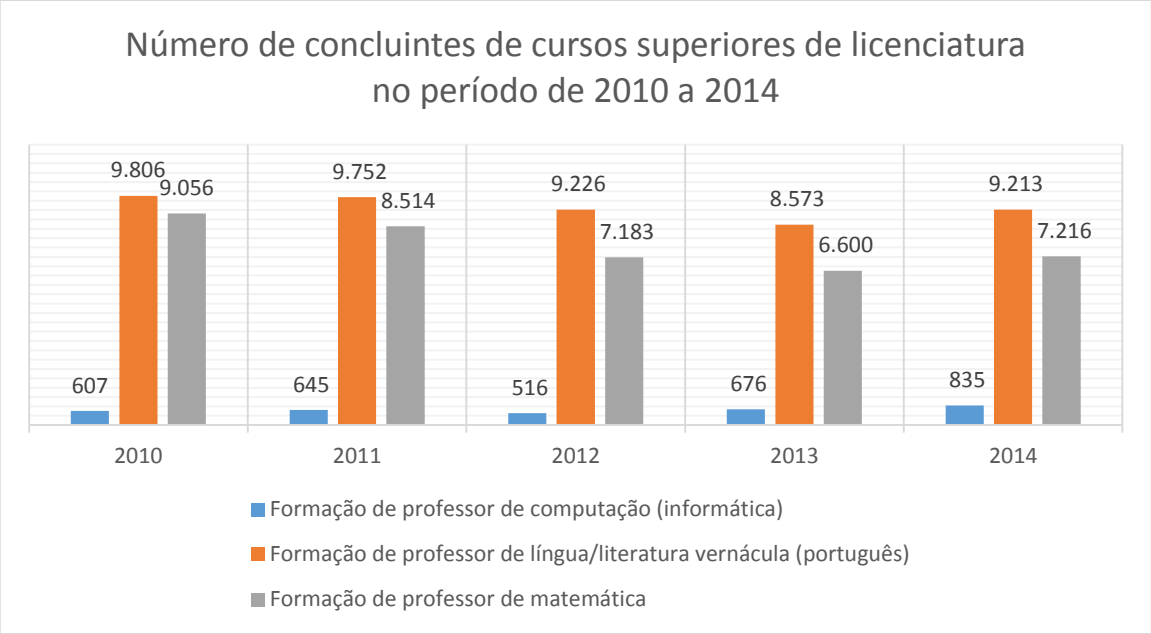


Figura 8 — Comparação entre o número de concluintes de cursos de matemática e português com informática (INEP, 2010 a 2014).

Nas escolas públicas, as aulas tipicamente possuem duração de 45 minutos e, no caso de 2 ou mais aulas por semana, estas podem ser realizadas em dias distintos. As turmas possuem em média 21 a 27 alunos (veja Tabela 21).

Tabela 21 — Média de alunos por turma no Ensino Fundamental (INEP, 2015).

Anos iniciais (1º ao 5º ano)	Anos finais (6º ao 9º ano)	Geral (1º ao 9º ano)
21,64	27	24

4.2 DESIGN INSTRUCIONAL UNIfICA v1.0 PILOTO

Nesta seção é apresentada a unidade instrucional **UNIfICA** (**UN**idade **I**nstrucional **I**nterdisciplinar de **C**omputação e Histórí**A**) na versão 1.0 piloto.

Objetivo geral de aprendizagem

Entender conceitos básicos da computação, principalmente relacionados à prática de computação/programação e ao pensamento computacional; usar o ambiente de desenvolvimento Scratch para criar jogos reforçando os conhecimentos sobre a realidade histórica em diferentes épocas.

Objetivos de aprendizagem específicos

1. Usar os passos básicos da solução de problemas algorítmicos/engenharia de software para projetar resoluções de problemas.
2. Descrever e analisar uma sequência de instruções a ser seguida.
3. Entender e aplicar conceitos de programação.
4. Usar o ambiente Scratch para criar, desenvolver e compartilhar um jogo de História
5. Colaborativamente criar, desenvolver, publicar e apresentar produtos usando recursos tecnológicos que demonstram e comunicam conceitos do currículo.
6. Demonstrar conhecimento sobre o modo de vida de diferentes grupos, em diversos tempos e espaços, em suas manifestações culturais e sociais, reconhecendo semelhanças e diferenças entre eles e seus conflitos.
7. Utilizar fontes históricas em suas pesquisas escolares.

Conteúdo

Aprender a utilizar o ambiente Scratch (entrar no site, criar programa, acessar programas, compartilhar programas). Conceber, programar e testar um jogo com Scratch ilustrando questões sobre civilizações europeias antigas ou cultura e história de Santa Catarina. Compartilhar e experimentar os jogos desenvolvidos pelos colegas. O sequenciamento do conteúdo é apresentado na Tabela 22.

Tabela 22 — Sequenciamento de conteúdo da UNIfICA v1.0 piloto.

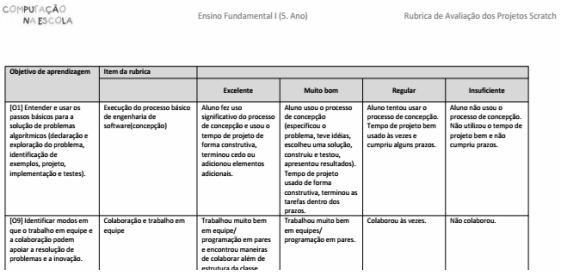
Encontro (2 horas)	Método instrucional	Recursos	Avaliação
Medição 1	-	Questionário aluno pré-unidade.	-
1.Introdução ao Scratch Programar um jogo exemplo com comandos básicos	Atividade prática passo a passo guiada pelo instrutor.	Guia do instrutor; Ambiente de programação Scratch.	-
2. Projeto de um jogo sobre um tema de História Concepção de jogo (Escolha do tipo e mecânicas do jogo e do tema de História).	Apresentação de jogos exemplos pelo professor; Cada grupo/dupla discute ideias sobre o projeto do jogo.	Jogos exemplos no Scratch; Ambiente de programação Scratch; Material instrucional das aulas de História.	-
3. Programação e testes do jogo	Prática de programação e teste do jogo pelos grupos/duplas; Professor disponível para tirar dúvidas.	Exemplos de jogos Scratch; Ambiente de programação Scratch; Material instrucional das aulas de História.	Rubrica para avaliar os projetos dos alunos.
4. Programação e testes do jogo			
5. Programação e testes do jogo			
6. Finalização dos projetos Compartilhamento, experimentação dos jogos da turma. Debate sobre a unidade	Jogos da turma são jogados pelos alunos; Discussão.	-	-
Medição 2	-	Questionário aluno pós-unidade; Questionário instrutor pós-unidade.	-

4.3 DESENVOLVIMENTO DE MATERIAIS PILOTO

De acordo com o plano de ensino definido são desenvolvidos vários materiais instrucionais. A Tabela 23 apresenta uma descrição e um extrato de cada material desenvolvido.

Tabela 23 — Materiais desenvolvidos para a UNIfICA v1.0 piloto.

Material	Descrição	Extrato
Guia do instrutor	Guia indicando passo a passo como explicar o desenvolvimento de um jogo com Scratch com comandos básicos, também incluindo a instalação do Scratch *off-line* e a criação de contas e projetos no site do Scratch.	COMPUTAÇÃO NA ESCOLA Guia do Instrutor – Jogo do Ratinho **6. Fazer o gato andar** Volte à aba **ROTEIROS.**

Jogos exemplos	Jogos desenvolvidos de diversos gêneros	
Rubrica	Documento que define os critérios de avaliação para medir o aprendizado dos alunos por meio dos jogos desenvolvidos.	
Questionários de avaliação	Questionários voltados à avaliação da unidade instrucional em termos de percepção, autoavaliação e motivação do aluno antes/depois da unidade instrucional e percepção do professor após a unidade instrucional.	

4.4 APLICAÇÃO PILOTO

A unidade instrucional piloto foi aplicada com duas turmas do 5º ano (5MAT e 5VES) e 2 turmas do 7º ano (7A/7B) do Ensino Fundamental da Escola Autonomia (www.autonomia.com.br) em Florianópolis/SC no primeiro semestre de 2015 (ALVES, VON WANGENHEIM, *et al.*, 2016). As aulas foram ministradas pelo professor da disciplina de História da escola Autonomia em conjunto com professores e alunos de computação do Departamento de Informática e Estatística (INE) da UFSC, no âmbito da iniciativa Computação na Escola (veja Figura 9).

Figura 9 — Alunos dos quintos e sétimos anos da escola Autonomia durante a aplicação da UNIfICA v1.0 piloto.

As aulas foram inseridas na carga horária da disciplina de História, presente no currículo escolar. Participaram da aplicação 105 alunos (veja Tabela 24).

Tabela 24 — Quantidade de alunos.

Ano	Turmas	Alunos
5º ano	5MAT e 5VES	45
7º ano	7A e 7B	60

Os alunos puderam formar equipes de dois ou três integrantes, para a realização das atividades. Seguindo o conteúdo abordado previamente nas aulas de História (7º ano) e estudos sociais (5º ano), os alunos do 5º ano desenvolveram jogos referentes à história/cultura de Santa Catarina (veja Figura 10 — lado direito) e os alunos do 7º ano referentes às Civilizações Antigas (veja Figura 10 — lado esquerdo).

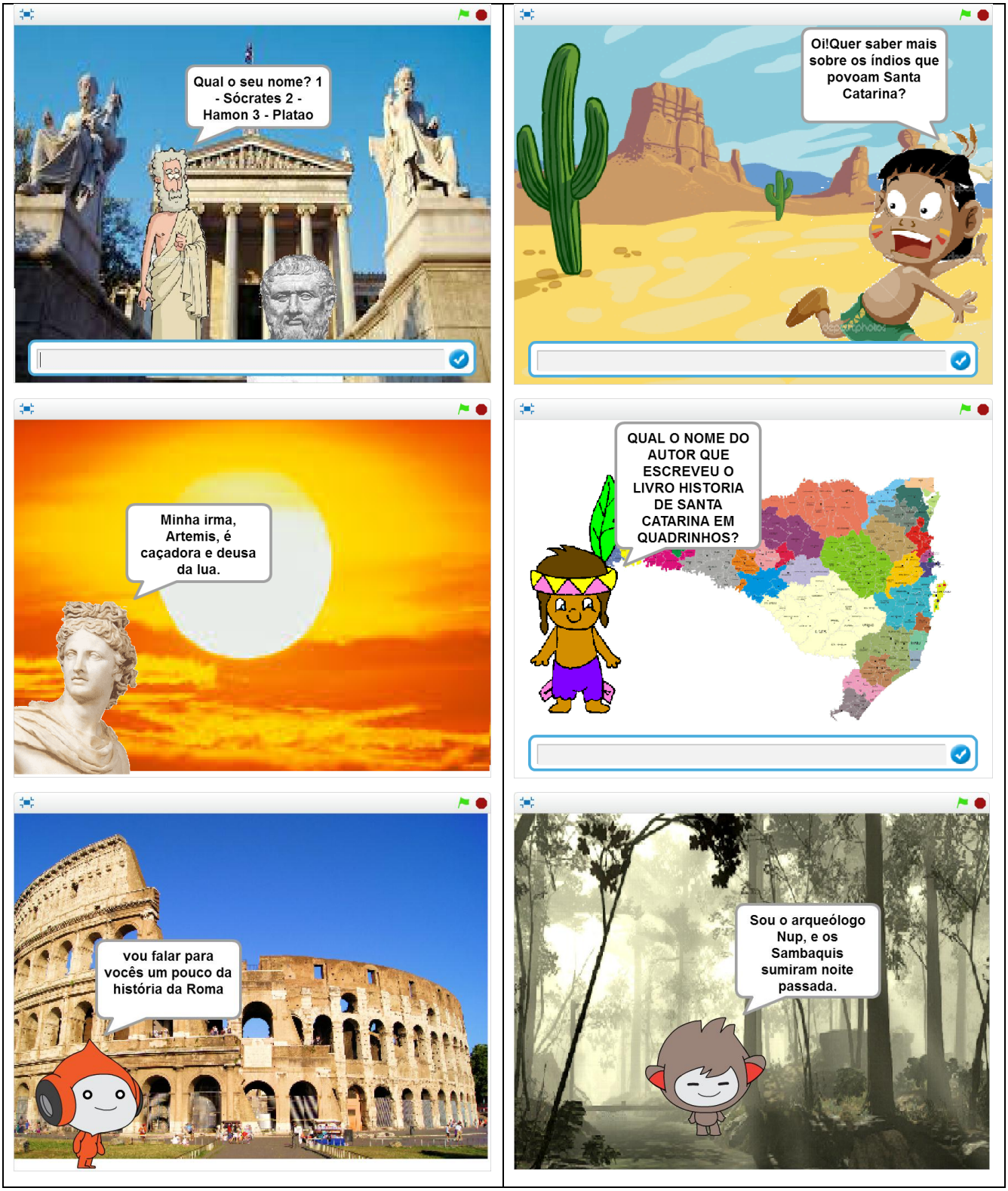


Figura 10 — Extrato de jogos desenvolvidos pelos alunos durante a unidade instrucional UNIfICA v1.0 piloto.

4.5 ANÁLISE DOS DADOS COLETADOS

Observando os alunos durante as aulas foi possível perceber que, em geral, todas as equipes conseguiram criar um jogo, sendo que a maioria dos alunos participou ativamente das seis aulas, programando, demonstrando entusiasmo e vontade de aprender (veja Tabela 25).

Tabela 25 — Atendimento dos objetivos da UNIfICA v1.0 piloto.

Objetivo	Análise
[OC1] Usar os passos básicos da solução de problemas algorítmicos/engenharia de software para projetar resoluções de problemas.	Todos os alunos atingiram o objetivo.
[OC2] Descrever e analisar uma sequência de instruções a ser seguida.	Todos os alunos atingiram o objetivo.
[OC3] Entender e aplicar conceitos de programação.	A maioria dos alunos atingiu o objetivo.
[OC4] Usar o ambiente Scratch para criar, desenvolver e compartilhar um jogo de história.	A maioria dos alunos atingiu o objetivo.
[OC5] Colaborativamente criar, desenvolver, publicar e apresentar produtos usando recursos tecnológicos que demonstram e comunicam conceitos do currículo.	A maioria dos alunos atingiu o objetivo.
[OH1] Demonstrar conhecimento sobre o modo de vida de diferentes grupos, em diversos tempos e espaços, em suas manifestações culturais e sociais, reconhecendo semelhanças e diferenças entre eles e seus conflitos.	A maioria dos alunos atingiu o objetivo.
[OH2] Utilizar fontes históricas em suas pesquisas escolares.	A maioria dos alunos atingiu o objetivo.

Os aluno puderam escolher o gênero dos jogos de forma livre. A maioria escolheu entre ação, aventura e quiz de acordo com a classificação do sistema de Herz (HERZ, 1997), o qual é parecido com o utilizado pelas indústrias de jogos em geral. Em alguns jogos pôde-se observar a presença de alguns elementos comumente encontrados em outros gêneros/subgêneros diferentes do escolhido, tornando-os híbridos. No total, foram desenvolvidos 41 jogos pelos alunos.

No ambiente Scratch praticamente todos os alunos utilizaram conceitos de programação durante a implementação dos jogos. Isto pode ser visto com mais detalhes no levantamento feito sobre quais conceitos foram implementados nos jogos (veja Tabela 26).

Tabela 26 — Contabilização da utilização de comandos/conceitos.

Comando/Conceito	Número total de jogos que utilizaram os comandos/conceitos de forma correta
Concorrência	39
Interatividade/Manipulação de eventos externos	32
Estruturas condicionais	28
Estruturas de repetição	23

Manipulação de eventos internos	22
Lógica booleana	21
Variáveis	19

Esses dados indicam que houve aprendizagem efetiva de conceitos de programação. Os alunos tiveram contato e usaram desde conceitos simples, como operações lógicas, até conceitos mais avançados como sincronização por meio da concorrência. Este efeito da aprendizagem também é percebido na autoavaliação dos alunos (veja Figura 11).

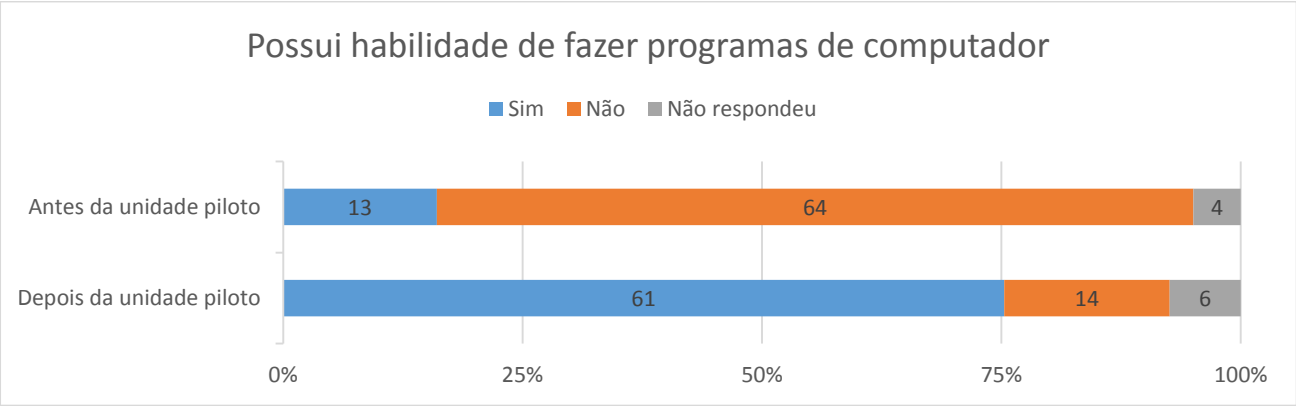


Figura 11 — Dados de antes/depois da aplicação da UNIfICA v1.0 piloto sobre habilidade de fazer programas de computador.

A maioria dos alunos achou as aulas muito fáceis ou fáceis. Poucos alunos consideraram as aulas difíceis e somente um aluno achou muito difícil. Isso, mesmo que em geral, os alunos consideram a programação em si não tão fácil (veja Figura 12).

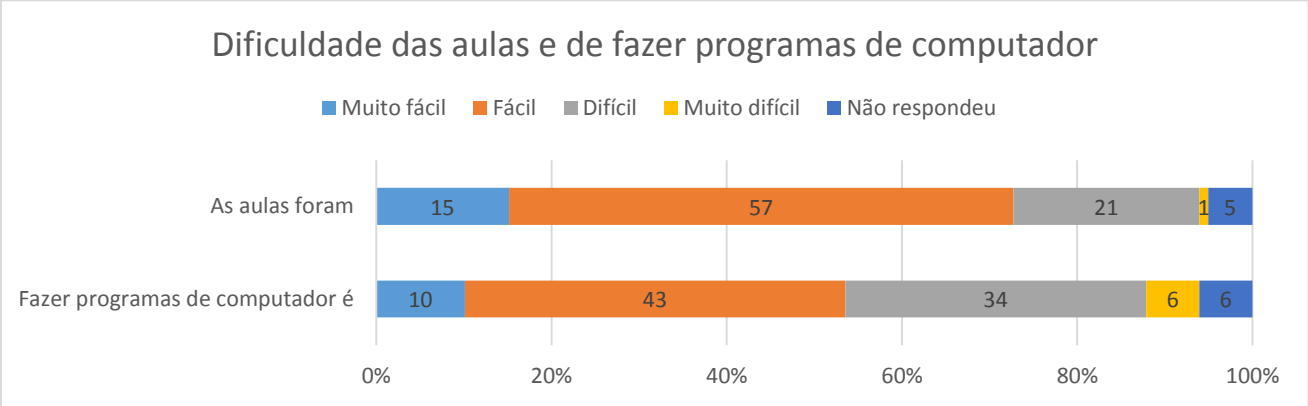


Figura 12 — Dados depois da aplicação da UNIfICA v1.0 piloto sobre dificuldade das aulas e de fazer programas de computador.

Percebeu-se que durante e após o final da unidade instrucional piloto grande parte dos alunos demonstraram vontade de continuar programando com Scratch (veja Figura 13). Isso

também foi percebido pelo professor da disciplina que observou que muitos dos alunos gostaram de fazer programas de computador.

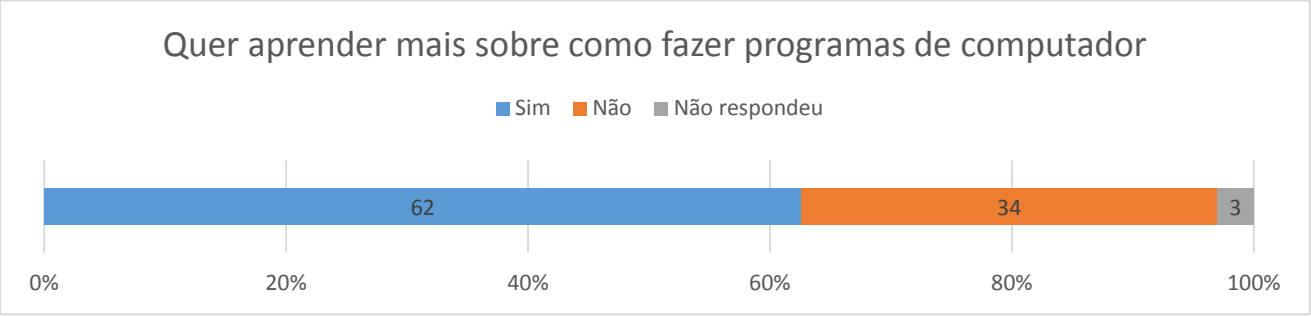


Figura 13 — Dados após a aplicação da UNIfICA v1.0 piloto.

Os resultados da aplicação piloto demonstram uma primeira indicação de que o ensino de computação usando Scratch pode ser adotado já no Ensino Fundamental de forma interdisciplinar com História. Esta pesquisa exploratória forneceu um *feedback* significante referente ao *design* instrucional e à sua estrutura e organização.

A partir deste *feedback*, a unidade instrucional é revisada e aprimorada significativamente, de forma que são desenvolvidos novos materiais e realizadas melhorias nos já existentes. O capítulo 5 apresenta a unidade UNIfICA v2.0 a qual é baseada na unidade piloto. Esta unidade é aplicada e avaliada sob diversos aspectos, incluindo a percepção do aluno em relação à computação, o atendimento dos objetivos de aprendizagem e a facilidade e qualidade em geral das aulas.

5. UNIDADE INSTRUCIONAL UNIfICA v2.0

Nesse capítulo é apresentada a unidade instrucional **UNIfICA** (**UN**idade **I**nstrucional **I**nterdisciplinar de **C**omputação e Histórí**A**) na versão 2, baseada na UNIfICA v1.0 piloto da pesquisa exploratória. O plano de ensino recebe mudanças e são desenvolvidos novos recursos instrucionais para cada aula.

A exemplificar, para a unidade piloto foi elaborado um plano de ensino com aulas de 2 horas, o que não ocorre nas escolas públicas de Ensino Fundamental pois as aulas tipicamente são de 45 minutos. Esta e outras modificações foram feitas com base nos resultados da pesquisa exploratória.

Objetivo geral de aprendizagem

Após esta unidade o aluno deverá ser capaz de aplicar conceitos básicos da computação relacionados à prática da computação e programação, colaboração e pensamento computacional de forma interdisciplinar com o conteúdo da disciplina de História do Ensino Fundamental.

Objetivos de aprendizagem específicos

Tabela 27 — Objetivos de aprendizagem específicos da unidade instrucional UNIfICA v2.0.

Objetivo	Referência
Compreender o que é um algoritmo e usar os passos básicos de algoritmos para a resolução de problemas ao projetar soluções. Descrever e analisar uma sequência de instruções a ser seguidas	Pensamento computacional (CSTA – Nível 2, 2011)
Colaborar com colegas em grupo para criar, desenvolver, publicar e apresentar projetos, utilizando Scratch.	Colaboração (CSTA – Nível 2, 2011)
Usar os passos básicos da engenharia de software (análise, projeto, implementação e teste) para criar o projeto de um jogo com tema de História utilizando conceitos básicos de programação (variáveis, operadores, estruturas condicionais e estruturas de repetição.	Práticas Computacionais e de Programação (CSTA – Nível 2, 2011)
Demonstrar conhecimento sobre o modo de vida de diferentes grupos, em diversos tempos e espaços, em suas manifestações culturais e sociais, reconhecendo semelhanças e diferenças entre eles e seus conflitos.	História (MEC; PCN 5.2, 1998)

5.1 DESIGN INSTRUCIONAL

Nesta seção são definidos o conteúdo, a sequência de apresentação e as estratégias instrucionais da unidade instrucional para ensinar computação na disciplina de História. A

unidade instrucional é desenvolvida de forma interdisciplinar, explorando a integração entre alguns dos conceitos práticos da computação (programação) e de conteúdos da disciplina de História.

Conteúdo

- Conhecer conceitos de computação e programação.
- Conhecer o Scratch (ambiente e linguagem de programação).
- Analisar, projetar, implementar, testar e compartilhar um jogo no Scratch, junto com o(a) instrutor(a), ilustrando um tema de História.
- Analisar, projetar, implementar, testar, compartilhar e apresentar um jogo no Scratch, em grupo, ilustrando questões sobre civilizações europeias antigas ou cultura e história de Santa Catarina ou pré-história.

Pré-requisitos para a aplicação da UNIfICA v2.0

Para a aplicação da UI, é indispensável o atendimento de alguns requisitos para que seja realizada com sucesso. Os pré-requisitos estão divididos em *hardware* — detalhes específicos de uma dada máquina, software — programas e Sistemas Operacionais que devem estar instalados nas máquinas, e outros — pré-requisitos que não se encaixam nas categorias anteriores.

- Hardware
    - Computador com monitor (mínimo de 14 polegadas), teclado e *mouse* – no máximo 3 alunos por computador.
    - Projetor multimídia – utilizado para apresentação de slides.
    - Acesso à internet – utilizado para criação, compartilhamento, apresentação de jogos exemplos no site do Scratch.
    - Opcional: Caixa de som/microfone – utilizado para apresentação de vídeos com som/criação de áudios.
- Software
    - Navegador Firefox ou Chrome <u>atualizado</u> – utilizado para acessar o ambiente Scratch *on-line*. Caso não esteja atualizado podem ocorrer problemas de acesso ao ambiente.
    - Sistema Operacional: Linux Educacional (versão 4 ou acima), Linux (qualquer

distribuição atualizada de 32 bits) Windows (XP ou acima) e Mac OS (versão 10 ou acima).

  - Opcional: Adobe Air e Scratch *off-line* instalado nos computadores – utilizado para acessar o ambiente Scratch *off-line*.

- Outros
  - Contas no ambiente Scratch *on-line* para cada grupo – criadas pelo professor.
  - Giz/Caneta e apagador – utilizado para escrever no quadro.

Sequenciamento da unidade instrucional

Tabela 28 — Sequenciamento da unidade instrucional UNIfICA v2.0.

Tempo	Tópico	Estratégia instrucional	Recursos didáticos	Avaliação
1ª aula (45 min)				
5 min	Apresentação de informações gerais do projeto (o que será feito, quantidade de aulas, equipe) e chamada.	Apresentação	Slides Introdução	--
10 min	M0. Medição Pré-unidade	Questionário	Questionário Pré-unidade Aluno	--
25 min	O que é computação? O que é um algoritmo, para que serve? O que é linguagem de programação? O que é o Scratch? Conceitos de linguagem de programação: variáveis, operadores, estruturas condicionais e estruturas de repetição	Discussão com os alunos Apresentação Laboratório de programação	Slides Introdução Opcional: Vídeo/Animação Quadro	Prova final
5 min	Divisão de grupos	Formação de grupos pelos alunos (máximo de 3 alunos por grupo)	Lista de grupos	--
2ª aula (45 min)				
5 min	Divisão de grupos e chamada.	Formação de grupos pelos alunos (máximo de 3 alunos por grupo)	Lista de grupos	--
10 min	Ciclo de Engenharia de Software e elementos de um jogo	Apresentação	Slides Criação de um jogo	Prova final
5 min	Distribuição de contas	Material impresso p/ cada grupo.	Lista de contas	--

10 min	Entrar no site do Scratch (todos os grupos)	Apresentação passo-a-passo	Lista de contas	--
15 min	Criação de um jogo: - Introdução ao ambiente de programação Scratch: Abrir o Scratch, Mudar o idioma para Português, Salvar o projeto dando um nome ao arquivo. - Incluir/remover novo ator Inserir/remover um ator – BRUXA e GATO.	Apresentação passo-a-passo Laboratório de programação	Tutorial Jogo da Bruxa	Participação na aula
3ª aula (45 min)				
5 min	Chamada	--	--	--
5 min	Entrar no site do Scratch (todos os grupos)	Apresentação passo-a-passo	Lista de contas	--
30 min	Criação de um jogo: - Criar função "Fazer a bruxa voar" - Localização da bruxa (Opcional) - Criar função "Fazer o morcego voar até a bruxa" - Adicionar uma variável: VIDA (criar variável, inicializar) - Adicionar uma variável: PONTOS (criar variável, inicializar) - Operações com variáveis: VIDA e PONTOS (adicionar valor, se senão) - Condição de término do jogo: identificar requisitos, criar algoritmo, programar, testar.	Apresentação passo-a-passo Laboratório de programação	Tutorial Jogo da Bruxa	Prova final Participação na aula
5 min	Explicação da tarefa de casa	Apresentação	Tarefa de casa	
4ª aula (45 min)				
5 min	Chamada	--	--	--
5 min	Entrar no site do Scratch (todos os grupos)	Apresentação passo-a-passo	Lista de contas	--
15 min	Término de um jogo: - Fazer o morcego falar - Colocar um fundo escolhendo da galeria do Scratch.	Apresentação passo-a-passo Laboratório de programação	Tutorial Jogo da Bruxa	Prova final Participação na aula

	- Fazer o upload no Scratch on-line (se estiver usando versão off-line) - Compartilhar e ver página do projeto.			
15 min	Apresentação de jogos exemplos, elementos de um jogo.	Apresentação de ideias Laboratório de programação	Slides Concepção de um jogo Jogos exemplos	Participação na aula
5 min	Distribuição de worksheet: Concepção do jogo.	Material impresso p/ cada grupo	Worksheet: Concepção do jogo (gêneros: ação, aventura ou *quiz* escolhidos pelo grupo) Material instrucional de história	Worksheet Prova final
5ª aula (45 min)				
5 min	Chamada	--	--	--
5 min	Entrar no site do Scratch (todos os grupos)	Apresentação passo-a-passo	Lista de contas	--
35 min	Concepção do jogo. Criar um projeto (inserir atores do jogo, planos de fundo, inicialização de atores).	Laboratório de programação Atividade em grupo Auxilio respondendo dúvidas dos alunos	Worksheet: Concepção do jogo (ação, aventura ou *quiz*) Material instrucional de história	Worksheet Prova final Projeto final
6ª aula (45 min)				
5 min	Chamada	--	--	--
5 min	Entrar no site do Scratch (todos os grupos)	Apresentação passo-a-passo.	Lista de contas	--
35 min	Continuação do projeto: programar função de atores/pano de fundo.	Laboratório de programação Atividade em grupo Auxilio respondendo dúvidas dos alunos	Worksheet: Concepção do jogo (ação, aventura ou *quiz*) Material instrucional de história	Worksheet Prova final Projeto final
7ª aula (45 min)				
5 min	Chamada	--	--	--
5 min	Entrar no site do Scratch (todos os grupos)	Apresentação passo-a-passo.	Lista de contas	--
35 min	Continuação do projeto: programar função de atores/pano de fundo.	Laboratório de programação Atividade em grupo. Auxilio respondendo dúvidas dos alunos	Worksheet: Concepção do jogo (ação, aventura ou *quiz*) Material instrucional de história	Worksheet Prova final Projeto final
8ª aula (45 min)				
5 min	Chamada	--	--	--
5 min	Entrar no site do Scratch (todos os grupos)	Apresentação passo-a-passo.	Lista de contas	

35 min	Continuação do projeto: programar função de atores/pano de fundo.	Laboratório de programação. Atividade em grupo Auxilio respondendo dúvidas dos alunos.	Worksheet: Concepção do jogo (ação, aventura ou *quiz*) Material instrucional de história	Worksheet Prova final Projeto final
9ª aula (45 min) – Opcional				
5 min	Chamada	--	--	--
5 min	Entrar no site do Scratch (todos os grupos)	Apresentação passo-a-passo.	Lista de contas	
35 min	Continuação do projeto: programar função de atores/pano de fundo.	Laboratório de programação. Atividade em grupo. Auxilio respondendo dúvidas dos alunos.	Worksheet: Concepção do jogo (ação, aventura ou *quiz*) Material instrucional de história	Worksheet Prova final Projeto final
10ª aula (45 min)				
5 min	Chamada	--	--	--
5 min	Entrar no site do Scratch (todos os grupos)	Apresentação passo-a-passo.	Lista de contas	--
15 min	Continuação do projeto: fazer ajustes finais.	Laboratório de programação. Atividade em grupo. Auxilio respondendo dúvidas dos alunos.	Worksheet: Concepção do jogo (ação, aventura ou *quiz*) Material instrucional de história	Worksheet Prova final Projeto final
20 min	Correção da Tarefa de casa	Discussão com os alunos	Tarefa de casa Quadro	--
11ª aula (45 min)				
5 min	Chamada	--	--	--
40 min	Avaliação da aprendizagem	Aplicação de prova	Prova final	Prova final
12ª aula (45 min)				
5 min	Chamada	--	--	--
40 min	Apresentação e comentários sobre os jogos da turma.	Laboratório de programação. Apresentação do jogo por cada grupo.		Apresentação Projeto final
13ª aula (45 min) – Opcional				
5 min	Chamada	--	--	--
40 min	Apresentação e comentários sobre os jogos da turma.	Laboratório de programação. Apresentação do jogo por cada grupo.		Apresentação Projeto final
14ª aula (45 min)				
5 min	Chamada	--	--	--

20 min	Apresentação e comentários sobre os jogos da turma, discussão final.	Laboratório de programação. Apresentação do jogo por cada grupo.		Apresentação Projeto final
15 min	M1. Medição Pós-unidade	Questionário	Questionários: Pós-unidade Aluno Pós-unidade Pais Pós-unidade Instrutor	--

5.2 DESENVOLVIMENTO DE MATERIAIS

De acordo com o plano de ensino definido, foram desenvolvidos materiais que servem como recurso didático de apoio a unidade instrucional UNIfICA. Foram desenvolvidos os jogos com temas de história, prova, atividades, etc. (veja Tabela 29). Estes materiais estão apresentados na sua forma íntegra no endereço: <http://www.computacaonaescola.ufsc.br/?page_id=1476>.

Tabela 29 — Materiais instrucionais da unidade UNIfICA v2.0.

Material	Descrição	Imagem (extrato)
Jogos com tema de história	Jogos desenvolvidos em Scratch para serem apresentados aos alunos.	
Lista de grupos	Documento para ser preenchido com o nome dos integrantes de cada grupo.	

Plano de ensino	Documento que explicita o planejamento da unidade instrucional, apresenta os objetivos, os conteúdos e o sequenciamento da unidade para cada aula.	COMPUTAÇÃO NA ESCOLA — Unidade Escolar Interdisciplinar **Plano de Ensino** **Ensino Fundamental (8 a 12 anos) - Programação com Scratch** **Objetivo geral: Ensinar conceitos básicos da computação relacionados à prática da computação e programação, colaboração e pensamento computacional de forma interdisciplinar com o conteúdo da disciplina de história do ensino fundamental.** **Objetivos de aprendizagem** / **Relação com currículos de referência** Compreender o que é um algoritmo e usar os passos básicos de algoritmos para a resolução de problemas ao projetar soluções. Descrever e analisar uma sequência de instruções a ser seguidas. — Pensamento computacional (CSTA – Nível 2, 2011) Colaborar com colegas em grupo para criar, desenvolver, publicar e apresentar projetos, utilizando Scratch. — Colaboração (CSTA – Nível 2, 2011) Usar os passos básicos da engenharia de software (análise, projeto, implementação e teste) para criar o projeto de um jogo com tema de história utilizando conceitos básicos de programação (variáveis, operadores, estruturas condicionais e estruturas de repetição). — Práticas Computacionais e de Programação (CSTA – Nível 2, 2011) Demonstrar conhecimento sobre o modo de vida de diferentes grupos, em diversos tempos e espaços, em suas manifestações culturais e sociais, reconhecendo semelhanças e diferenças entre eles e seus conflitos. — História (PCN – MEC, 1998)
Prova	Documento para avaliar a aprendizagem dos alunos.	COMPUTAÇÃO NA ESCOLA — Unidade Escolar Interdisciplinar **Prova final** Data: / / Turma: Aluno(a): **1. (1 ponto) - Cite 3 exemplos onde a computação está presente.** **2. (1 ponto) - Cite 3 coisas que são possíveis fazer no Scratch.**
Questionários	Documentos para serem aplicados antes e depois da unidade.	COMPUTAÇÃO NA ESCOLA — Questionário pré-unidade (Aluno) — Unidade escolar interdisciplinar Escola: Turma: Data: / / Nome completo: Idade: Você é: ☐ Menina ☐ Menino Você já teve aulas para aprender a fazer programas de computador? ☐ Sim ☐ Não **Conte-nos o que você sabe:** Você sabe criar um programa de computador? ☐ Sim ☐ Não Você sabe ensinar alguém como criar um programa de computador? ☐ Sim ☐ Não
Roteiro tutorial instrutor	Documento que ensina ao professor como acessar o ambiente Scratch (on-line e off-line) e ensina a programar um jogo simples.	COMPUTAÇÃO NA ESCOLA — Guia do Instrutor – Jogo da Bruxa (simples) **FUNÇÃO**: Inicialmente a bruxa terá **3 pontos de vida**. quando clicar em / mude vida para 3 Programe os comandos ao lado. **TESTE**: Clique na bandeira verde. Funcionou? **9. Adicionar uma variável – PONTOS** Crie uma **variável** com o nome: "pontos" da mesma maneira que você criou a variável anterior.
Slides	Documento para apresentação de conceitos durante as aulas.	**Linguagem de programação** Linguagem usada para se conversar com computador e dizer o que ele deve fazer e quando deve fazer

Tarefas de casa	Documento para os alunos preencherem em casa.	COMPUTAÇÃO NA ESCOLA — Tarefa de casa — Unidade Escolar Interdisciplinar **Operadores** Na primeira aula vimos os conceitos de operadores, variáveis e estruturas condicionais **Operadores aritméticos**: servem para somar, subtrair, multiplicar e dividir ● + ● ● - ● ● * ● ● / ● **Operadores relacionais**: servem para comparar valores numéricos ■ < ■ ■ = ■ ■ > ■ **Operadores lógicos**: servem para operar valores booleanos (um valor booleano só pode ser verdadeiro ou falso) e não ou **Marque V para Verdadeiro e F para Falso para as afirmações a seguir:** Exemplo 1 — V — F — 3 < 10 Exemplo 2 — V — F — não 3 < 10
Rubrica	Documento para avaliar se os objetivos de aprendizagem foram atingidos.	COMPUTAÇÃO NA ESCOLA — Unidade Escolar Interdisciplinar **Rubrica – Atingimento dos objetivos da unidade** Este documento apresenta como avaliar os objetivos de aprendizagem da Unidade Instrucional Interdisciplinar. **Recursos de avaliação** • Worksheet (ação, aventura, quiz) (nota de 0 a 10) • Prova (nota de 0 a 11) • Projeto final (nota de 0 a 10) • Apresentação do projeto final (nota de 0 a 10) **Objetivos** Compreender o que é um algoritmo e usar os passos básicos de algoritmos para a resolução de problemas ao projetar soluções. Descrever e analisar uma sequência de instruções a ser seguidas. **MEDIDA**: Prova * 0,5 + Apresentação do projeto final * 0,5
Worksheet	Documento para auxiliar os alunos na documentação do jogo a ser desenvolvido por eles.	COMPUTAÇÃO NA ESCOLA — Unidade Escolar Interdisciplinar **Worksheet: Concepção de um jogo de ação** **Jogo onde o jogador utiliza reflexo rápido para realizar ações.** Data: / / Turma: Aluno(a): Aluno(a): Aluno(a): **Exemplo simples** **Tema**: Cultura de Florianópolis **Ideia**: Fazer um jogo sobre as bruxas em que os nativos de Florianópolis acreditavam **Nome**: Ilha da magia **Atores**: Bruxa e morcego. **Objetivo**: Controlar a bruxa voando mais alto ou mais baixo para fugir do morcego **Interatividade**: Teclado (seta para cima e seta para baixo)

6. AVALIAÇÃO

Este capítulo apresenta o planejamento da avaliação, definindo-se o que será avaliado e como será avaliado. Em seguida, é descrita a execução da unidade, bem como sua avaliação. Ao final, é apresentada a análise dos dados obtidos durante a execução.

6.1 PLANEJAMENTO DA AVALIAÇÃO

O objetivo geral da avaliação consiste em explorar e compreender aspectos relacionados a unidades instrucionais para o ensino de conceitos de computação no Ensino Fundamental de forma interdisciplinar no conteúdo da disciplina de História. A avaliação consiste em: analisar o **grau de aprendizagem**, a **facilidade de aprendizagem**, a **experiência de aprendizagem** da unidade e a **percepção em relação à computação** da unidade instrucional **UNIfICA** do ponto de vista do aluno da Escola Básica Municipal Prefeito Reinaldo Weingartner.

Para definir as medições em relação a estes objetivos, a abordagem GQM (BASILI, CALDEIRA e ROMBACH, 1994) é utilizada, na qual os quatro tópicos que se deseja analisar são decompostos em perguntas de análise e medidas, operacionalizadas por instrumentos de coleta de dados. A Tabela 30 apresenta o detalhamento das perguntas de análise, medidas e instrumentos de coleta de dados.

Tabela 30 — Detalhamento do planejamento da avaliação da UNIfICA v2.0.

Pergunta de análise	Medida	Instrumento de coleta de dados
Os objetivos de aprendizagem são atingidos usando a unidade instrucional?	Grau de aprendizagem referente à compreensão do que é um algoritmo, saber de descrever e analisar uma sequência de instruções a ser seguida.	Apresentação Projeto final
	Grau de aprendizagem referente à colaboração com colegas, saber ensinar o que aprendeu a outras pessoas e trabalhar em grupo.	Questionário pré-unidade aluno Questionário pós-unidade aluno Observações durante as aulas
	Grau de aprendizagem referente à capacidade de programar usando conceitos básicos de engenharia de software e de programação.	Apresentação Projeto final Worksheet Questionário pré-unidade aluno Questionário pós-unidade aluno
	Grau de aprendizagem referente ao modo de vida de diferente grupos, em diversos tempos e espaços.	Apresentação Projeto final
A unidade instrucional facilita a aprendizagem?	Grau de facilidade das aulas	Questionário pós-unidade aluno
	Grau de facilidade de fazer um programa de computador	Questionário pré-unidade aluno Questionário pós-unidade aluno

	Pontos positivos/negativos em relação à facilidade das aulas.	Questionário pós-unidade aluno Observações durante as aulas
A unidade instrucional promove uma experiência de aprendizagem agradável e divertida?	Grau de diversão das aulas	Questionário pós-unidade aluno
	Grau de qualidade das aulas	Questionário pós-unidade aluno
	Grau de imersão das aulas	Questionário pós-unidade aluno
	Grau de interação social	Questionário pós-unidade aluno Observações durante as aulas
	Pontos positivos/negativos em relação à experiência das aulas.	Questionário pós-unidade aluno
A unidade instrucional proporciona uma percepção positiva da computação?	Grau de diversão em fazer um programa de computador	Questionário pré-unidade aluno Questionário pós-unidade aluno
	Grau de satisfação em fazer um programa de computador	Questionário pós-unidade aluno
	Vontade de aprender computação na escola	Questionário pós-unidade aluno

Os questionários estão disponíveis na íntegra no endereço <http://www.computacaonaescola.ufsc.br/?page_id=1476>.

6.2 EXECUÇÃO DA AVALIAÇÃO

A unidade instrucional foi aplicada na turma 64, turma de **6º ano** do Ensino Fundamental, da **Escola Básica Municipal Prefeito Reinaldo Weingartner**, escolhida por critério de proximidade geográfica da presente autora, durante o primeiro semestre de 2016. A Escola Reinaldo Weingartner é uma escola pública localizada na cidade de Palhoça/SC. Os procedimentos de pesquisa foram aprovados pela Secretaria da Educação da Prefeitura de Palhoça, conforme descrito no Anexo A e pelo CEPSH - Comitê de Ética em Pesquisa com Seres Humanos da UFSC, conforme parecer número 1.021.541.

As aulas foram ministradas semanalmente pela autora do presente trabalho. Parte das aulas foram acompanhadas por um professor do INE/UFSC e pela professora de informática da escola. As aulas foram inseridas na carga horária da disciplina de educação física e da disciplina de ensino religioso presentes no currículo escolar. Isto foi necessário devido à dificuldade de encontrar horários em que havia: a disponibilidade das aulas dos professores da escola, e a disponibilidade do laboratório de informática da escola, e a disponibilidade da autora do presente trabalho.

A aplicação da unidade instrucional foi realizada seguindo o plano de ensino apresentado na seção 5.1 do capítulo 5, com exceção da prova final. Seguindo o conteúdo abordado previamente nas aulas de História, o tema escolhido para os jogos dos alunos foi a Pré-História. Participaram da aplicação um total de **31 alunos** (veja Figura 14).

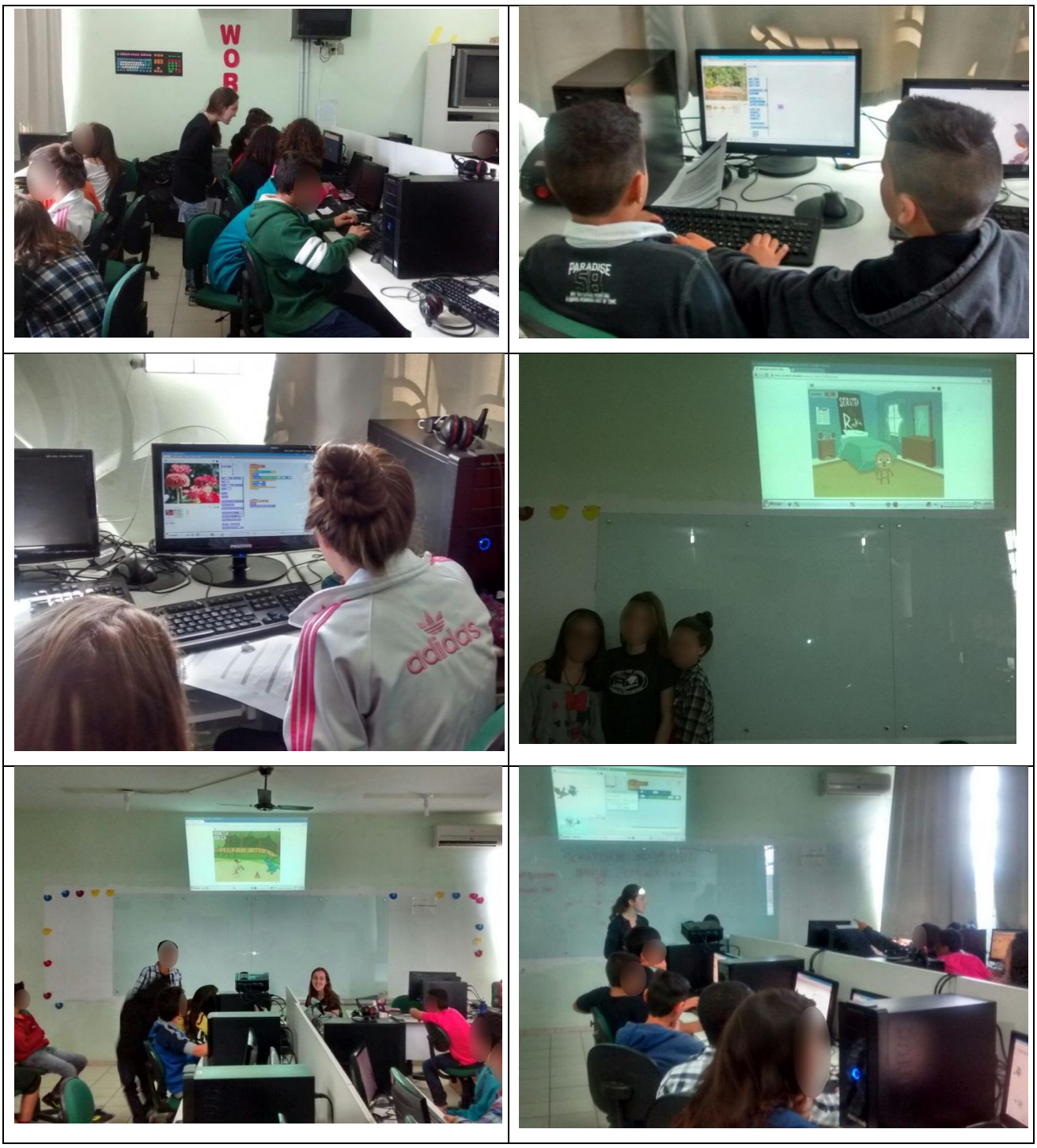


Figura 14 — Alunos do sexto ano do colégio Reinaldo Weingartner durante a aplicação da UNIfICA v2.0.

Os alunos desenvolveram, em grupos de 1 a 3 integrantes, um total de **15 jogos**. Os resultados finais desenvolvidos pelos alunos durante o projeto estão compartilhados no ambiente Scratch (veja Figura 15) e podem ser encontrados no Estúdio: <https://scratch.mit.edu/studios/2201691/>.

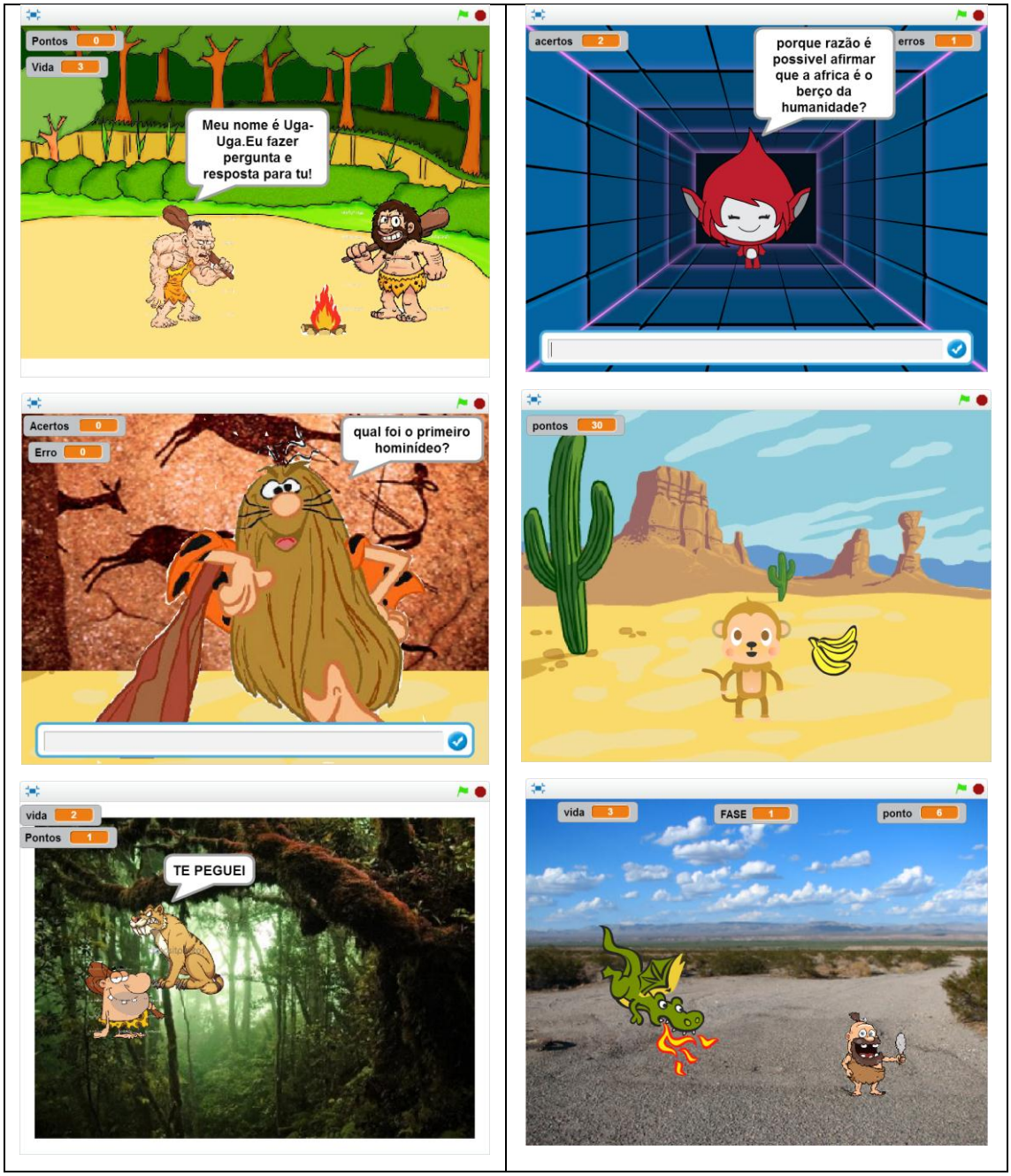


Figura 15 — Extrato de jogos desenvolvidos pelos alunos durante a UNIfICA v2.0.

Nas aulas foram utilizadas: a versão 4.45 do Scratch *on-line* para uso em navegadores *web* e a versão 2.0 do Scratch *off-line* para Linux. Todos os computadores da sala de informática da escola tem como sistema operacional **Linux Educacional**. A maioria dos computadores tem instalada a versão 3, os demais a versão 4. A versão 3 do Linux Educacional não suporta a praticidade de edição e salvamento de projetos do Scratch direto no navegador. Isto ocorre pois os navegadores desta versão são muito antigos e não possuem

compatibilidade com aplicações mais avançadas como o Scratch. Por este motivo, alguns grupos utilizaram a versão *off-line* (nos computadores com a versão 3) e outros utilizaram a versão *on-line* (nos computadores com a versão 4).

6.3 ANÁLISE DOS DADOS COLETADOS

Os dados coletados via rubricas, questionários e observações durante a aplicação da unidade foram analisados por meio de análises qualitativas e quantitativas utilizando estatística descritiva. Os alunos puderam criar, de acordo com o plano de ensino, jogos do gênero de ação, aventura e quiz. Alguns alunos criaram jogos que denotam mais de um gênero, como por exemplo, um jogo que mistura elementos de aventura com quiz. Contudo, para fins de análise, os jogos foram classificados somente em um gênero, isto é, foram classificados pelo gênero que denota a maior quantidade de elementos presentes no jogo. Foram criados no total 15 jogos (veja Figura 16).

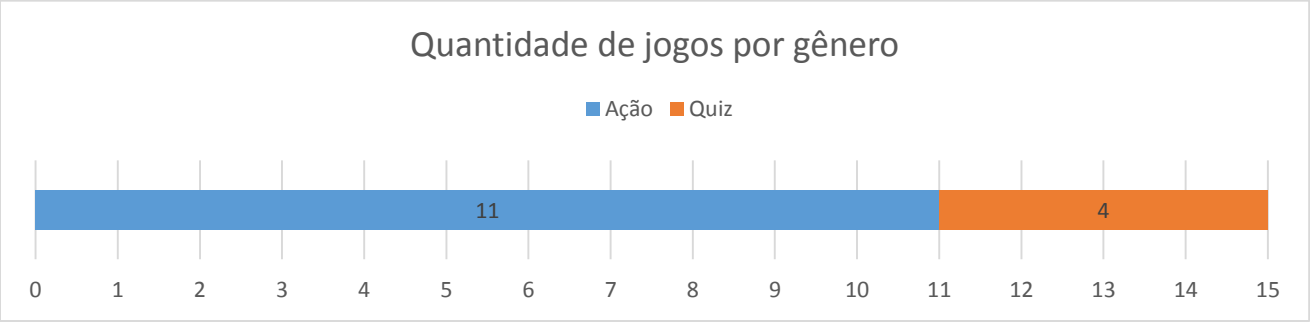


Figura 16 — Gênero dos jogos criados pelos alunos durante a aplicação da UNIfICA v2.0.

6.3.1 Análise do atingimento dos objetivos de aprendizagem da unidade

Durante a apresentação dos jogos, foi possível verificar que a maioria dos alunos sabia descrever e analisar uma sequência de instruções a ser seguida. No entanto, poucos alunos demonstraram conhecimento referente à compreensão do que é um algoritmo.

Notou-se também durante as aulas uma dicotomia entre alunos que queriam trabalhar em grupo e outros que não queriam. Inclusive 3 grupos da turma foram compostos por apenas um integrante. Apesar disto, os alunos que escolheram trabalhar sozinhos, em alguns momentos, ajudavam e/ou recebiam ajuda dos colegas próximos.

De acordo com os questionários pré/pós aplicação, na pergunta sobre saber ensinar o que aprendeu, verificou-se que após a aplicação houve um aumento dos alunos que

responderam que saberiam. No questionário pré nenhum aluno respondeu que sabia (veja Figura 17).

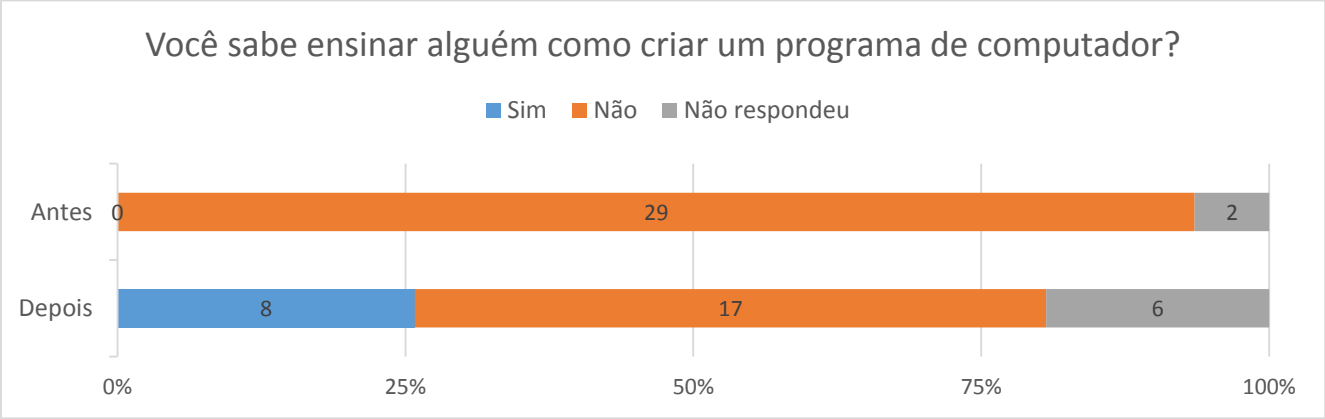


Figura 17 — Capacidade de ensinar alguém a criar um programa de computador antes e depois da UNIfICA v2.0.

A partir do projeto e da sua relação com o *worksheet* - documento onde deveria se planejar o jogo – verificou-se que pouco mais da metade da turma fez o jogo de acordo com o planejado. Durante as aulas foi possível perceber que todos os grupos executaram testes sobre programações feitas. De acordo com as respostas dos questionários, o número de alunos que sabem criar um programa de computador dobrou (veja Figura 18).

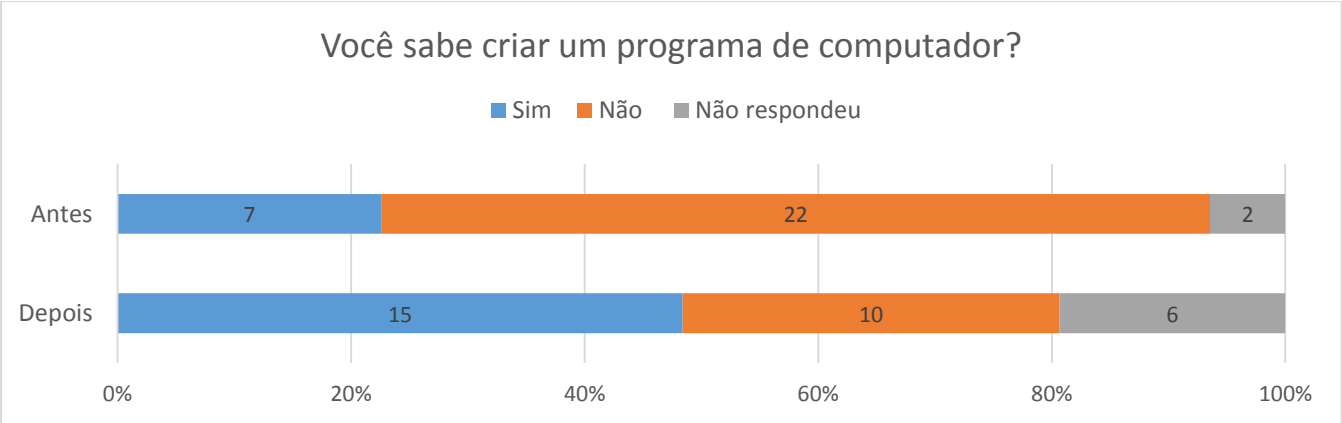


Figura 18 — Capacidade de criar um programa de computador antes e depois da UNIfICA v2.0.

Os jogos foram analisados também por meio do Dr.Scratch (veja Figura 19). Pôde-se verificar na parte de **lógica** que a maioria dos grupos recebeu pontuação excelente (3 pontos). Isto demonstra que os grupos utilizaram comandos da lógica booleana e condicional (se, então). Somente 1 grupo não utilizou comando condicional, este grupo em específico era formado por apenas um aluno que não conseguiu concluir o jogo durante as aulas.

A parte de **paralelismo** recebeu na maioria pontuação baixa (1 ponto). Isto evidencia

que muitos alunos não utilizaram mais de uma vez o mesmo comando de manipulação de eventos, como por exemplo, utilizar o comando “quando uma tecla for pressionada” para disparar duas ações diferentes. Contudo, verificou-se que os alunos utilizaram estes comandos com intuito de disparar somente uma ação de forma coerente. Cabe ressaltar que esta pontuação também demonstra que a maioria dos grupos programou mais de um *script* dentro do jogo.

Na **interatividade com o usuário** todos os jogos receberam a mesma classificação (2 pontos). Isto mostra que todos os grupos utilizaram comandos ou para fazer perguntas ao usuário (em jogos do tipo quiz) ou para disparar ações a partir de eventos (em jogos do tipo ação), como por exemplo, quando uma tecla é pressionada o ator movimenta-se para determinada direção. Nenhum grupo utilizou comandos de sensores de câmera ou microfone (a escola não provia estes dispositivos).

Na **representação de dados** a maioria recebeu classificação boa (2 pontos). Isto ocorreu pois quase todos os grupos criaram e realizaram operações com variáveis. Grande parte também fez operações sobre os atributos de atores, como por exemplo, modificação de suas coordenadas. Nenhum grupo fez operações com listas (o que permitiria ter uma pontuação igual a três), no entanto, isto não foi ensinado aos alunos.

Na parte de **controle de fluxo** a maioria recebeu classificação boa (2 pontos) ou acima. Isto ocorreu pois os grupos utilizaram-se de comandos de laço, como por exemplo, repita uma quantidade de vezes um bloco de comandos ou repita sempre. Uma minoria recebeu pontuação baixa (1 ponto) pois não utilizaram os comandos citados. Estes grupos programaram somente sequências de blocos, o que não é ruim em si, pois verificou-se que não havia a necessidade de utilizar estes comandos especificamente em alguns destes jogos.

Na **sincronização**, aproximadamente 70% dos jogos receberam pontuação excelente (3 pontos) ou boa (2 pontos). Isto demonstra que estes grupos utilizaram comandos de envio/recebimento de mensagens ou comandos de sincronização utilizando-se da satisfatibilidade de condições booleanas ou da manipulação eventos, como por exemplo, realizar uma ação quando ocorrer a mudança do pano de fundo. Uma parcela dos grupos recebeu classificação baixa (1 ponto) ou muito baixa (0 pontos), o que demonstra que os grupos com classificação baixa utilizaram comandos de sincronização mais simples, como por exemplo, espere uma quantidade de tempo até realizar uma ação. Os grupos com classificação muito baixa não utilizaram nenhum comando deste tipo.

E por fim, na parte de **abstração** a maioria dos grupos recebeu pontuação baixa (1 ponto). Isto ocorreu porquê nenhum grupo utilizou o comando de definição de funções e nem criou clones a partir dos atores presentes no projeto, estes dois tópicos também não foram ensinados aos alunos. Um grupo recebeu classificação muito baixa (0 pontos) porque programou todo o jogo em um script somente.

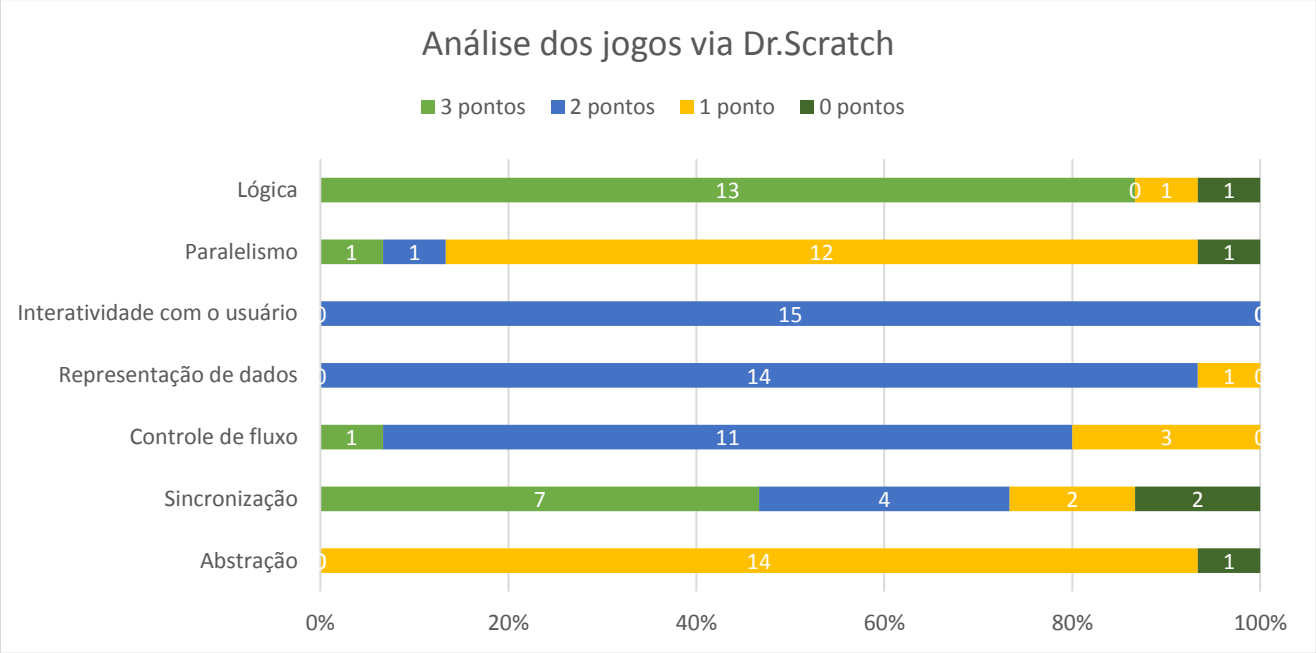


Figura 19 — Análise dos jogos via Dr.Scratch.

No que se refere ao objetivo de História, os jogos foram classificados em relação ao tema proposto “Pré-História” por três conceitos: atende totalmente, atende parcialmente, e não atende. Notou-se um desvio do tema em alguns jogos do gênero Ação, enquanto que todos os jogos do gênero Quiz atendem totalmente ao tema proposto (veja Figura 20). Um dos motivos que pode ter ocasionado este fato, seria a falta de envolvimento do professor de História durante as aulas.

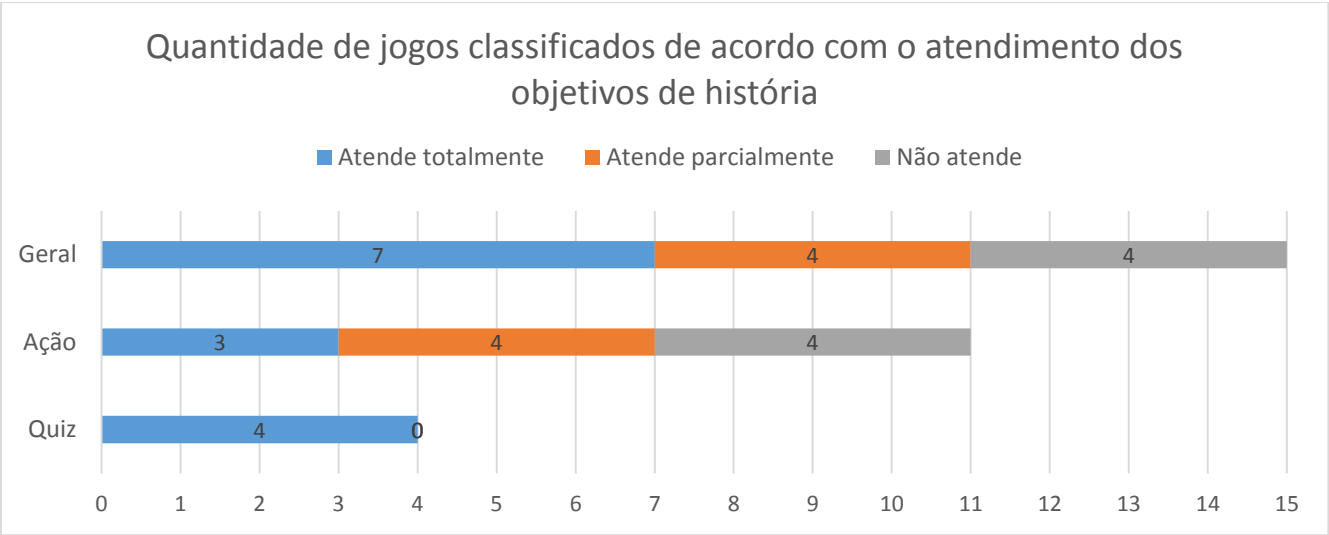


Figura 20 — Atendimento dos objetivos de história nos jogos desenvolvidos durante a UNIfICA v2.0.

6.3.2 Análise do grau de facilidade de aprendizagem da unidade

A partir das respostas dos alunos nos questionários pré/pós unidade ficou evidente que apesar parte os alunos não acharem fácil/muito fácil fazer um programa de computador, metade dos alunos achou as aulas em si fáceis/muito fáceis (veja Figura 21 e 22).

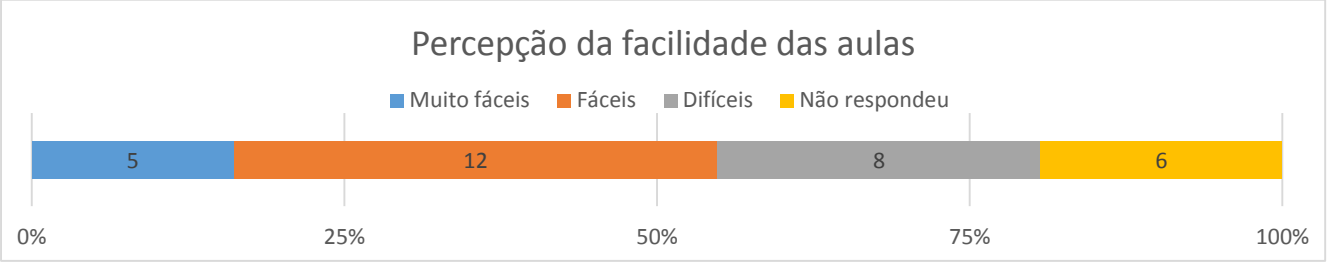


Figura 21 — Percepção da facilidade das aulas pelos alunos durante a UNIfICA v2.0.

Também verificou-se que anteriormente a aplicação parte alunos não conheciam o grau de facilidade em fazer programas de computador e alguns também achavam que deveria ser difícil. Após a aplicação o número de alunos que considera fácil fazer programas dobrou, assim como os que consideram difícil/muito difícil (veja Figura 22).

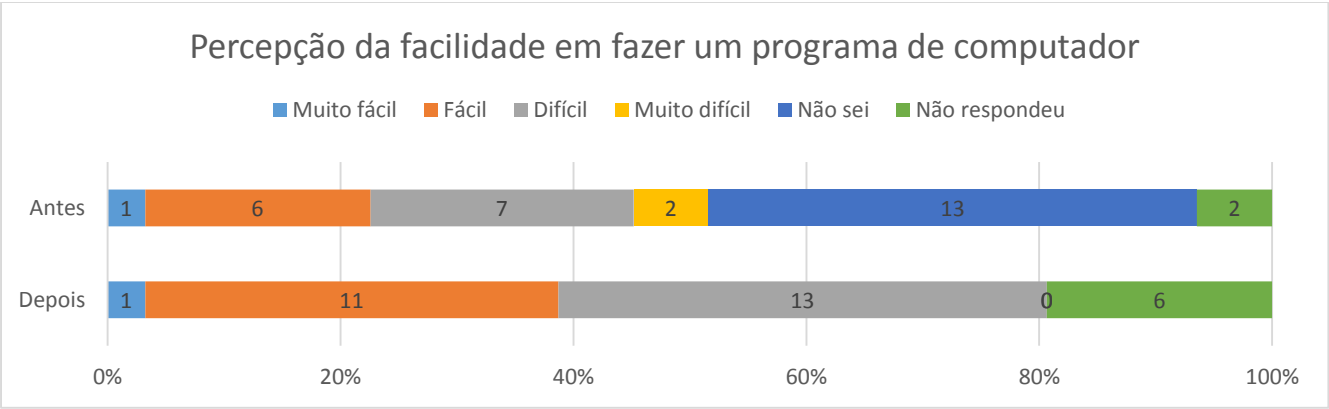


Figura 22 — Percepção dos alunos em relação à facilidade de fazer programas de computador antes e depois da UNIfICA v2.0.

Os alunos também mencionaram alguns pontos positivos/negativos em relação à facilidade das aulas. A maior dificuldade foi em relação à programação, os alunos acharam os comandos de “se, então”, “se, então; senão” e comandos da lógica booleana particularmente difíceis de se entender. As facilidades estão relacionadas aos comandos do Scratch serem bastante intuitivos.

6.3.3 Análise da experiência de aprendizagem da unidade

Durante as aulas, pode-se perceber que mais da metade dos alunos demonstravam estar tendo uma experiência positiva (veja Figura 23). No entanto, alguns alunos demonstraram estar tendo uma experiência chata, um dos motivos foi devido à pequena quantidade de pessoas disponíveis para atendimento aos grupos. Também foi observada frustração nos alunos por causa de problemas para progredir devido ao software desatualizado. Além disso, poucos alunos demonstraram em algumas aulas profunda falta de motivação, apesar de tentativas de incentivo por parte da presente autora e até mesmo por parte dos colegas de classe. Notou-se que estes alunos tiveram uma mudança de atitude após o início da apresentação dos jogos pelos colegas.

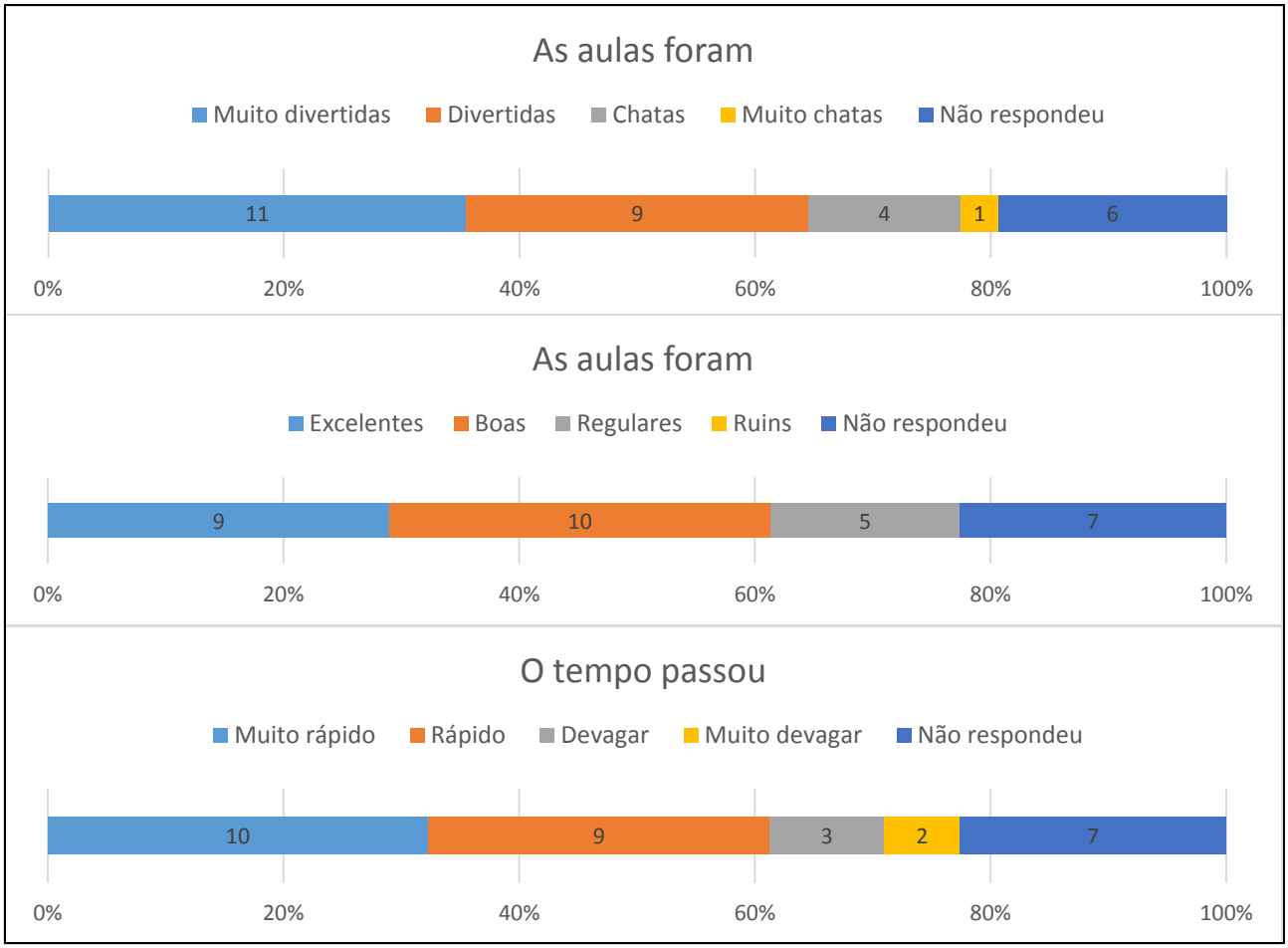


Figura 23 — Experiência das aulas durante a UNIfICA v2.0.

Cabe ressaltar que os alunos gostaram de assistir às apresentações dos colegas e criar um jogo próprio (veja Tabela 31). Entretanto, aproximadamente metade dos alunos não gostou de apresentar ou mostrar o próprio jogo a outras pessoas (veja Figura 24).

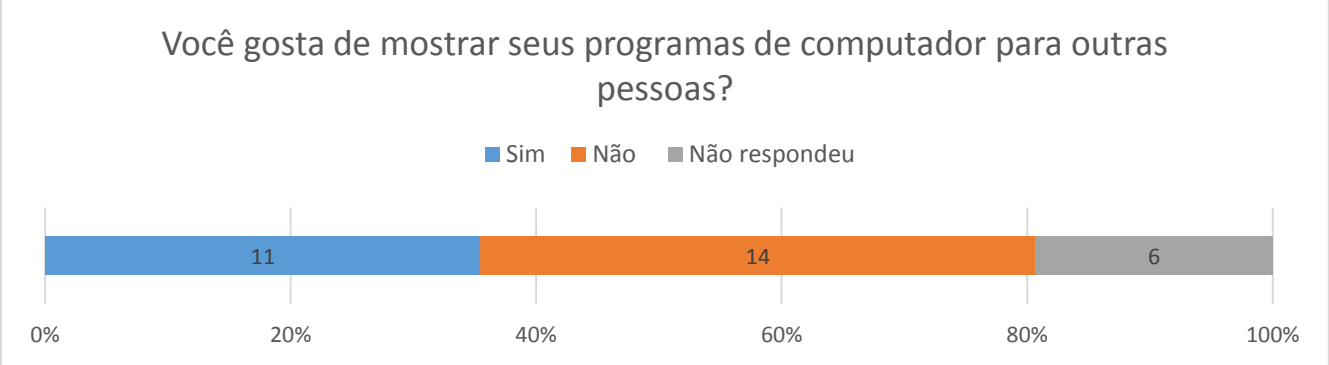


Figura 24 — Satisfação em mostrar o jogo a outras pessoas durante a UNIfICA v2.0.

A Tabela 31 apresenta alguns pontos positivos/negativos levantados pelos alunos em relação à experiência das aulas de forma geral.

Tabela 31 — Pontos levantados pelos alunos durante a UNIfICA v2.0.

Pontos positivos	Pontos negativos
- Oportunidade de criar um jogo próprio - Apresentar para todos os colegas a nossa ideia e o jogo. - Criar os personagens de forma livre do jogo. - Utilizar o computador. - Assistir a apresentação dos outros jogos dos meus colegas. - Aprender como movimentar personagens no jogo.	- O tempo, pois não consegui terminar meu jogo. - Mostrar o jogo para os colegas. Ter que apresentar é muito vergonhoso. - Não obtive ajuda sempre que necessário. - Alguns colegas não prestavam atenção. - Bagunça na sala. - Programar, achei muito difícil.

6.3.4 Análise da percepção do aluno em relação à computação (antes e depois)

Não se notou uma alteração significante em relação ao grau de diversão em fazer programas de computador antes e depois da aplicação (veja Figura 25). Contudo, pode-se afirmar que nas respostas de antes da aplicação, a maioria dos alunos respondeu sobre o que achava, isto é, respondeu por meio de uma especulação, isto porquê muitos alunos nunca haviam experienciado na prática a programação.

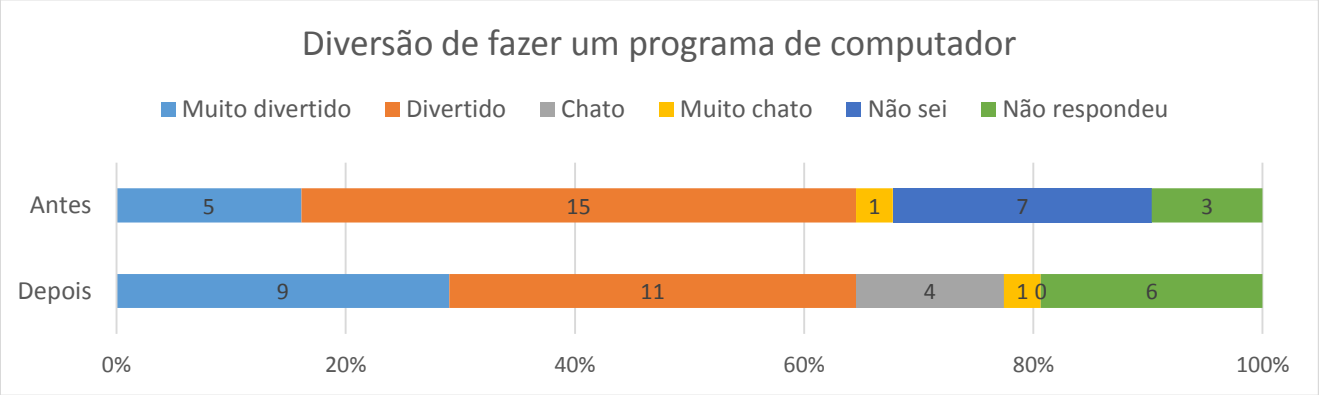


Figura 25 — Percepção dos alunos sobre diversão em criar um programa de computador antes e depois da UNIfICA v2.0.

A motivação de aprender computação na escola e a satisfação em fazer programas de computador tiveram resultados semelhantes. Pode-se afirmar, a partir das respostas dos questionários, que os alunos que afirmaram gostar de programar, gostariam também de continuar aprendendo computação na escola. Inclusive, na última aula do projeto, alguns alunos comentaram com a autora do presente trabalho que gostariam que as aulas se estendessem até o fim do ano.

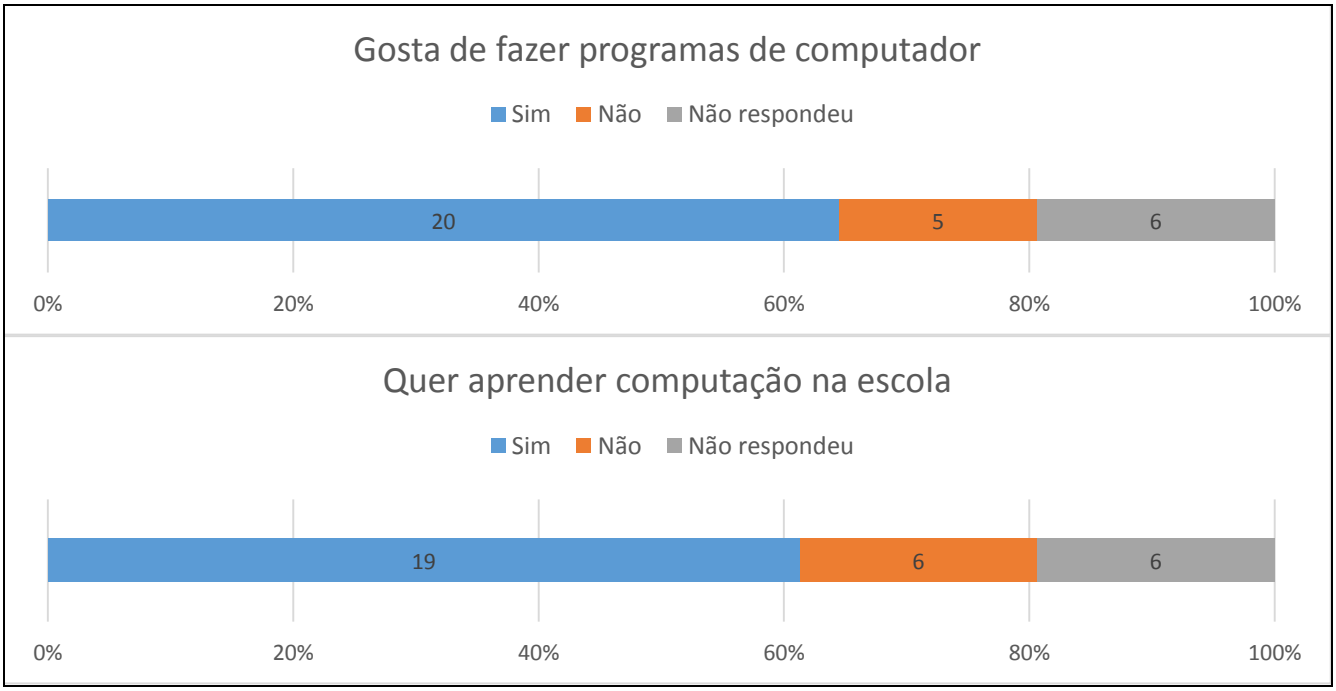


Figura 26 — Motivação em relação à computação depois da UNIfICA v2.0.

6.4 DISCUSSÃO

A avaliação da unidade pelos alunos apresentou resultados interessantes, apesar das dificuldades encontradas especificamente nesta aplicação. Por exemplo, mais de 60% dos alunos consideraram as aulas boas ou excelentes. Cabe ressaltar que a aplicação foi realizada numa situação incongruente, a citar como exemplo a versão do sistema operacional da escola ser muito desatualizado – Linux Educacional versão 3, é possível visualizar na Figura 7 que esta versão representa uma porcentagem pequena em relação a todas as escolas públicas brasileiras. Isto impossibilitou que todos os alunos trabalhassem da mesma maneira, dificultando bastante o trabalho da aplicação em si.

Como resultado desta aplicação, foram percebidos alguns pontos que podem influenciar diretamente no sucesso/fracasso da aplicação da unidade instrucional. Desta forma, são criadas recomendações para que a unidade possa atingir da melhor forma seu objetivo (veja Tabela 32).

Tabela 32 — Recomendações para a aplicação da UNIfICA v2.0.

Recomendações
Para o professor da disciplina - Conhecer conceitos básicos do Scratch; - Conhecer todos os materiais da unidade UNIfICA; - Trabalhar em conjunto com o professor/instrutor de informática da escola; - Avaliar o comportamento dos alunos de forma que estejam ciente disto.
Para o professor de informática/laboratório - Conhecer conceitos básicos do Scratch; - Conhecer todos os materiais da unidade UNIfICA; - Auxiliar o professor da disciplina a aplicar a unidade, como por exemplo, tirando dúvidas dos alunos em relação ao Scratch. - Conhecimento sobre como instalar o Scratch *off-line*, se necessário.
Para o laboratório - O sistema operacional das máquinas deve ser Linux educacional versão 4 ou superior, Windows 7 ou superior, Mac 10 ou superior. - Conexão estável à internet.
Para melhor aproveitamento das aulas - A cada 10 alunos, estar disponível um professor/instrutor para auxílio de dúvidas nas aulas de laboratório. No caso de 30 alunos sugere-se que haja pelo menos 3 professores auxiliando os alunos, podendo ser professores da disciplina ou do laboratório/informática; - Grupos de, no máximo, 3 alunos para compartilhar um computador.

7. CONCLUSÃO

O objetivo do presente trabalho foi o desenvolvimento de uma unidade instrucional interdisciplinar para ensinar computação no ensino fundamental com História. Nesse contexto, foi feita uma revisão sobre o estado da arte de unidades instrucionais interdisciplinares para se ensinar computação para crianças e jovens. Após a busca, foi possível perceber uma carência de unidades desse tipo já que foram encontrados poucos resultados. Também foi realizada uma revisão teórica, na qual foram apresentados tópicos para serem abordados durante a criação, o desenvolvimento, a aplicação e a avaliação da unidade em si. Após a revisão teórica, deu-se início ao desenvolvimento da unidade instrucional.

Foi criada uma unidade e aplicada a partir de uma pesquisa exploratória. Verificou-se que poderia se realizar melhoramentos em vários tópicos. A unidade foi melhorada e então aplicada novamente. Os resultados indicam que a mesma é adequada tanto em relação aos materiais de apoio desenvolvidos, quanto em relação a diversos aspectos, incluindo motivação, facilidade e atendimento dos objetivos de aprendizagem. Como resultado do presente trabalho, é disponibilizada a unidade instrucional para ser aplicada por qualquer professor que tenha interesse, levando-se em conta os pré-requisitos e as recomendações para uma aplicação bem sucedida.

Como conteúdo para trabalhos futuros, é possível criar oficinas para professores do Ensino Fundamental para se ensinar conceitos de computação e de programação, com o Scratch, a partir do roteiro e de outros materiais desenvolvidos neste trabalho. Desta forma a aplicação da unidade pode ser realizada levando-se em consideração as recomendações e, assim, pode atingir de maneira mais eficiente o objetivo proposto. Também podem ser feitas adaptações deste material para outras disciplinas do Ensino Fundamental, como, por exemplo, Geografia e Matemática.

REFERÊNCIAS

ALVES, N. D. C. et al. **Ensino de Computação de Forma Interdisciplinar em Disciplinas de História no Ensino Fundamental – Um Estudo de Caso**. Artigo submetido para RBIE – Revista Brasileira de Informática na Educação. Florianópolis. 2016.

ANGELO, A. G. S.; HENNO, J. H.; CAMPOS, P. E. F. **Projeto Fab Social:** Introdução à programação em um curso livre aplicado na periferia de Guarulhos. Alice Brasil: anais 2013. São Paulo: Páginas & Letras Editora e Gráfica. 2013. p. 27-32.

BASILI, V. R.; CALDEIRA, G.; ROMBACH, H. D. Goal Question Metric Paradigm. **Encyclopedia of Software Engineering**, John Wiley & Sons, v. 2, 1994.

BASQUE, J. **Ingénierie Pédagogique Et Technologies Éducatives**. Québec: Télé-université, 2010.

BLIKSTEIN, P. **O Pensamento Computacional e a Reinvenção do Computador na Educação**, 2008. Disponivel em: <http://bit.ly/1IXIbNn>. Acesso em: Julho 2015.

BLOOM, B. S. **Taxonomy of Educational Objectives, The Classification of Educational Goals**. New York: David McKay, 1956.

BNCC. **Base Nacional Comum Curricular**. MEC. Brasil. 2016.

BRANCH, R. M. **Instructional Design:** The ADDIE Approach. New York: Springer, 2009.

BRASIL. **Lei de Diretrizes e Bases da Educação Nacional - Lei Nº 9.394**. BRASILIA. 1996.

CLARK, D. R. **ADDIE Model**, 2009. Disponivel em: <http://www.nwlink.com/~donclark/hrd/bloom.html>. Acesso em: Agosto 2015.

CNE. GQS/INCoD/INE/UFSC. **Iniciativa Computação na Escola**, 2013. Disponivel em: <http://www.computacaonaescola.ufsc.br>. Acesso em: Abril 2015.

CODE. **Code.org**, 2013. Disponivel em: <https://code.org/>. Acesso em: Julho 2015.

CSTA. ACM. **CSTA K –12 Computer Science Standards**, 2011. Disponivel em: <http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf>. Acesso em: Julho 2015.

FILATRO, A. **Design instrucional contextualizado:** educação e tecnologia. São Paulo: SENAC, 2004.

FILHO, R. L. L. S. et al. **Cadernos de Pesquisa**, v. 37, n. 132, set./dez. 2007. Disponivel em: <http://www.scielo.br/pdf/cp/v37n132/a0737132.pdf>. Acesso em: Julho 2015.

FINE, L. The Role of the University in Computers, Data Processing, and Related Fields. **Communications of the ACM**, v. 2, n. 9, p. 7–14, 1959.

FRANKLIN, D. et al. **Animal tlatoque, Attracting middle school students to computing through culturally-relevant themes**. Proceedings of the 42nd ACM Technical Symposium on

Computer Science Education - SIGCSE '11. NY, EUA: ACM. 2011. p. 453-458.

GOLDSCHMIDT, D. et al. An interdisciplinary approach to injecting computer science into the K-12 classroom. **Journal of Computing Sciences in Colleges**, v. 26, n. 6, p. 78-85, Junho 2011.

GUZDIAL, M. **Programming environments for novices**. Computer Science Education Research. Lisse, The Netherlands: Taylor & Francis. 2004. p. 127-154.

HAIDT, R. C. **Curso de Didática Geral**. São Paulo: Editora Ática, 2001.

HERZ, J. C. **Joystick nation, how videogames ate our quarters, won our hearts, and rewired our minds**. Boston, MA: Little, Brown and Company, 1997.

HUIZINGA, J. **Homo Ludens, o jogo como elemento da cultura**. 4. ed. São Paulo: Perspectiva, 1993. Tradução João Paulo Monteiro.

IBGE. **Pesquisa Nacional por Amostra de Domicílios (PNAD)**. Brasil. 2014.

INEP. **Censo escolar da educação básica**. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Brasil. 2010 a 2014.

INEP. **Sinopses Estatísticas da Educação Superior – Graduação**. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Brasília. 2010 a 2014.

INEP. **Indicadores Educacionais 2014**. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Brasília. 2015.

JIMENEZ, Y.; GARDNER-MCCUNE, C. **Using App inventor & history as a gateway to engage African American students in computer science**. Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT). Charlotte, NC, EUA: IEEE. 2015. p. 1-2.

KAFAI, Y. **The educational potential of electronic games, From games-to-teach to games-to-learn**. Conference on playing by the rules: the cultural policy challenges of video games. Chicago/EUA. 2001.

KITCHENHAM, B. **Procedures for Performing Systematic Reviews**. Joint Technical Report, TR/SE-0401 and NICTA 0400011T.1: Keele University. Reino Unido, 2004.

KONZACK, L. Video Game Genres. In: KHOSROW-POUR, M. **Encyclopedia of Information Science and Technology**. 3. ed. Hershey, PA: IGI Global, 2014.

LIN, H. S. IT Fluency: What Is It, and Why Do We Need It? In: JOSSEY-BASS **Technology Everywhere A Campus Agenda for Educating and Managing Workers in the Digital Age**. EUA: EDUCAUSE, v. 6, 2002.

MEC. **Diretrizes Curriculares dos cursos de Bacharelado em Ciência da Computação**. Brasília: Diário Oficial da União, 2003.

MEC; PCN. **Parâmetros Curriculares Nacionais: Introdução aos Parâmetros Curriculares**

Nacionais, v. 1, 1998.

MEC; PCN 5.1. **Parâmetros Curriculares Nacionais: História e Geografia (1ª a 4ª série)**, v. 5.1, 1998.

MEC; PCN 5.2. **Parâmetros Curriculares Nacionais: História e Geografia (5ª a 8ª série)**, v. 5.2, 1998.

MERRILL, M. D. et al. Reclaiming instructional design. **Educational Technology**, v. 36, n. 5, p. 5-7, 1996. Disponivel em: <http://mdavidmerrill.com/Papers/Reclaiming.PDF>.

MICHAELIS. **Michaelis Moderno dicionário da lingua portuguesa**. 1. ed. São Paulo: Melhoramentos, 2000.

ORLICH, D. et al. **Teaching strategies, A guide to effective instruction**. 7ª. ed. EUA: Houghton Mifflin, 2004.

PROGRAMAMOS; DR.SCRATCH. **Dr.Scratch - Analise seus projetos Scratch aqui**. Disponivel em: <http://drscratch.programamos.es/>. Acesso em: Junho 2016.

PROINFO; MEC. **Coleta de dados do projeto PROINFO/MEC de inclusão digital nas escolas públicas brasileiras**. Brasil. 2016.

QUINN, C. N. **Engaging Learning:** Designing e-Learning Simulation Games. EUA: Pfeiffer, 2005.

RESNICK, M. **Sowing the Seeds for a more creative society**. Learning and Leading with Technology. US & Canada: International Society for Technology in Education (ISTE). 2007. p. 18-22.

RUSK, N.; RESNICK, M.; MALONEY, J. **Scratch and 21st Century Skills**. MIT Media Lab. US: Lifelong Kindergarten Group. 2006.

SALAZAR, L. H. A. **Desenvolvimento de uma Ferramenta para Auxiliar a Execução de Revisões Sistemáticas da Literatura**. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade Federal de Santa Catarina. 2015.

SCRATCH BRASIL. **Scratch Brasil**, 2014. Disponivel em: <http://scratchbrasil.net.br/>. Acesso em: Junho 2016.

SCRATCH; MIT. **Scratch**, 2013. Disponivel em: <http://scratch.mit.edu>. Acesso em: Junho 2016.

SCRATCHED; RANDALL. **Relato "Embedding Scratch in US History/Geography"**, 2009. Disponivel em: <http://scratched.gse.harvard.edu/resources/embedding-scratch-us-historygeography>. Acesso em: Outubro 2015.

SCRATCHED; WILSON. **Relato "Introduction to Programming"**, 2011. Disponivel em: <http://scratched.gse.harvard.edu/resources/introduction-programming>. Acesso em: Outubro 2015.

SILVA, M. A. D.; FONSECA, S. G. Ensino de História hoje: errâncias, conquistas e perdas. **Rev. Bras. Hist.**, v. 30, p. 13-33, 2010.

SIMPSON, E. J. **The classification of educational objectives, psychomotor domain**. Washington: Gryphon House, 1972.

SOFTEX. **Relatório anual 2013**. Associação para promoção da Excelência do Software Brasileiro. Campinas/SP, p. 79. 2013.

SPRINTHALL, N. A.; SPRINTHALL, R. **Psicologia Educacional**. Lisboa: Mc Graw-Hill, 1993.

SPRINTHALL, R. **Psicologia educacional**. Mcgraw hill. [S.l.]. 1993.

VENTURA, L. **O processo de ensino aprendizagem**. 3. ed. Florianópolis: Programa de Desenvolvimento de Educadores, Serviço Nacional de Aprendizagem Comercial, 2005.

WANGENHEIM, A. V.; WANGENHEIM, C. G. V. **Ensinando Computação com Jogos**. Florianópolis/SC: Bookess Editora, 2012.

WANGENHEIM, C. G.; NUNES, V. R.; DOS SANTOS, G. D. Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso. **Revista Brasileira de Informática na Educação**, v. 22, n. 3, 2014.

WINN, W. The assumptions of constructivism and instructional design. **Educational Technology**, NJ, EUA, v. 31, n. 9, p. 38-40, Setembro 1991. Disponivel em: <http://dl.acm.org/citation.cfm?id=133166>.

YIN, R. K. **Estudo de caso:** planejamento e métodos. Tradução de Daniel Grassi. 2ª. ed. Porto Alegre: Bookman, 2001.

ANEXO A – APROVAÇÃO PREFEITURA DE PALHOÇA

ESTADO DE SANTA CATARINA
PREFEITURA MUNICIPAL DE PALHOÇA
SECRETARIA DE EDUCAÇÃO
AV. HILZA TEREZINHA PAGANI, 280 – PARQUE RESIDENCIAL PAGANI – PALHOÇA/SC – CEP: 88132-271
FONE/FAX: (48) 3279-1745 – CNPJ: 82.892.316/0001-08 – VISITE NOSSO SITE: www.palhoca.sc.gov.br

Palhoça, 15 de abril de 2016.

AUTORIZAÇÃO DE VISITA ÀS ESCOLAS

Sr.(a) Diretor(a)

Autorizamos através de Nathália da Cruz Alves, estudante do curso de Ciência da Computação da Universidade Federal de Santa Catarina a realizar uma pesquisa referente a " Computação na Escola" para alunos de 5° ao 9° Ano do Ensino Fundamental. A mesma se responsabiliza por ligar com antecedência para as escola marcando data e horário para ser atendida.

Salientamos que fica exclusivamente a critério do Diretor(a) aceitar ou não a realização do projeto em sua escola.

Atenciosamente;

Marcos Moser
Diretor de Ensino

Marcos Moser
Diretor de Ensino
Secretaria de Educação de Palhoça
Matrícula 802085-2

ANEXO B – ARTIGO

Ensinando Computação de Forma Interdisciplinar em Disciplinas de História*

Nathalia da Cruz Alves[1], Christiane G. von Wangenheim[1], Jean C. R. Hauck[1], Adriano F. Borgatto[1], Pedro Eurico Rodrigues[2]
[1]Intituto Nacional para Convergência Digital (INCoD)/ Departamento de Informática e Estatística (INE)/ Universidade Federal de Santa Catarina (UFSC)
[2]Escola Autonomia

nathaliaalves@grad.ufsc.br, c.wangenheim@ufsc.br, jean.hauck@ufsc.br, adriano.borgatto@ufsc.br, pedro.eurico.rodrigues@gmail.com

Abstract. *Computing is becoming increasingly ubiquitous in human life. Nevertheless, in Brazil, computing concepts are only taught in specific courses of higher education. In this context, this project consists of the systematic development of an interdisciplinary instructional unit aligned to the curriculum guidelines of computer science and History in Elementary/Middle School. The instructional unit focuses on teaching the basics of computing, such as the design of logical and computational thinking and programming. The results show that the unit is suitable, as the students achieve most of the learning objectives through the instructional unit, and motivational, as it raises interest in students about computing.*

Resumo. *A computação está se tornando cada vez mais onipresente na vida do ser humano. Apesar disso, no Brasil, seus fundamentos só são ensinados em cursos específicos do ensino superior. Nesse contexto, este trabalho visa o desenvolvimento sistemático de uma unidade instrucional interdisciplinar alinhada às diretrizes de currículo para computação e para o Ensino de História no Ensino Fundamental 2. A unidade instrucional tem como foco o ensino de conceitos básicos de computação, tais como a concepção do pensamento lógico e computacional e programação. Os resultados demonstram que a unidade é adequada pois os objetivos de aprendizagem são atingidos e desperta interesse nos alunos acerca da computação.*

1. Introdução

A inclusão do ensino de computação no Ensino Fundamental é uma tendência mundial, sendo que já existem diversas iniciativas nesse sentido, como, por exemplo, Code.org ou codeclub. Porém, no Brasil, atualmente o ensino de computação ainda não é incluído na grade curricular [PCN 1998], e a sua inclusão tem sido dificultada por diversos fatores. Um fator é a falta de tempo disponível nas grades curriculares já lotadas pelas disciplinas que devem ser ministradas pela LDB – Lei de Diretrizes e Bases da Educação – Lei 9.394/96. Além disso, existe também a falta de professores de computação para o Ensino Fundamental [INEP 2010-2014], o que dificulta a implantação de propostas relacionadas à computação [PCN 1998].

Assim uma solução no Ensino Fundamental pode ser o ensino da computação de forma interdisciplinar em disciplinas já existentes na grade curricular [Santomé 1998]. Neste contexto, o artigo apresenta uma unidade instrucional (UNIfICA - UNidade Instrucional Interdisciplinar de Computação e HistóriA) visando o desenvolvimento de um jogo digital sobre conteúdo da disciplina de História. Em relação à disciplina de História o objetivo geral é que o aluno compreenda a realidade nas diversas dimensões temporais, destacando questões regionais, nacionais e mundiais das diferenças entre culturas e modos de viver/pensar/fazer. Em relação ao ensino da computação enfoca-se o ensino do pensamento computacional, prática computacional e da colaboração. Para operacionalizar o ensino de programação é utilizada a ferramenta Scratch (2013), um ambiente para introduzir a programação de modo simples para crianças que não possuem nenhum tipo de experiência prévia neste assunto [Aureliano e Tedesco 2012]. O Scratch possibilita a programação de histórias interativas, jogos e animações de maneira fácil e intuitiva. Atualmente o Scratch possui uma comunidade que abrange mais de 7 milhões de usuários ativos, dos quais 155 mil são brasileiros [Scratch 2013].

Muitos estudos têm demonstrado que o Scratch contribui de forma positiva para o ensino de computação nas escolas [Resnick et al. 2009, Pazinato e Teixeira 2013, Wilson e Moffat 2010]. Contudo, a maioria desses estudos foi conduzida sobre o uso do Scratch para ensinar exclusivamente conceitos de computação [Aureliano e Tedesco 2012, Pazinato e Teixeira 2013, Wilson e Moffat 2010] ou de forma interdisciplinar em disciplinas na área de ciências exatas, como matemática [Pinto 2010, Andrade et al. 2013]. Até hoje foram realizados poucos estudos relacionados a aplicação do Scratch na área de ciências humanas [ScratchEd 2015; Franklin et al. 2011; Jimenez e Gardner-McCune 2015]. Porém, estes ou não apresentam nenhum tipo de avaliação ou apresentam unidades instrucionais em Inglês. Assim, para possibilitar o ensino de computação com SCRATCH por meio da programação de jogos de forma interdisciplinar em escolas Brasileiras necessita-se o desenvolvimento de uma unidade instrucional interdisciplinar envolvendo computação e História, que ao mesmo tempo proporciona uma aprendizagem não só efetiva e eficiente, mas também motivadora e divertida.

2. Contexto

A natureza da experiência relatada consiste no ensino de programação/conceitos básicos de computação por meio de programação de jogos com SCRATCH de forma interdisciplinar em disciplinas de História. O público alvo deste trabalho são alunos do Ensino Fundamental de escolas no Brasil com idade entre 8 e 14 anos.

Os objetos gerais de aprendizagem da unidade instrucional são: entender conceitos básicos da computação, principalmente relacionados à prática/programação e ao pensamento computacional, usar o ambiente de desenvolvimento Scratch para criar jogos reforçando os conhecimentos sobre a realidade histórica em diferentes épocas.

O conteúdo da unidade instrucional inclui: utilização do ambiente Scratch; concepção, programação e testes de um jogo com Scratch ilustrando questões relacionadas a História (como pré-história, civilizações antigas ou cultura e história de Santa Catarina). Compartilhamento e experimentação dos jogos desenvolvidos pela turma. A unidade instrucional é composta de 8 encontros de 2 horas/aulas. Todos os materiais didáticos são compartilhados sob a licença Creative Commons Atribuição-NãoComercial-Compartilhalgual 4.0 Internacional disponível no site: http://www.computacaonaescola.ufsc.br/?page_id=1476.

Figura 1: Cenas da aplicação e exemplos do material didático.

3. Procedimentos metodológicos (coleta/análise de dados)

O objetivo desta pesquisa é o desenvolvimento, a aplicação e a avaliação de uma unidade instrucional para o ensino de computação no Ensino Fundamental. Para atingir este objetivo é realizado um estudo de caso exploratório para compreender os fenômenos observados durante as aplicações da unidade instrucional em um contexto particular e identificar direcionamentos para trabalhos futuros (Fig. 2).

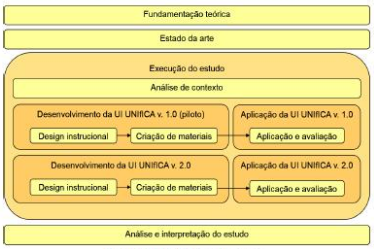


Figura 2: Metodologia de pesquisa.

O estudo de caso é realizado conforme os procedimentos propostos por Yin (2013) e Wohlin et. al (2012):

Definição do estudo: O estudo é definido em termos do objetivo e perguntas de pesquisa e o design de pesquisa. A partir do objetivo e das perguntas de análise são sistematicamente derivadas as medidas para a coleta de dados utilizando o método GQM [Basili, Caldeira e Rombach 1994]. Para a operacionalização da coleta de dados são definidos instrumentos de coleta de dados.

Execução do estudo: A execução do estudo é realizada adotando o modelo ADDIE [Branch 2009] como abordagem para o design instrucional. Em uma primeira etapa a unidade instrucional é desenvolvida. Para isto, são caracterizados os aprendizes e o ambiente. São levantadas as necessidades de aprendizagem e definidos os objetivos de aprendizagem. De acordo com a análise de contexto, é projetada a unidade instrucional, definindo o seu conteúdo, a sequência e os métodos instrucionais a serem adotados. Em seguida, o material instrucional é desenvolvido. Durante a segunda etapa da execução do estudo, a unidade instrucional é aplicada em duas iterações na prática e avaliada, coletando-se os dados.

Análise e Interpretação do estudo: Nesta etapa são analisados os dados em relação às perguntas de pesquisa, usando métodos quantitativos e qualitativos. Ao final, os resultados são interpretados e discutidos.

4. Principais conclusões

A unidade instrucional desenvolvida foi aplicada em quatro turmas de uma escola privada localizada em Florianópolis/SC em 2015 e em uma turma de uma escola pública localizada em Palhoça/SC em 2016 (Fig. 3).

Figura 3: Cenas das aplicações da unidade instrucional.

As aplicações demonstram que o ensino de computação usando Scratch pode ser adotado com sucesso já no Ensino Fundamental de forma interdisciplinar. Por meio de respostas de questionários antes/depois da aplicação, verificou-se a partir da autoavaliação pelos alunos, mais de 60% sabe criar um programa de computar após a unidade (Fig. 4).

* Este artigo está sendo submetido de forma paralela como resumo para um capítulo do livro "Computação na Educação Básica: Fundamentos e Experiências" sendo organizado pelo Centro Lemann da Universidade de Stanford e a Sociedade Brasileira de Computação.

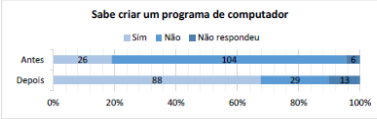


Figura 4: Percepção dos alunos de antes/depois da unidade.

A unidade instrucional permitiu o ensino e aprendizagem de vários conceitos de computação, incluindo principalmente a programação e o pensamento computacional além de reforçar as competências referentes à disciplina de História, comparando aspectos culturais e políticos de diferentes épocas. Diversos tipos de jogos foram desenvolvidos pelos alunos relacionados aos conteúdos das respectivas disciplinas de História: cultura regional (5º ano), Pré-História (6º ano) e Civilizações Antigas (7º ano) (Fig. 5).

Figura 5: Exemplos de jogos desenvolvidos pelos alunos.

Observou-se que o ambiente Scratch também facilita a aprendizagem de diversos conceitos de programação, como: variáveis, eventos, condicionais, laços de repetição e paralelismo. A maior parte dos alunos também achou as aulas fáceis de maneira geral (Fig. 6).

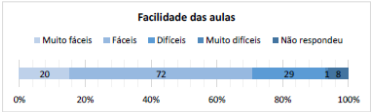


Figura 6: Percepção da facilidade das aulas pelos alunos.

Foi observado também, que as aulas motivaram os alunos a aprender mais sobre coisas novas e promoveram uma experiência de aprendizagem agradável e muito satisfatória para os alunos, já que mais de 75% dos alunos achou as aulas divertidas ou muito divertidas (Fig. 7).

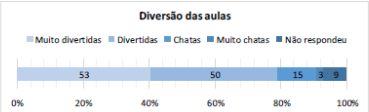


Figura 7: Percepção da diversão das aulas pelos alunos.

Muitos alunos também demonstraram grande interesse em aprender mais sobre computação e mais de 70% dos alunos apresentam satisfação em fazer programas de computador (Fig. 8). Ao produzirem os jogos no ambiente Scratch os alunos tornam-se pesquisadores do conteúdo, e não meros receptores, os jogos auxiliaram não só no aprendizado da linguagem computacional, mas também na construção do conhecimento histórico. Além de estudarem as temáticas eles as reelaboraram e deram suas interpretações acerca do passado deixando a disciplina de história mais palpável e fácil de aprender.

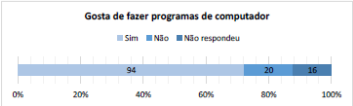


Figura 8: Satisfação em fazer programas de computador pelos alunos.

5. Experiência da equipe proponente

A oficina foi desenvolvida como parte da Iniciativa Computação na Escola, coordenada pelo Departamento de Informática e Estatística (INE) da Universidade Federal de Santa Catarina (UFSC), que é dedicada a estimular o ensino de computação na escola primária.

A oficina é resultado do TCC e de projeto de iniciação científica da Nathalia da Cruz Alves sob a orientação da Profa. Dr. rer. nat. Christiane Gresse von Wangenheim, PMP, professora do Departamento de Informática e Estatística (INE) da UFSC. Seus principais interesses de pesquisa são na área de qualidade de software, ensino de engenharia de software e o ensino/popularização da computação. Recebeu o título de Diplom-Informatikerin pela Universidade de Kaiserlautern (Alemanha) em 1995; o título de Dr. rer. nat. pelo Departamento de Informática da Universidade de Kaiserslautern (Alemanha) em 2002. Ela também é PMP – Project Management Professional – certificada pelo Project Management Institute (PMI) e Implementadora MPS.BR/Avaliadora Adjunta MPS.BR/SOFTEX.

Jean Carlo Rossa Hauck é professor do Departamento de Informática e Estatística (INE) da Universidade Federal de Santa Catarina (UFSC). Seus principais interesses de pesquisa são relacionados ao ensino da Engenharia de Software, Gerência de Projetos e Melhoria de Processos de Software. Como professor e pesquisador, atuou anteriormente na Universidade do Sul de Santa Catarina (UNISUL) e na Universidade do Vale do Itajaí (UNIVALI). Foi pesquisador visitante no *Regulated Software Research Centre - Dundalk Institute of Technology* - Irlanda.

Adriano Ferreti Borgatto é professor do Departamento de Informática e Estatística (INE) da Universidade Federal de Santa Catarina (UFSC). Sua principal linha de pesquisa é a Teoria da Resposta ao Item com publicações na área da educação. Como professor atua ministrando disciplinas e orientando alunos de mestrado e doutorado em dois programas de pós-graduação, sendo um na área de avaliação educacional e outro na área de Atividade Física e Saúde na Educação Física.

Christiane A. Gresse von Wangenheim, Jean C. R. Hauck e Adriano F. Borgatto coordenam o GQS - Grupo de Pesquisa de Qualidade de Software (www.gqs.ufsc.br), que enfoca na pesquisa científica, desenvolvimento e transferência de modelos, métodos e ferramentas de engenharia de software e do ensino da engenharia de software para apoiar a melhoria da qualidade de software. Eles também coordenam/participam da iniciativa Computação na Escola (www.computacaonaescola.ufsc.br) dedicada ao ensino e popularização da computação no ensino básico, envolvendo a pesquisa científica, desenvolvimento, aplicação e avaliação de unidades instrucionais inovadoras para o ensino de computação.

Pedro Eurico Rodrigues é doutorando em História Social pela Universidade de São Paulo (USP), é bacharel e licenciado em história pela UDESC, e é mestre pela mesma instituição, além de atuar como professor de história no ensino fundamental da rede particular de Florianópolis.

Este trabalho é apoiado pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

6. Referências

Andrade, M. et al. "Desenvolvendo games e aprendendo matemática utilizando o Scratch". Anais do SBGames, São Paulo/SP, 2013

Aureliano, V. C. O. & Tedesco, P. C. A. R. "Avaliando o uso do Scratch como abordagem Alternativa para o processo de ensino-aprendizagem de programação". Anais do XX Workshop sobre Educação em Computação, Curitiba/Paraná, 2012.

Basili, V. R.; Caldeira, G. & Rombach, H. D. "Goal Question Metric Paradigm". In Encyclopedia of Software Engineering, John Wiley & Sons, 1994.

Branch, R. M. "Instructional Design: The ADDIE Approach". New York: Springer, 2009.

Code.org. 2013. Disponível: http://code.org. Acesso: jul. 2016.

Franklin, D. et al. "Animal tlatoque, Attracting middle school students to computing through culturally-relevant themes". Proceedings of the 42nd ACM Technical Symposium on Computer Science Education. NY, EUA: ACM. 2011. p. 453-458.

INEP. "Sinopses Estatísticas da Educação Superior – Graduação". Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Brasília. 2010 a 2014.

Jimenez, Y. & Gardner-McCune, C. "Using App inventor & history as a gateway to engage African American students in computer Science". Research in Equity and Sustained Participation in Engineering, Computing, and Technology. Charlotte, NC, EUA, 2015. p. 1-2.

MEC. PCN. "Parâmetros Curriculares Nacionais". Brasília, 1998.

MIT. Scratch. 2013. Disponível: http://scratch.mit.edu. Acesso: jul. 2016.

Pazinato, A. M. & Teixeira, A. C. O. "Uso do Software SCRATCH no Desenvolvimento da Aprendizagem e na Interação Construtivista dos Alunos". Anais do XI Congresso Nacional de Educação, Curitiba/Paraná, 2013.

Pinto, A. S. "Scratch na aprendizagem da Matemática no 1.ºCiclo do Ensino Básico: estudo de caso na resolução de problemas". 2010. Dissertação de Mestrado em Estudos da Criança. Instituto de Educação. Universidade do Minho, Braga/ Portugal, 2010

Resnick, M. et al. "Scratch: programming for all". Communications of the ACM, 52(11), 2009, pp. 60-67

Santomé, J. T. "Globalização e interdisciplinaridade". Porto Alegre: Artmed, 1998.

ScratchED - Relato de Karen Randall. "Embedding Scratch in US History/Geography". Disponível: http://scratched.gse.harvard.edu/resources/embedding-scratch-us-historygeography. Acesso: mar. 2015.

Wohlin, C. et al. "Experimentation in Software Engineering". Springer Verlag, 2012.

Wilson, A. & Moffat, D. C. "Evaluating Scratch to introduce younger schoolchildren to programming". Proc. of the Psychology of Programming Interest Group Workshop, Madrid/Espanha, 2010

Yin, R. K. "Case Study Research: Design and Methods". SAGE Publications, Inc; 5. Edição, 2013.