

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS APLICADAS A EDUCAÇÃO
DEPARTAMENTO DE CIÊNCIAS EXATAS
LICENCIATURA EM CIÊNCIA DA COMPUTAÇÃO

HÉVELLYN DE MORAIS RABÊLO

**A PROGRAMAÇÃO COM SCRATCH NA PRODUÇÃO DE
RECURSOS NAS DISCIPLINAS DE ARTE E CULTURA E
LÍNGUA TUPI**

RIO TINTO - PB
2015

HÉVELLYN DE MORAIS RABÊLO

**A PROGRAMAÇÃO COM SCRATCH NA PRODUÇÃO DE
RECURSOS NAS DISCIPLINAS DE ARTE E CULTURA E
LÍNGUA TUPI**

Monografia apresentada ao Curso de Licenciatura em Ciências da Computação do Centro de Ciências Aplicadas e Educação (CCAIE), Campus IV da Universidade Federal da Paraíba, para obtenção do título de licenciado.

Orientadora: Profa. MSc. Ana Liz Souto Oliveira de Araújo.

RIO TINTO - PB
2015

Ficha catalográfica preparada pela Seção de Catalogação e Classificação da Biblioteca da UFPB

R112p Rabêlo, Hévellyn de Moraes.

A programação com Scratch na produção de recursos nas disciplinas de arte e cultura e língua Tupi. / Hévellyn de Moraes Rabêlo. – Rio Tinto: [s.n.], 2015.
71f. : il.

Orientador(a): Prof. Msc. Ana Liz Souto Oliveira de Araújo.
Monografia (Graduação) – UFPB/CCAIE.

1. Programação - computação. 2. Scratch - programação. 3. Escola indígena.

UFPB/BS-CCAIE

CDU: 004.021

Hévellyn de Moraes Rabêlo

A PROGRAMAÇÃO COM SCRATCH NA PRODUÇÃO DE RECURSOS NAS DISCIPLINAS DE ARTE E CULTURA E LÍNGUA TUPI

Trabalho de Conclusão de Curso submetido ao Curso de Licenciatura em Ciência da Computação da Universidade Federal da Paraíba, Campus IV, como parte dos requisitos necessários para obtenção do grau de LICENCIADO EM CIÊNCIAS DA COMPUTAÇÃO.

Assinatura do autor: _____

APROVADO EM: _____/_____/_____

Orientadora: Profa. MSc. Ana Liz Souto
Oliveira de Araújo

Universidade Federal da Paraíba – Campus IV

Profa. MSc. Pasqueline Dantas Scaico

Universidade Federal da Paraíba – Campus IV

Profa. MSc. Thaíse Kelly de Lima

Universidade Federal da Paraíba – Campus IV

RIO TINTO - PB
2015

“Maquina, tua tecnologia foi semente de todo o ensinamento que recebi. Foste companheira inseparável durante quatro anos. Tempo suficiente para que eu jamais me esqueça de teus grandes olhos luminosos abertos até a madrugada, de tuas teclas gordurosas tantos foram os dedos que as tocaram, de vossos cartuchos de tinta dançantes que tão bem apresentavam meus trabalhos em papel. Tua tecnologia pode ter evoluído, talvez, mais rápido que eu próprio, mas perdeu com o tempo a característica de semente e tornou-se toda a minha árvore de aprendizado. E tu, venerável máquina causadora de miopias, serás instrumento de meu trabalho”.

Alexandre Nóbrega Duarte

AGRADECIMENTOS

Agradeço a priori a Deus, que me deu a graça da vida, que iluminou meus passos, me deu discernimento e sabedoria para fazer as escolhas que me proporcionaram chegar nessa fase tão almejada. E colocou verdadeiros anjos na terra para me ajudar nessa dádiva de viagem fantástica ao qual ousamos chamar de vida.

A minha mãe Maria Ivonete D. Morais, que por uma fonte inesgotável de amor que brota em seu peito me acompanhou em todos os momentos, mudando todo um estilo de vida para me proporcionar a benção de hoje estar concluindo este curso. Uma guerreira que sempre foi dedicada em sua responsabilidade materna e nada me deixou faltar. Que me ensinou muito mais que conteúdos didáticos, mas me ensinou a beleza de sentimentos e princípios que me tornaram o que hoje eu sou, fortalecendo o meu caráter e construindo a minha personalidade. Que me ensinou uma base religiosa. Que foi enfermeira, médica, cozinheira, amiga, confidente, professora e cientista. Que me mostrou o primeiro arco-íris. Deu-me o primeiro beijo e acima de tudo mesmo reconhecendo os meus inúmeros defeitos, nunca desistiu de ser minha mãe. Ao meu pai Digler Rabêlo, que ajudou a me gerar, me ensinou, me amou e tantas vezes compreendeu minha ausência.

Aos meus irmãos Rostand, Hélder, Kassiana e Digler Neto que estiveram presentes em todos os momentos da minha vida, que me amaram e me proporcionaram experiências incríveis, me passaram confiança e esperança, me apoiaram em minhas escolhas, auxiliaram nas minhas dúvidas. De fato nenhum me mostrou o atalho sem pedras, mas me ensinaram como transpassar cada uma que encontrei nas estradas que segui. Agradeço especialmente por terem me dado três joias que são meus sobrinhos. Felipe, Mariana e Rafael. Muito obrigada por fazer a vida de tia mais feliz, mais espontânea, mais leve.

Agradeço a um presente de Deus, que atende pelo nome de Wênnio Kelson. Este suportou todas as minhas inseguranças com paciência, me ensinou a confiar no meu potencial, há mesmo com medo não deixar de arriscar. Esteve presente todo esse tempo de curso. Em muitos momentos acreditou mais em mim do que eu mesma. Companheiro inseparável de vida e de profissão. Tantos foram os trabalhos realizados juntos, experiências que compartilhamos e dificuldades superadas para chegar até aqui. Mas chegamos e o melhor de tudo: Chegamos juntos! E sei que juntos vamos muito mais longe, pois juntos somos ainda mais fortes e somos ainda melhores.

Gostaria de agradecer a minha família como um todo, por compreender a ausência e a distância. Especial tia Agui tantas ligações que fiz para desabafar, mais do que eu não via a hora da apresentação deste trabalho. Vanderson, primo querido tantas foram às ligações de socorro ao longo do curso, que fizeram de você o meu Google particular. Aos meus melhores amigos: Luan Luna que tanto me ajudou ao longo do curso, inclusive com contribuições especiais neste trabalho. Elineide dos Santos, Izabelly Pontes, Jefferson Fernandes e Andressa Fabião, amigos de infância, de uma vida inteira, com quem compartilhei tristezas e alegrias, medos e descobertas. Que mesmo longe se fizeram presentes e acompanharam todos os momentos da minha vida inclusive este. Meus irmãos do coração!

As minhas mestras Ana Liz que orientou todo esse trabalho, auxiliou, incentivou, dedicou parte das férias, com uma atenção, paciência e préstimo o qual nunca vi, dando o seu melhor e tirando o meu melhor para conceber este trabalho. Thaíse e Pasqueline que se destacaram em minha graduação e aceitaram fazer parte da minha banca. Que mesmo afastadas para o doutorado e distante, trouxeram nesse momento suas considerações tão importantes para que eu pudesse desenvolver um trabalho como sempre sonhei, capaz de ser útil ao meu próximo.

Agradeço a todos os docentes que influenciaram direta e indiretamente na construção da minha identidade profissional, me trazendo experiências teóricas e práticas. A família PIBID que me acolheu tão bem quando entrei no projeto. Aos funcionários, que cada um com suas respectivas incumbências transformaram o Campus IV em um lugar acolhedor e repleto de amigos. Aos colegas do curso que compartilharam as mesmas alegrias, as mesmas dúvidas e hoje compartilham a concretização do mesmo sonho.

RESUMO

Integrar o ensino de computação ao currículo do ensino básico é um desafio atual da área de informática na educação. Nessa vertente, este trabalho é uma iniciativa que apresenta o planejamento, a execução e a avaliação de um curso que procurou integrar o pensamento computacional por meio da programação nas disciplinas de Arte e Cultura e Língua Tupi em escolas de Rio Tinto/PB. Então, foi proposto às escolas indígenas um curso visando estimular essas disciplinas, por meio da produção de recursos desenvolvidos na linguagem de programação *Scratch*, abordando conteúdos da cultura indígena. Tal curso tem como objetivo desenvolver o pensamento computacional nos alunos, por meio do ensino de conceitos de programação transmitidos em uma linguagem lúdica e interativa. O curso foi ministrado com carga horária total de 16h e foi norteado pela teoria de aprendizagem construtivista. Ao final, as produções dos alunos foram avaliadas segundo um modelo para mensurar a aplicação do pensamento computacional, bem como sua percepção a respeito dos assuntos de computação. Como resultado, percebeu-se que os alunos conseguiram atingir em sua maioria os melhores níveis de desenvolvimento do modelo de avaliação utilizado, conseguindo assim atingir com bons resultados o objetivo.

Palavras Chave: *Scratch*, Pensamento computacional, Escola indígena, Ensino de programação.

ABSTRACT

It is a current challenge integrate computer into the educational curriculum. This work is an initiative that presents the planning, execution and evaluation of a course that integrate computational thinking by programming in discipline of Art and Culture and Tupi Language in Rio Tinto/PB. Then, it was proposed a course to indigenous schools to stimulate these disciplines, through the production of resources developed in the Scratch programming language, covering contents of indigenous culture. This course aims to develop computational thinking in students, through teaching programming concepts transmitted in a fun and interactive language. The course was taught with 16 hours and was guided by the constructivist learning theory. In the end, the resources developed were evaluated by a model to measure the application of computational thinking as well as their perception of computing issues. As a result, it was noticed that students mostly have achieved the best levels of development according the assessment model, thereby achieving good results.

Keywords: Scratch, computational thinking, indigenous culture, teaching programming.

LISTA DE FIGURAS

Figura 1 - Versão 1.4 do <i>Scratch</i> (off-line)	22
Figura 2 - Gráfico de avaliação da primeira atividade	33
Figura 3 - Gráfico de avaliação da segunda atividade.....	34

LISTA DE QUADROS

Quadro 1 - Modelo PECT Seiter(2013)	31
Quadro 2 - Modelo de avaliação desse trabalho.....	32

LISTA DE SIGLAS

MIT	Instituto de Tecnologia de Massachusetts
O.A.	Objeto de Aprendizagem
S.E.	Software Educacional
S.O.	Sistemas Operacionais
CSTA	Computer Science Teachers Association
ISTE	Internacional Society for Technology in Education
PECT	Progression of Early Computational Thinking

SUMÁRIO

1 INTRODUÇÃO	14
1.1 MOTIVAÇÃO	14
1.2 JUSTIFICATIVA	16
1.3 OBJETIVOS	17
1.3.1 Objetivo Geral	17
1.3.2 Objetivos Específicos	17
1.4 ESTRUTURA DO TRABALHO	17
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 PENSAMENTO COMPUTACIONAL	18
2.2 SCRATCH	20
3 PLANEJAMENTO E APLICAÇÃO DO CURSO	24
3.1 AS ESCOLAS	24
3.2 INTERDISCIPLINARIDADE DE SCRATCH COM ARTE E CULTURA E LÍNGUA TUPI	24
3.3 PLANEJAMENTO DO CURSO	25
3.4 APLICAÇÃO DO CURSO	27
4 AVALIAÇÃO	30
4.1 MODELO PECT	30
4.2 AVALIAÇÃO DAS ATIVIDADES	33
5 CONCLUSÃO	38
REFERÊNCIAS	40
APÊNDICES	42
ANEXOS	58

1 INTRODUÇÃO

Neste capítulo serão apresentados os tópicos de motivação, o objetivo geral e os objetivos específicos, a justificativa deste trabalho bem como a estrutura na qual ele está desenvolvido.

1.1 MOTIVAÇÃO

O uso da informática como ferramenta auxiliar no ensino-aprendizagem das disciplinas do currículo escolar brasileiro é de extrema importância tendo em vista que pode facilitar a assimilação dos mais diversos conteúdos. Eles podem ser trabalhados de forma lúdica e interativa, despertando maior interesse nos alunos de todas as séries, em especial, os de nível médio, cujo principal objetivo geralmente é alcançar a aprovação no vestibular e o ingresso no nível superior.

Entretanto muitas iniciativas para introduzir a informática na escola se dão por meio de cursos que trabalhem funções elementares dos sistemas operacionais mais comuns, como, por exemplo, o *Windows* e o *Linux*. Ou ainda o uso de *softwares* como o *Office*, ou ensinando os alunos a utilizarem a *Internet* para fazerem pesquisas ou mesmo criar e-mails. Porém, a maioria dos alunos já sabe acessar a *Internet* e até usam redes sociais para se comunicar com os amigos. Observa-se que essas iniciativas nem sempre conseguem auxiliar no processo de ensino-aprendizagem de conteúdos de outras disciplinas do currículo, como química, física, biologia.

A carência de cursos que trabalhem nos alunos a interdisciplinaridade entre a informática e as disciplinas do currículo escolar brasileiro ainda é grande. É preciso iniciativas nas quais os alunos possam desenvolver o pensamento computacional e trabalhar habilidades cognitivas que possam colaborar com a melhora do seu desempenho na sala de aula. Dessa forma, eles terão uma nova visão a respeito da computação, e de como ela pode fornecer mecanismos que auxiliem atividades do cotidiano.

O pensamento computacional pode ser compreendido como uma forma de pensar e resolver problemas utilizando conceitos da ciência da computação Wing (2006). Trate-se de um conjunto de conceitos, habilidades, práticas e ferramentas da computação que podem ser aplicadas tanto no saber usado no cotidiano como em diversas áreas do conhecimento científico.

Com esses aprendizados acerca do pensamento computacional, os alunos deixam de ser agentes passivos que usam a informática de forma superficial, apenas consumindo os recursos que a mesma oferece, e passam a ser agentes ativos, que conhecem e aplicam conceitos da Ciência da Computação. E de posse desses conhecimentos, buscam utilizá-los para melhorar as situações cotidianas, solucionando problemas do seu meio. Os exemplos podem ser diversos, desde utilizar ferramentas que contribuam no seu processo educacional, interagindo com mais propriedade com os recursos tecnológicos disponíveis, e até mesmo desenvolvendo recursos que viabilizem melhorias em determinados problemas da sociedade a qual fazem parte.

É importante que os alunos aprendam cada vez mais cedo a pensar de forma computacional. Isso irá permitir que esses se destaquem no mercado de trabalho, de forma que possam sugerir ou lidar com tecnologias que eles saibam que irão ajudar sua área de trabalho ou pesquisa. Mas então, como ou o que deve ser feito para capacitar os alunos de forma que esses desenvolvam tais habilidades?

Neste contexto, surgem iniciativas de integrar o desenvolvimento do pensamento computacional com os conteúdos do ensino básico usando ferramentas computacionais. Uma das ferramentas que pode ser planejada para ser utilizada nesse meio é o *Scratch*, como relata Scratch (2011).

O *Scratch* é uma linguagem de programação de nível introdutório. Seu objetivo é ensinar programação para quem ainda não sabe programar, tendo em vista que ele é um software educacional lúdico, totalmente interativo, que trabalha com o paradigma visual, além de ser facilmente utilizado para desenvolver os programas.

No *Scratch*, os programas são construídos por meio de blocos de encaixe, atrelados uns aos outros, até que o código seja finalizado, o que remete a ideia do jogo: LEGO. Ele não exige, a priori, do usuário nenhum conhecimento básico em programação, apenas o uso do raciocínio lógico e da criatividade. Outro grande atrativo é o uso de recursos midiáticos na construção de jogos, animações, programas em contextos diversos.

Diversos autores afirmam que o processo de ensino-aprendizagem de conceitos introdutórios de programação por meio da linguagem *Scratch* são fatores de sucesso, como mostram Araújo et al, (2013) e Pereira et al (2012). Experiências relatam o desenvolvimento de criatividade, raciocínio lógico, pensamento abstrato, por meio dessa ferramenta. Entretanto, Cambraia e Oliveira (2012) afirmam que a integração entre a programação e disciplinas de núcleo comum como português, matemática e física não é tarefa trivial. Há

desafios a serem suplantados, desde o acesso às máquinas até o conhecimento de como utilizar seus recursos. É necessário planejamento e integração entre licenciados em computação e professores dessas disciplinas.

Diante desse contexto, foi percebida a oportunidade de se planejar atividades interdisciplinares visando desenvolver o pensamento computacional integrado com as disciplinas de Arte e Cultura e Língua Tupi do currículo básico. As atividades visam desenvolver o pensamento computacional nos alunos, por meio do ensino de conceitos de programação com *Scratch*, principalmente por usar recursos midiáticos e que serão transmitidos fazendo paralelos com a cultura dos alunos.

1.2 JUSTIFICATIVA

Este documento relata o planejamento e desenvolvimento de atividades interdisciplinares de conceitos do pensamento computacional em disciplina no currículo básico. Apesar de existirem outros trabalhos que abordem interdisciplinaridade, aqui se tem uma peculiaridade que se dá pelo seu contexto sociocultural não ser comumente abordado nos mesmos.

O trabalho em questão foi desenvolvido no município de Rio Tinto/PB, no qual está localizado o Campus IV da Universidade Federal da Paraíba que possui o curso de Licenciatura em Ciência da Computação. Essa região tem como particularidade o fato de ter um grande número de habitantes índios ou descendentes de índios, e possui escolas públicas destinadas exclusivamente a eles. Elas buscam resgatar a cultura por meio de disciplinas específicas, como Arte e Cultura e de Língua Tupi. Que foram incorporadas à grade curricular e ao projeto político pedagógico devido à singularidade da localidade.

Essas disciplinas foram selecionadas pela pesquisa com o objetivo de valorizar, realçar e fortalecer a importância da cultura, arte e língua para os índios e descendentes, evitando que esse conhecimento vá se extinguindo. As disciplinas propiciam que os costumes e as crenças dos estudantes sejam resgatadas e preservadas para se tornarem conhecidas pelos demais brasileiros, a fim de aproveitar a riqueza cultural desse povo.

Para isto, foram desenvolvidas as atividades desta pesquisa, focando utilizar os conhecimentos que eles já possuem de sua cultura, tendo em vista que os alunos que participaram deste estudo são todos descendentes de indígenas. Os assuntos foram abordados em atividades nas quais os alunos tiveram uma temática livre, ou seja, eles criaram animações

usando a própria criatividade e dentro do contexto indígena e que estavam sendo reforçados nas disciplinas em questão. Dessa forma, este trabalho também visa auxiliar o processo de ensino-aprendizagem dessas duas disciplinas, estimulando os alunos a trabalharem as temáticas de forma mais lúdica, fugindo do tradicionalismo da sala de aula. Por se tratar de assuntos que os alunos já conhecem de cedo, por fazer parte da sua cultura e da sua rotina, e aprofundam nessas disciplinas, julgou-se inovador capacitá-los a expressar seus conhecimentos por meio de animações com *Scratch*.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Estimular o pensamento computacional através de atividades interdisciplinares integrando programação, arte e cultura e língua Tupi com alunos do 9º ano do Ensino Fundamental e 1º ano do Ensino Médio.

1.3.2 Objetivos Específicos

1. Apresentar o pensamento computacional e os conceitos introdutórios de programação por meio da ministração de um curso da ferramenta *Scratch*.
2. Produzir atividades que estimulem o pensamento computacional e que favoreçam a interdisciplinaridade através de conceitos de arte e cultura e língua tupi.
3. Avaliar as atividades dos alunos segundo um modelo para mensurar a aplicação do pensamento computacional, bem como sua percepção a respeito dos assuntos de computação.

1.4 ESTRUTURA DO TRABALHO

O capítulo I apresenta a introdução, abrangendo a motivação, a justificativa, o objetivo geral bem como os objetivos específicos, e a estrutura do trabalho. O capítulo II aborda a fundamentação teórica, dissertando neste tópico sobre o pensamento computacional e o *Scratch*. O capítulo III fala das escolas as quais receberam esta proposta de curso e sobre a interdisciplinaridade entre o *Scratch* e as disciplinas de Arte e Cultura e Língua Tupi, bem como do planejamento do curso e sua aplicação. Finalmente o Capítulo IV apresenta a avaliação das atividades desenvolvidas pelos alunos ao longo do curso.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção descrevem-se trabalhos relacionados ao tema, como pensamento computacional (seção 2.1) e *Scratch* (seção 2.2).

2.1 PENSAMENTO COMPUTACIONAL

A informática está presente em diversas áreas de conhecimento, fazendo parte da maioria das atividades humanas, sendo muito difícil pensar nas pessoas longe dos computadores, como relata Nunes (2008). Sendo assim, é imprescindível que um dos conhecimentos desenvolvidos para o bom exercício da cidadania atualmente seja o pensamento computacional, como defende Blikstein (2008).

A *Computer Science Teachers Association*¹ (CSTA) reconheceu a relevância do pensamento computacional na educação básica e criou a *Computational Thinking Task Force* com o objetivo de explorar, disseminar e recomendar recursos para ensino e aprendizagem relacionados ao pensamento computacional. Ela conta com a parceria da *Internacional Society for Technology in Education* (ISTE) para desenvolver definições operacionais do pensamento computacional e oferecer ferramentas e recursos para os professores utilizarem em todos os níveis de ensino e em áreas de estudo.

A CSTA define o pensamento computacional como um processo de resolução de problemas nos quais estão relacionadas habilidades de:

- Formular problemas de forma a se beneficiar de conceitos, de ferramentas e de recursos computacionais para ajudar a resolvê-los.
- Organizar e analisar dados.
- Representar dados através de abstrações, identificando modelos e realizando simulações.
- Automatizar soluções através do pensamento algorítmico.
- Identificar, analisar e implementar soluções com o objetivo de alcançar a combinação mais eficiente e eficaz de passos e recursos.
- Generalizar e transferir o processo de solução de um problema para uma gama de problemas semelhantes.

¹ <http://csta.acm.org/Curriculum/sub/CompThinking.html#CTResources>. Acessado em: 20 nov. 2014.

Essas habilidades são apoiadas e reforçadas por um conjunto de atitudes essenciais a prática do pensamento computacional. Tais atitudes estão relacionadas à capacidade de: lidar com problemas com diferentes níveis de complexidades; ter persistência ao trabalhar com problemas difíceis; lidar com problemas ainda não resolvidos; e se comunicar e trabalhar em grupo para atingir um objetivo em comum.

Em suma, o pensamento computacional se propõe a desenvolver habilidades cognitivas como, por exemplo, raciocínio lógico, abstração e decomposição que podem ser utilizados na resolução de problemas das mais diversas áreas, como relata Wing (2008). Dessa forma, o pensamento computacional permite que quando as pessoas se depararem com problemas difíceis de resolver, passem a pensar neles como um conjunto de problemas menores, passíveis de resolução. Assim, a solução desses problemas menores compõe a solução do problema original.

Pensar computacionalmente muitas vezes pode estar relacionado a criar um algoritmo, ou seja, descrever suas ideias em forma de passos. Esses passos podem ser codificados em uma determinada linguagem de programação, desde que a construção desses códigos proponha alcançar um objetivo capaz de solucionar problemas cotidianos de forma sistemática e automática.

Resnick et al (2011) apresentaram um quadro listando os conceitos e as práticas computacionais esperadas quando se desenvolvem atividades que buscam explorar o pensamento computacional. Os conceitos são as definições comumente utilizadas nas disciplinas da computação. Os autores listam:

- Sequência: é a identificação de um conjunto de passos para se realizar uma tarefa.
- Loop: é a repetição de uma sequência de passos.
- Paralelismo: consiste em fazer que eventos aconteçam simultaneamente.
- Eventos: algo que faz com que um acontecimento produza uma ação.
- Condicionais: permitem que decisões sejam tomadas tendo em vista condições pré-definidas.
- Operadores: fornece suporte para expressões matemáticas e lógicas.
- Dados: valores são armazenados, retornados e atualizados.

Com relação às práticas computacionais, Resnick et al (2011) listam:

1. Iterativo e incremental: trata-se de desenvolver uma parte da solução, testar essa parte e depois produzir outra parte. Em seguida, repetir o ciclo até que a solução seja implementada completamente.
2. Testar e debugar: testa-se se o que foi produzido funciona corretamente. Caso contrário, encoraja a encontrar e corrigir os erros.
3. Reusar e recombinar: estimula a, quando se for construir algo, tentar fazê-lo usando algo que outra pessoa ou você já construiu.
4. Abstrair e modularizar: trata-se de construir algo grande combinando partes menores.

Para quem estuda cursos na área da computação, o pensamento computacional é desenvolvido por meio de uma somatória dessas habilidades as quais são trabalhadas ao longo das disciplinas no curso técnico ou superior. Entretanto, como já argumentado, a literatura relata a importância de compreender este pensamento o quanto antes. Dessa forma, é importante que o pensamento computacional seja incorporado na educação básica, no intuito de explorar os seus benefícios, com, por exemplo: (i) atividades interativas e incrementais, para que, aos poucos e em etapas, os alunos conquistem os conhecimentos desejados; (ii) realização de testes e correções de erros, no intuito dos alunos serem engajados a buscarem soluções para seus próprios erros em cima de sua produção, (iii) reuso de recursos preexistentes e atividades anteriores, para redução de esforços repetitivos, (iv) abstração e modularização, no intuito de trabalhar com diferentes níveis de imaginação.

Além disso, trabalhar com perspectivas que exigem de um usuário ativo da computação a capacidade de questionar, de se conectar e expressar seus pensamentos por meio das práticas aprendidas são características esperadas por quem pensa computacionalmente. Bem como a aplicação dos conceitos como, por exemplo, **sequência** onde o aluno aprende a escrever o que lhe é necessário em forma de código respeitando um determinado padrão e transformando em algoritmo; ou mesmo utilizando os *loops* uma vez que dentro de sua sequência deseja otimizar algum desempenho; é possível citar também o **paralelismo** onde séries de acontecimentos seguem ocorrendo em paralelo; outros exemplos são os **eventos**, as **condições** que determinam a ocorrência de um evento, os **operadores** que auxiliam a manipulação dos dados.

2.2 SCRATCH

O *Scratch* foi criado pelo *Massachusetts Institute of Technology* (MIT), no ano de 2003, e se trata de um *software* educacional que tem como propósito ensinar programação Lifelong Kindergarten (2007). Ele tem em sua interface um ambiente de desenvolvimento atrelado a uma linguagem de programação, permitindo que o usuário, ao mesmo tempo em que acesse a sintaxe desta linguagem, construa seu código.

O *Scratch* possui versões *off-line* e *online*, a versão *off-line* atual é a versão 2.0, a versão *off-line* anterior a 2.0 é a 1.4, sendo ela utilizada no desenvolvimento das atividades deste projeto, uma vez que a mesma já se encontrava instalada nos laboratórios. Já a versão *online* é acessada diretamente pelo *site* do *Scratch*, nele é possível encontrar as duas versões *off-line* da ferramenta, para *download*, gratuitamente e em Português, estando às mesmas compatíveis com os sistemas operacionais (S.O.) mais utilizados, como, por exemplo, *Windows*, *Linux* e *Mac*.

Este objeto de aprendizagem (O.A.) é uma ferramenta que permite ao usuário uma experiência de programação visual, propiciando explorar práticas, perspectivas e conceitos computacionais, já citados na seção anterior, e que são fundamentais na dispersão do pensamento computacional. Segundo Brennan (2011) tais práticas, perspectivas e conceitos são trabalhados no *Scratch* de forma criativa onde a aprendizagem está baseada no conceito de design e que, por sua vez, enfoca quatro processos distintos, sendo eles: o processo de **concepção**, ou seja, a arte de produzir ao invés de simplesmente utilizar de maneira interativa; a **personalização** que permite as produções serem relevantes e significativas ao produtor; a **reflexão** que interpreta e repensa as atividades dos colegas e por fim a **colaboração** relacionada ao processo de desenvolver atividades em grupo.

O *Scratch* então disponibiliza ao usuário o despertar de habilidades importantes para o processo de aprendizagem. Um exemplo dessas habilidades é a aquisição de informação, adquirida ao usar os diversos tipos de mídia como imagens, textos e áudios, podendo esses serem importados ou gravados. Com esses recursos, é possível produzirem seus projetos e com eles trabalharem a comunicação. Outro fator importante para isso é a disponibilização de uma versão *online* da ferramenta, que facilita a relação entre os membros que fazem uso dela.

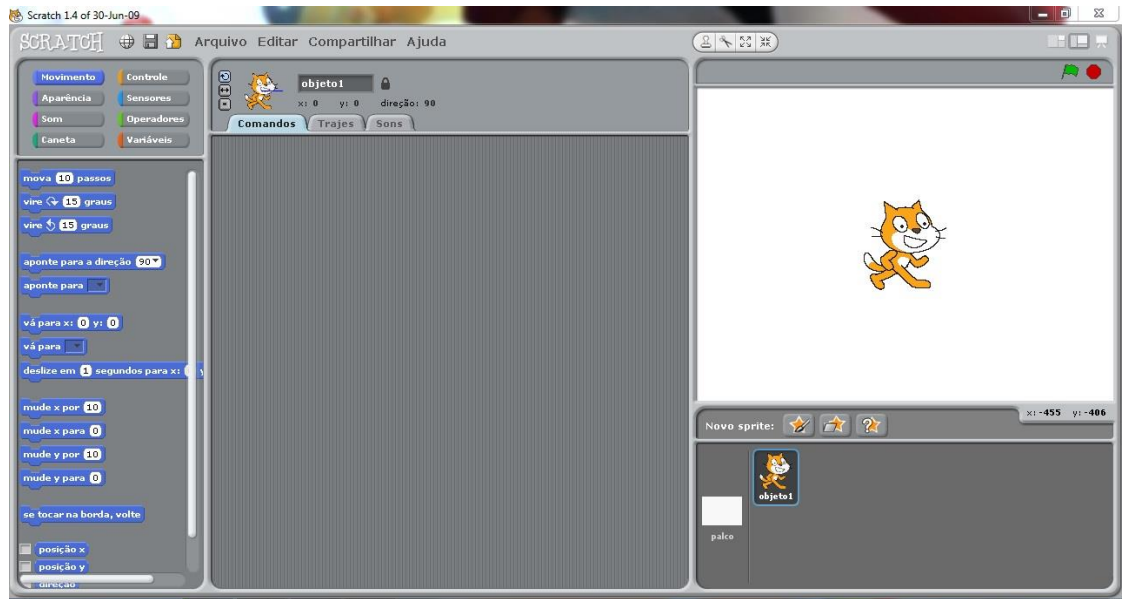
Ao trabalharem com os conceitos computacionais via *Scratch*, os usuários estão aprendendo a programar e dessa forma vão aprimorando o pensamento sistemático, o raciocínio lógico e demais habilidades que, ao se somarem, lhes trarão uma melhor compreensão do pensamento computacional. Com tudo isso o pensamento criativo é despertado e os aprendizes podem abrir novos horizontes trazendo problemas do seu contexto

sociocultural para serem solucionados, ao contrário de se aterem exclusivamente aos problemas padrões sugeridos.

Na interface deste *software* educacional (S.E.) está localizada a área de sintaxe da linguagem de programação *Scratch*, que está dividida em blocos de comandos, classificados por contexto. Cada bloco de comando tem uma cor e os comandos de cada respectivo bloco possuem a mesma cor deste, visando facilitar a assimilação dos comandos ao contexto nos quais se encontram inseridos. Ao lado da área destinada à sintaxe da linguagem, temos a área de desenvolvimento. Ela consiste no espaço determinado para a criação dos jogos, programas, diálogos e animações. Os códigos são desenvolvidos de maneira simples: basta escolher o comando que será usado por vez, em seu respectivo bloco de comando, e arrastá-lo até a área de desenvolvimento e ir encaixando um comando ao outro, de forma similar ao jogo de LEGO.

Por sua vez, paralelo a área destinada ao desenvolvimento, está localizado o palco, este se refere à área onde será exibida em tempo real a execução do algoritmo. Tendo em vista que o *Scratch* possui recursos multimídia, a execução dos algoritmos pode permitir movimentação dos *Sprites* (objetos da ferramenta que possuem as características de comandos, trajes e sons), reprodução de sons, gravações e demais efeitos, permitindo ao usuário ter uma real noção do que está sendo reproduzido pelo seu código. Isso facilita a percepção de possíveis *bugs*, que podem ser corrigidos. Embaixo do palco fica o espaço destinado para os *Sprites*, onde é possível visualizar todos os *Sprites*. Tais informações da interface podem ser visualizadas na Figura 1.

Figura 1 - Versão 1.4 do *Scratch* (off-line)



No exemplo da Figura 1 é possível ver os comandos do bloco de movimento. Na parte mais ao meio está localizada a área de desenvolvimento, exibindo os botões de comandos, onde são construídos os algoritmos, o botão de trajés, onde é feita a seleção dos trajés do *Sprite* selecionado, e o botão som, onde são importados ou gravados os sons que serão emitidos pelo *Sprite* selecionado.

3 PLANEJAMENTO E APLICAÇÃO DO CURSO

Esse capítulo apresenta a descrição das duas escolas nas quais os cursos foram aplicados (seção 3.1), aborda a interdisciplinaridades de *Scratch* com Arte e Cultura e Língua Tupi (seção 3.2), planejamento do curso (seção 3.3), e aplicação do curso (seção 3.4).

3.1 AS ESCOLAS

Escola Indígena de Ensino Fundamental e Médio Guilherme da Silveira, localizada na Aldeia Monte-Mór, no Município de Rio Tinto/PB, acolhe uma média de 413 alunos, dispostos nos níveis fundamental e médio, em turmas regulares ou de Educação de Jovens e Adultos. A escola disponibiliza aos alunos um laboratório de informática que possui quatro computadores, dos quais três são para uso dos alunos e um para uso administrativo. Todos possuem acesso à internet por meio de banda larga Qedu¹ (2014). Alguns dos alunos que participaram do curso fazem parte do 9º ano do Ensino Fundamental desta escola no ano de 2014.

A Escola Indígena de Ensino Fundamental e Médio Doutor José Lopes Ribeiro, situada na Vila Regina também no município Rio Tinto/PB, acolhe uma média de 507 alunos, dispostos também nos níveis de ensino Fundamental e Médio, e também disponibiliza ensino regular e Educação de Jovens e Adultos. A escola disponibiliza um laboratório de informática para uso de alunos, contendo cinco computadores: um para uso administrativo e os outros quatro para uso dos alunos. Todos possuem acesso à internet, porém sem banda larga Qedu² (2014). Alguns alunos da escola Doutor José Lopes também participaram do curso de *Scratch*, estes fazem parte do primeiro ano do Ensino Médio do ano de 2014.

3.2 INTERDISCIPLINARIDADE DE SCRATCH COM ARTE E CULTURA E LÍNGUA TUPI

Este trabalho busca constatar se o ensino de programação com uma abordagem construtivista e de forma interdisciplinar com as disciplinas que envolvem a cultura dos alunos permite uma maior assimilação dos conceitos introdutórios de programação e assim ajudam a despertar nos alunos o pensamento computacional. Tendo em vista que a programação é de muita valia para desenvolver habilidades cognitivas, como: raciocínio

lógico, criatividade e poder de abstração para solucionar problemas, essas habilidades serão úteis e poderão ser empregadas em diversas áreas de conhecimento.

O curso aqui aplicado, além de estimular tais habilidades, enfoca, de forma lúdica, o envolvimento dos alunos nas disciplinas de Arte e Cultura e Língua Tupi. Acredita-se que, uma vez que os alunos já conhecem parte dos conteúdos didáticos trabalhos nessas disciplinas, provenientes de um aprendizado cotidiano na sua aldeia, podem se sentir desmotivados a estudarem de forma tradicional, na sala de aula, conceitos que a priori já são conhecidos por eles.

Dessa forma, o contexto sociocultural no qual os estudantes estão inseridos foi explorado nas aulas para que eles possam, ao passo que aprendem a programar, conceber atividades que enfatizem a personalização. Assim eles têm experiências de atividades com um significado importante ao tempo em que também são reflexivas, e podem até trabalhar a colaboração. De fato, os alunos estão usando uma ferramenta interativa e que permite o aprendizado da programação visual e com uma abordagem baseada ao design que estimula por ser lúdica e ter um foco em algo significativo ao usuário, despertando com maior facilidade a assimilação do pensamento computacional.

3.3 PLANEJAMENTO DO CURSO

Como teoria da aprendizagem pensada para nortear a metodologia deste curso, foi escolhida a abordagem construtivista, que trabalha diretamente com ideias que já foram construídas anteriormente pelos indivíduos. Como os alunos com os quais trabalhamos ao longo deste curso são indígenas ou descendentes de indígenas, podemos concluir que muitas temáticas utilizadas nas atividades são de conhecimento dos alunos. Elas serviram para construção de atividades nas quais eles usaram a criatividade dentro do contexto da cultura indígena.

Uma das abordagens mais conhecidas e utilizadas no ambiente escolar é o construtivismo de Jean Piaget, como afirma Reategui et al. (2007). “Todo esquema de assimilação é construído e toda abordagem à realidade supõe um esquema de assimilação. Quando a mente assimila, ela incorpora a realidade a seus esquemas de ação, impondo-se ao

meio.” Como cita Ostermann e Cavalcanti (2010). Os mesmos autores dizem que: “Apesar de Piaget já propor estas ideias na década de 30 e ser um precursor da linha construtivista, sua teoria só conquistou um maior espaço na área educacional na década de 80, quando iniciou o declínio do comportamentalismo”.

No construtivismo, o homem constrói seu conhecimento em contato com o meio em que está inserido. Partindo do pressuposto que o conhecimento não é algo que está concluído, sendo então um processo de constante construção e desenvolvimento. Assim ele sugere a ideia de que o homem não nasce inteligente, mas também não é refém do meio, isto é, por meio das relações dele com o meio, reagindo às ações externas, ele pode construir e organizar seu próprio pensamento, de forma cada vez mais eficiente. Piaget, um dos teóricos mais defensores desta corrente, acreditava na habilidade da criança, em ter capacidade de absorver e trazer como herança tudo aquilo que lhe é exposto, de forma a transformá-la em conhecimento. Segundo Costa (2009), é necessário compreender que os estudantes levam ao convívio escolar, uma série de habilidades, hábitos e costumes, por meio dos seus valores, da sua cultura, de suas crenças, de suas atitudes e dos significados que trazem consigo.

A ideia central da preparação das atividades deste trabalho foi utilizar costumes rotineiros dos alunos de forma a parodiar com os novos ensinamentos ministrados no curso. Por exemplo, na primeira aula para ensinar o conceito de algoritmo foi feita uma parodia com o conceito de sequência, demonstrando que os passos utilizados para sair de casa na aldeia e chegar até a escola, poderiam facilmente ser transformados em um código. Mais adiante no curso, como no projeto final, tratou-se da construção de diálogos cuja temática esteja diretamente relacionada com a cultura indígena: lendas, costumes, músicas, rituais, danças e etc. Assim, os alunos tiveram que usar os conhecimentos aprendidos da computação criativa bem como os conceitos assimilados nas aulas anteriores para conceber o projeto, desenvolvido no *Scratch*, por meio da programação visual. Isso tudo partindo do pressuposto que os alunos já possuem conhecimento de sua cultura e de seus costumes rotineiros e que estão mais uma vez sendo reforçados nas disciplinas de Arte e Cultura e Língua Tupi.

O curso foi planejado com uma carga horária total de 16h, dividido em encontros semanais de 4h cada. Todas as aulas seguem o seu plano de aula apresentados no apêndice deste trabalho. Foram planejadas atividades teóricas e práticas para avaliar a aprendizagem dos alunos ao longo das aulas, além de um projeto final. As atividades desenvolvidas ao longo

das aulas tinham o objetivo de exercitar os conceitos ensinados nas respectivas aulas. O projeto proposto foi à construção de um diálogo no *Scratch* com uma temática indígena livre para todos os alunos, que pudessem abordar o uso de expressões da língua Tupi, bem como lendas, músicas, danças, entre outros temas do seu contexto sociocultural. O objetivo do projeto final consiste em fazer o aluno, utilizando os comandos aprendidos, explorar a programação visual e a computação criativa, enriquecendo as suas atividades com os efeitos multimídia por meio dos sons com a gravação de palavras ou expressões indígenas que eles queiram usar nos diálogos. Após a concepção, os alunos deveriam apresentar para seus colegas o projeto produzido. Dessa forma, eles trabalhariam o conceito de reflexão e também colaboração, uma vez que ao analisar a atividade do colega, poderiam sugerir reformulações.

3.4 APLICAÇÃO DO CURSO

Este curso trabalhou com práticas computacionais em todas as aulas seguindo uma hierarquia que visou facilitar a compreensão das mesmas, pois como relata Brennan e Resnick (2012) tais práticas se preocupam não apenas com o que será assimilado, mas em como se dará tal processo trabalhando assim o aprender bem como o pensar.

O curso teve início no dia 22 de novembro de 2014 e data de término o dia 13 de dezembro de 2014, totalizando quatro encontros consecutivos, aos sábados, das 13h00min às 17h00min. Houve a participação de trinta alunos indígenas, ou descendentes. Eles fazem parte das turmas do 1º ano do Ensino Médio e do 9º ano do Ensino Fundamental da Escola Estadual e Indígena José Lopes Ribeiro e Escola Indígena de Ensino Fundamental e Médio Guilherme da Silveira, respectivamente. Contou-se também com a presença do professor da disciplina de Arte e Cultura da Escola Estadual e Indígena José Lopes Ribeiro, que também é o mesmo professor da disciplina de Língua Tupi da Escola Indígena de Ensino Fundamental e Médio Guilherme da Silveira. Houve a presença do professor de informática da Escola Estadual Jaraguá, que apesar de não ministrar nenhuma disciplina nas escolas trabalhadas, soube do curso e pediu para participar das aulas.

A primeira aula (APÊNDICE A) foi iniciada com um questionário sociocultural que abordou perguntas relacionadas à cultura, conhecimentos prévios de computação, faixa etária, visando conhecer melhor a turma. Em seguida foram abordadas algumas teorias que embasam

os conteúdos introdutórios de programação. Assim, foi explicado o conceito de programação e a sua relação com os conteúdos que seriam abordados no curso e com a ferramenta *Scratch*. Para essas explicações foram feitas paródias com a rotina dos discentes, buscando facilitar a compreensão deles com os primeiros conceitos computacionais abordados. Por exemplo, para exemplificar o conceito de sequência foi feita a descrição de um algoritmo que exemplificava a rotina do caminho à escola. Para isso, os alunos disseram o passo a passo da rotina desde a saída de casa até a chegada na escola, e dessa forma, o algoritmo foi construído permitindo a discussão da ideia de sequência, bem como do conceito de colaboração, tendo em vista que todos os alunos auxiliaram na construção do algoritmo.

A segunda aula (APÊNDICE B) foi iniciada com a revisão dos conteúdos vistos na primeira aula e a execução da atividade que na aula anterior não foi possível ser realizada. Ela foi trabalhada como exercício de revisão para os alunos, sendo uma boa oportunidade para os que não presenciaram a aula anterior compreenderem melhor os conteúdos ministrados na mesma, e reforçar o conhecimento aprendido pelos presentes. Depois, foram abordados conceitos computacionais mais complexos em relação à aula anterior, todos foram trabalhados com a programação com *Scratch* e parodiados com o contexto sociocultural dos discentes.

As atividades da segunda aula, novamente na construção de algoritmos, voltaram-se para o desenvolvimento do conceito de **condição** com os **laços de decisão**, e também do conceito de dados por meio do uso de variáveis. Neste módulo também foi focado o conceito de evento, aproveitando um contato mais aprofundado nos blocos de comando movimento, controle e aparência presentes na ferramenta.

Na terceira aula (APÊNDICE C) o conceito de **evento** foi reforçado, uma vez que os discentes tiveram a oportunidade de se aprofundar no bloco de comando som. A computação criativa teve maior destaque nesse encontro tendo em vista que os alunos foram desafiados a realizar a atividade do “Seja criativo”. Essa atividade propõe a construção de um algoritmo de temática livre usando a prática de abstração, na qual, por meio da programação visual permitida pelo *Scratch*, eles trabalhariam as perspectivas computacionais se questionando ao pensarem em algo que quisessem expressar, e uma vez conectados, concebessem os seus algoritmos utilizando o maior número de comandos aprendidos, utilizando a prática de reuso. Nesse módulo também foi possível trabalhar as práticas computacionais de **teste e correção de erros**, pois os alunos tiveram como atividade analisar um código desenvolvido no *Scratch*

para identificar seus *bugs*, corrigindo-os e testando o código até que estivesse em pleno funcionamento.

A quarta aula (APÊNDICE D) foi iniciada com uma revisão dos conceitos introdutórios de programação e como eles se relacionam com os conceitos, as práticas e as perspectivas computacionais bem como com os comandos de cada bloco da ferramenta *Scratch*. Nesse módulo do curso, os alunos foram estimulados a desenvolver o projeto final do curso. Tratou-se de uma atividade interdisciplinar que buscou, mais uma vez, estimular o pensamento computacional. Os alunos, para desenvolvê-la, utilizaram todos os conteúdos aprendidos no curso, aplicando-os na construção de um diálogo com uma temática indígena livre.

Assim com o desenvolvimento das atividades ao longo do curso, os discentes foram aprendendo a formular questões, fazendo uso de conceitos e da ferramenta computacional *Scratch*, abstraindo ideias, seguindo modelos, manipulando dados, pensando e implementando algoritmos e a partir deles simulando eventos e combinando recursos testando e debugando erros, por meio de um processo iterativo e incremental. Dessa forma a soma de todas essas habilidades computacionais transmitidas no curso buscou explorar pela criação das atividades no S.E. *Scratch* o despertar bem como a disseminação do pensamento computacional com os alunos.

4 AVALIAÇÃO

Este capítulo apresenta o modelo de avaliação e a avaliação dos recursos produzido pelos alunos segundo o modelo apresentado.

4.1 MODELO PECT

O modelo de aprendizagem *Progressions of Early Computational Thinking* (PECT) segundo Seiter (2013) consiste em um modelo que visa mensurar as progressões do pensamento computacional via *Scratch* em estudantes do primeiro grau, por meio de uma sintetização que quantifica os trabalhos desenvolvidos pelos alunos em três níveis, sendo eles: 1 - básico, 2 - intermediário e 3 - profissional. Observando a utilização de padrões de projeto na codificação dos alunos, ou seja, à medida que os discentes fazem uso de elementos de programação que são comumente utilizados através de padrões de projeto, se torna possível fazer o mapeamento dos conceitos do pensamento computacional que estão sendo aplicados nos recursos produzidos.

Para avaliar as atividades do curso, foi utilizado um subconjunto do modelo PECT referente aos conceitos do pensamento computacional expressas nas variáveis de prova. As **variáveis de prova** são conjuntos de comandos do código escrito no *Scratch* que classificam em níveis de conhecimento o aprendizado dos conceitos do pensamento computacional. Eles são mensurados segundo comandos do *Scratch* que foram usados na construção das atividades dos alunos. Esses comandos classificam os três elementos nos níveis básico, intermediário e profissional.

Dessa forma, se o aluno em sua codificação atinge um conjunto de comandos referente a uma dada variável de prova, através do modelo, é possível classificar o seu nível de aprendizado na mesma, e assim atribuir uma medida sobre quanto o discente assimilou dos conceitos de pensamento computacional daquele item.

O subconjunto do modelo PECT utilizado contempla itens do pensamento computacional trabalhados durante o curso e está representado no Quadro 1, a seguir. Na primeira coluna observa-se os conceitos computacionais; na primeira linha, os níveis de

conhecimento adquirido em cada conceito; e as variáveis de prova são os conjuntos de comandos do *Scratch* que mensuram em que nível de conhecimento o aluno chegou.

Quadro 1 - Modelo PECT (Seiter, 2013).

	Básico	Intermediário	Profissional
Looks – bloco aparência.	Diga, pense.	Próxima fantasia, mostre, esconda.	Mudar para fantasia, Mudar a cor, tamanho etc.
Sons.	Reproduza o som, a nota.	Tocar X tocar até fazer.	
Movimento.	Mover, vá para o Sprite/ponto, vire.	Ir para x, y. Deslizar para x, y.	Set, a mudança X, Y.
Variáveis.	Inicie a variável (sprite, ponteiro do mouse, resposta, etc...).	Nova variável, <i>Set/change</i> .	Nova lista.
Sequencia e loop – bloco controle.	Sequencia.	Repita, sempre	Sempre se, repita até
Expressões booleanas - bloco operadores.	Operadores de detecção.	<, =, >.	<i>And, or, not</i> .
Operadores – bloco de operadores.	Matemáticos.	Randomizar.	Lista.
Condicionais – bloco de controle.	<i>If</i> .	<i>If... else</i> .	Junção do <i>if</i> com <i>if...else</i> .
Coordenadas – bloco de movimento.	Espere.	Quando eu receber, broadcast.	Espere ate.
Interface de usuário de evento – bloco de controle.	Quando bandeira verde clicado.	Quando chave pressionada, Quando objeto clicado.	Anuncie para todos e espere.
Paralelismo – bloco de controle.		Dois scripts iniciam com o mesmo evento.	
Inicializar localização – bloco de controle- bloco de movimento.		Quando bandeira verde clicado Defina propriedades de localização (x, y, etc.).	
Inicializar looks - bloco de controle/aparência.		Quando bandeira verde clicada mude as propriedades dos looks (traje, visibilidade, etc.).	

O Quadro 1 - Modelo PECT, orientou a avaliação das atividades dos alunos. Entretanto nem todos os critérios foram analisados, uma vez que não foram cobrados nas atividades desenvolvidas nesse trabalho. Dessa forma, foi então concebido o Quadro 2, o qual foi baseado no modelo PECT, porém se adaptando aos quesitos cobrados no curso ministrado. Foram retiradas unicamente as linhas referentes aos blocos variáveis e operadores. Nele é possível analisar todos os demais critérios que fazem parte do subconjunto do Modelo PECT exibido no Quadro 1 e que foram cobrados nas atividades desse trabalho, utilizando assim o mesmo padrão para mensurar o desenvolvimento dos alunos.

Quadro 2 - Modelo de avaliação desse trabalho adaptado de Seiter (2013).

	Básico	Intermediário	Profissional
Looks – bloco aparência.	Diga, pense.	Próxima fantasia, mostre, esconda.	Mudar para fantasia, mudar a cor, tamanho etc.
Sons.	Reproduza o som, a nota.	Tocar X tocar até fazer.	
Movimento.	Mover, vá para o Sprite/ponto, vire.	Ir para x, y. Deslizar para x, y.	Set, a mudança X, Y.
Sequencia e loop – bloco controle.	Sequencia.	Repita, sempre.	Sempre se, repita até.
Expressões booleanas - bloco operadores.	Operadores de detecção.	<, =, >.	And, or, not.
Condicionais – bloco de controle.	If.	If...else.	Junção do if com if...else.
Coordenadas – bloco de movimento.	Espere.	Quando eu receber, broadcast.	Espere ate.
Interface de usuário de evento – bloco de controle.	Quando bandeira verde clicado.	Quando chave pressionada, quando objeto clicado.	Anuncie para todos e espere.
Paralelismo – bloco de controle.		Dois scripts iniciam com o mesmo evento.	
Inicializar localização – bloco de controle/movimento.		Quando bandeira verde clicado, Defina propriedades de localização (x, y, etc.).	
Inicializar looks - bloco de controle/aparência		Quando bandeira verde clicada mude as propriedades dos looks (traje, visibilidade, etc.).	

Como esse curso foi norteado pela abordagem construtivista, os alunos foram avaliados em todas as aulas por participação e questionamentos. Todavia, a avaliação dos recursos produzidos pelos alunos, segundo o modelo, ocorreu em dois momentos: o primeiro ocorreu na terceira aula e o segundo ocorreu na quarta aula. Em ambos os dias, os alunos implementaram suas linhas de código sem seguir modelos pré-definidos. Dessa forma, foi possível analisar os dados obtidos sem correr o risco de haver manipulação.

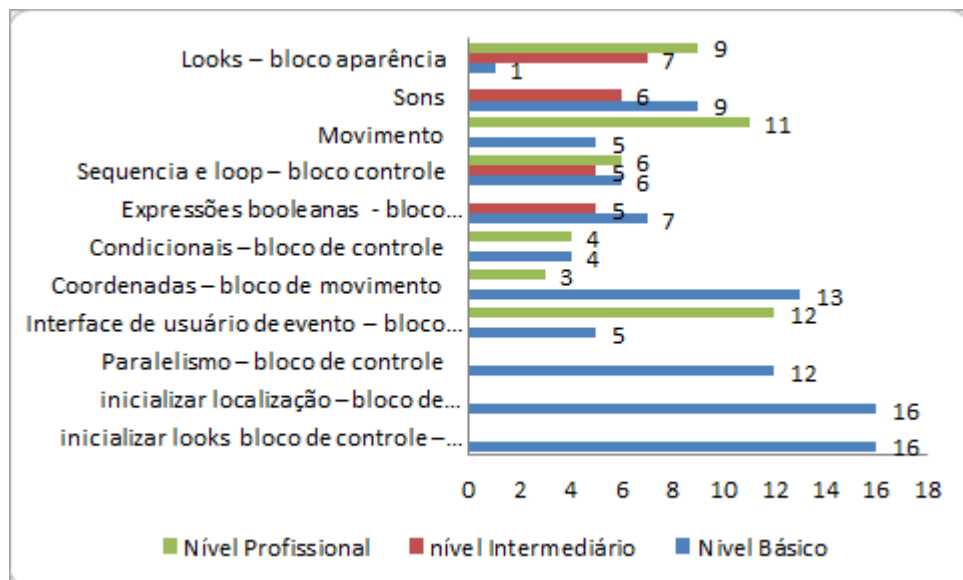
Na primeira atividade avaliada, sendo esta referente à terceira aula, todos os critérios do Quadro 2 receberam a classificação. Já na segunda atividade avaliada, sendo ela referente a quarta aula e projeto final, o único critério não avaliado foi o quesito **Sons**. Não foi possível avaliar esse quesito devido a problemas técnicos nos computadores durante a aula. Uma vez que foi trabalhada a abordagem construtivista, a avaliação foi realizada de forma hierárquica, para acompanhar o progresso dos alunos.

Para cada atividade realizada pelos alunos, foi gerada uma tabela avaliativa correspondente, que realizou a classificação de acordo com o modelo proposto. Após essa etapa, os resultados individuais de cada tabela foram condensados em uma tabela geral por aula e a partir dessas duas tabelas, foram concebidos os gráficos que serão exibidos em seguida no intuito de facilitar a visualização e discussão da avaliação geral do curso.

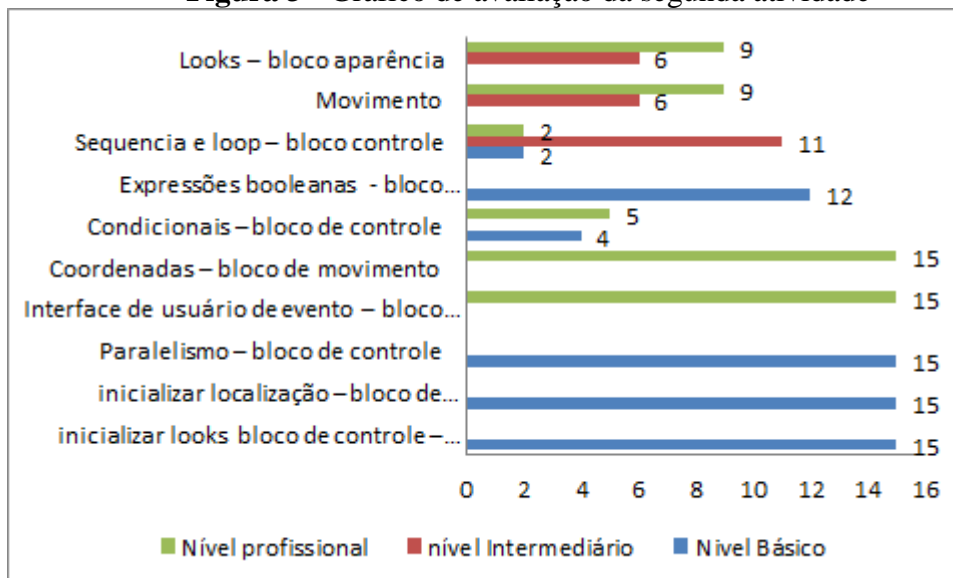
4.2 AVALIAÇÃO DAS ATIVIDADES

A Figura 2 apresenta o gráfico de avaliação do desempenho da turma durante a primeira atividade. O eixo horizontal do gráfico apresenta a quantidade de alunos que atingiram cada variável de projeto. Já o eixo vertical representa as variáveis de projeto, enquanto que as cores representam o nível básico, intermediário e profissional. Ao fim de cada barra de nível é apresentado o valor exato de alunos que atingiram o nível em questão da variável de projeto analisada na atividade. Essa atividade foi realizada por 17 alunos.

Figura 2 - Gráfico de avaliação da primeira atividade



Na Figura 3 é apresentado o gráfico de avaliação do desempenho da turma na segunda atividade. O gráfico segue o mesmo esquema do gráfico anterior. A diferença está na quantidade de alunos que realizaram essa atividade: total de 15 alunos.

Figura 3 - Gráfico de avaliação da segunda atividade

Os gráficos apresentam indícios de desenvolvimento do pensamento computacional, por meio dos critérios sugeridos pelo modelo PECT de avaliação. Os critérios **variáveis** e **operadores**, não foram cobrados em nenhuma das atividades, por isso não estão presentes no Quadro 02 e não foram avaliados nas atividades.

No gráfico referente à primeira atividade é possível analisar a presença de um critério a mais que o gráfico da segunda atividade, sendo ele **Sons**. Tal acontecimento se deu, pois durante a quarta aula, onde os alunos conceberam a segunda atividade, houve um problema técnico com a ferramenta, o que impossibilitou os discentes de fazerem uso desse quesito, sendo este retirado da avaliação da respectiva atividade (e não por incapacidade dos alunos de manipularem esse quesito).

Os três últimos critérios dos quadros, sendo eles referentes ao **paralelismo**, **iniciar localização** e **iniciar looks**, não são mensurados na tabela do modelo PECT como os demais critérios (em básico, intermediário e profissional). Os mesmos são verificados apenas como presentes ou ausentes nos códigos desenvolvidos pelos alunos. Assim, tais variáveis de projeto são padronizadas como nível satisfatório se estiverem presentes nos códigos, contemplando o valor de alunos que as utilizaram.

Apesar do paralelismo não possuir níveis como citado acima, existe um destaque para o uso desse critério observado na análise das atividades. Apesar dele ter alcançado um número favorável de discentes que atingiram seu conhecimento nas duas atividades analisadas,

observou-se que o **sincronismo**, não foi satisfatório, ou seja, nas atividades em que os alunos utilizaram o paralelismo para que dois eventos iniciassem juntos eles não obtiveram êxito na sincronização deles, e os mesmos não se encaixavam no tempo de execução um do outro, se levado em consideração toda a extensão das atividades, especialmente nas atividades referentes ao gráfico 2.

Ocorreu que, do início para o meio dessa atividade, o sincronismo foi usado corretamente, respeitando o tempo de diálogo entre os personagens. Entretanto, do meio para o final, o sincronismo não obteve a mesma qualidade do início, e a fala dos personagens do diálogo não respeitavam o tempo correto. Pode-se justificar esse acontecimento pelo fato do tempo destinado ao projeto final não ter sido suficiente para os alunos desenvolverem a atividade por completo com qualidade. Apesar disso, acredita-se que os alunos compreenderam e sabem utilizar o sincronismo, pois eles utilizam corretamente no início da atividade.

Na quarta aula, o número de alunos foi inferior ao da terceira aula. Isso justifica o comportamento de alguns critérios terem sofrido alteração, podendo em alguns casos ter sido até negativa. Por exemplo, se compararmos os critérios **expressões booleanas, iniciar localização e iniciar look**, nos gráficos, podemos observar que o número de alunos que utilizaram essas variáveis de projeto decresceu em algumas unidades, quando esperava-se que o número fosse aumentar. Entretanto, os critérios como **movimento, sequencia e loop, condicionais, coordenadas, interface e paralelismo** foram positivamente crescentes, o que nos permite a ideia de um progresso contínuo, como espera o construtivismo.

Um destaque que pode ser observado dá-se a **variável** referente ao bloco coordenadas, que inicialmente foi avaliada como nível básico em sua maioria, e posteriormente atingiu o nível profissional. Tal fato pode ser justificado, pois o conceito de coordenadas foi trabalhado desde o início do curso até seu final, sendo aplicado em exemplos, inclusive na aula 3 (referente ao gráfico 1) e como revisão na aula final (referente ao gráfico 2).

Geralmente a utilização dos comandos, referente aos operadores, esteve muito relacionado ao uso dos comandos dos critérios de aparência e interface. Uma vez que as atividades e exemplos utilizados ao longo do curso tiveram um foco voltado para a construção de histórias e animações, por meio de diálogos, já pensando em facilitar o desenvolvimento do projeto final, tendo em vista que este teve a mesma temática.

De um modo geral, pode-se analisar como favorável o crescimento dos níveis entre o gráfico 1 e o gráfico 2. Os discentes, ao final do curso, alcançaram em sua maioria os melhores níveis das variáveis presentes no Modelo PECT. Então, pode-se considerar como satisfatório, segundo o modelo, o aprendizado dos alunos em relação aos conceitos, práticas e perspectivas do pensamento computacional.

Por meio das atividades, observou-se a presença dos conceitos computacionais, baseado no modelo PECT de avaliação. A utilização desses conceitos pode sugerir o entendimento destes, no entanto, é necessário um maior tempo de acompanhamento, bem como análise de um maior número de atividades.

4.3 PERCEPÇÃO DO DOCENTE

Esta seção visou relatar a visão docente do curso, ou seja, pontos positivos e negativos, por meio de uma avaliação crítica do professor em relação ao desenvolvimento da turma durante todo o curso.

Foi possível perceber durante as primeiras aulas o interesse dos alunos em relação à ferramenta, uma vez que ela possui recursos visuais e multimídia. O que segundo eles facilitou o aprendizado dos conteúdos, desde os referentes aos conceitos, práticas e perspectivas do pensamento computacional, até mesmo a assimilação de conteúdos de matemática como, por exemplo, o entendimento do plano cartesiano, utilizado em algumas atividades do curso, onde os alunos relataram que foi mais fácil à compreensão do mesmo após a experiência prática de desenvolver animações que utilizavam tal conceito.

Ainda em relação ao fato da ferramenta ser lúdica, interativa e possuir recursos visuais e multimídia, os próprios alunos constataram que as cores utilizadas para separar os blocos de comando presentes no *Scratch*, facilitavam as construções dos algoritmos ao longo do curso. Logo eles perceberam que a divisão de um determinado problema em partes menores facilitaria a resolução do mesmo como um todo, o que eles até compararam com os problemas de física e matemática vistos na escola. E que de tal forma para construir um algoritmo era preciso utilizar a mesma técnica, tendo que vislumbrar as pequenas partes (os comandos) que seriam organizadas em uma ordem lógica concebendo ao seu final o algoritmo desejado.

Em relação ao software em si, os alunos disseram que o mesmo apresentava em suas funcionalidades, exatamente a ideia desejada, entretanto a explanação do que eram os conceitos, práticas e perspectivas computacionais foi de fundamental importância para chegarem ao entendimento do que era o pensamento computacional em si. Ou seja de forma autodidata eles até poderiam fazer uso da ferramenta, chegando a construir algoritmos, porém sem o conhecimento transmitido, não teriam compreendido que o desenvolvimento do pensamento computacional estava presente em suas construções, fazendo uso da ferramenta como um jogo digital que remete a ideia de um LEGO.

Um dos alunos, como já mencionado anteriormente, era professor de informática de uma das escolas participantes do curso. Ele relatou que em suas aulas ministra conteúdos relacionados a informática básica, ou internet. E que participar de tal curso foi de muita valia para o seu desenvolvimento enquanto docente, pois ele pretende retransmitir os conceitos, práticas e perspectivas computacionais, despertando em seus alunos o pensamento computacional, além de tentar fazer uso da ferramenta *Scratch* para trabalhar conteúdos de outras disciplinas, como português, podendo trabalhar conteúdos específicos com o uso de diálogos e na matemática com a construção de demais algoritmos, dessa forma estará obtendo em seu ponto de vista, uma melhora qualitativa em suas aulas, pois nelas “mostrará que a computação não se resume ao uso da informática básica, pesquisas no Google e utilização de redes sociais”.

A princípio os alunos de forma unânime constataram que acreditavam que aprenderiam conceitos relacionados à informática básica, como a utilização do conjunto de ferramentas do *Office*, ou um curso de montagem e manutenção de computadores, entretanto durante o curso relataram que foi uma surpresa grata, compreender que a computação é um universo ainda maior. Constatando que o curso transformou suas visões sobre a informática, tendo em vista que antes olhavam para ela apenas como uma fonte de recursos já existentes esperando serem utilizados (os programas) e agora já compreendem que ela lhes permite conceber a ferramenta que lhes é necessário. Assim passaram de usuários passivos para ativos da computação. Entendendo o pensamento computacional e sabendo utiliza-lo até mesmo em situações que não requerem a computação.

5 CONCLUSÃO

Estimular o pensamento computacional através de atividades interdisciplinares integrando programação, arte e cultura e língua Tupi com alunos do 9º ano do Ensino Fundamental e 1º ano do Ensino Médio, foi o maior objetivo desse trabalho. Essa meta foi alcançada por meio do desenvolvimento das habilidades pensamento computacional e os conceitos introdutórios de programação ministrados nesse curso utilizando *Scratch* como ferramenta.

O curso foi planejado para trabalhar habilidades básicas do pensamento computacional levando em consideração que os alunos possuíam um conhecimento mínimo sobre computação. Escolheu-se as disciplinas de Arte e Cultura e Língua Tupi pela possibilidade de criar interdisciplinaridade com a cultura da cidade natal onde o curso foi aplicado. Além disso, vislumbrou-se o potencial que essas disciplinas possuíam para criação de histórias e animações, nas quais eram possíveis trabalhar o pensamento computacional.

Atentando-se para a cultura dos alunos e o conteúdo das disciplinas selecionadas, as atividades ministradas pela professora e desenvolvidas pelos alunos no curso estavam referenciadas em seus costumes rotineiros. Em outras palavras, baseavam-se em conhecimentos prévios nativos que puderam ser expressados de outra forma, desenvolvendo novas habilidades intelectuais, por meio do *Scratch*. Nesse ponto, a abordagem construtivista guiou esse trabalho.

Dessa forma, os alunos, que já tinham o conhecimento de sua cultura, puderam aprender novos conhecimentos em relação à computação, através de uma ferramenta de fácil uso. Durante o curso buscou-se introduzir novos conceitos comparando-os com conhecimentos prévios. Assim, eles foram convidados a aprender a produzir atividades como as que lhes foram apresentadas de exemplo, e dessa forma, de posse de dos conhecimentos cultural e técnico, os discentes produziram atividades que estimularam o pensamento computacional e que favoreceram a interdisciplinaridade.

A avaliação deste trabalho contemplou as atividades desenvolvidas pelos alunos. Elas foram observadas buscando enxergar a presença da computação criativa, bem como as habilidades do pensamento computacional, expressadas e aplicadas pelos conceitos computacionais (paralelismo, sequencia, operadores, loops, eventos, condições) presentes nas estruturas/comandos da programação com *Scratch*. Ademais, a concepção dessas atividades

retratam também práticas computacionais (como reuso, abstração, *debug*, teste, concepção incremental) e outras perspectivas computacionais, tendo em vista que os estudantes se questionaram sobre o que poderiam conceber e como expressariam tudo que desejavam.

Para avaliação das atividades buscou-se um modelo adequado na literatura e aplicou-se esse modelo para cada atividade. O resultado dessa avaliação mostrou que os alunos atingiram um nível satisfatório de desempenho, sendo a maioria classificados nos níveis intermediário e profissional. Assim, há indícios que os alunos conseguiram assimilar tais conceitos fazendo uso das práticas computacionais já citadas e atingindo as perspectivas computacionais, segundo o modelo adotado. Os alunos também conceberam algoritmos, favorecendo a ideia de que todos conseguiram atingir o objetivo desse curso, conhecendo e entendendo o pensamento computacional.

REFERÊNCIAS

- ARAÚJO, A. L. S. O; SCAICO, P. D; PAIVA, L. F; RABÊLO, H. de M; SANTOS, L. de L; PESSOA, F. I. R; TARGINO, J. M; COSTA, L. dos S. (2013) Aplicação da Taxonomia de Bloom no ensino de programação com Scratch. In: XIX Workshop de Informática na Escola, Campinas – São Paulo.
- BLIKSTEIN, P. (2008). O pensamento computacional e a reinvenção do computador na educação. Disponível em <http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html> Acesso em: 10 de dez. de 2014.
- BRENNAN, K. (2011). “Creative computing: A design-based introduction to computational thinking”. ScratchEd.
- BRENNAN, K; RESNICK, M. (2012) “New frameworks for studying and assessing the development of computational thinking”. In: AERA 2012
- CAMBRAIA, A. C; OLIVEIRA, M. A. F. (2012) Aprender a programar: brincadeira ou bixo-papão. In: I Seminário Nacional de Inclusão Digital, Passo Fundo – Rio Grande do Sul.
- COSTA, T. A importância da gestão democrática no fortalecimento da qualidade de ensino. (2009). 39 f. Dissertação – Universidade Candido Mendes. Rio de Janeiro - 2009.
- Lifelong Kindergarten (2007). Disponível em: <http://ilk.media.mit.edu/projects.php?id=783>. Acessado em: 07 dez 2014.
- MARJI, M. Aprenda a programar com Scratch. [S.l.]: novatec, 2014.
- NUNES, D. J. (2008). “Licenciatura em Computação”. Jornal da Ciência, 30 de Maio.
- OSTERMANN, F; CAVALCANTI, C.J.H. Teorias de Aprendizagem. UFRGS – INSTITUTO DE FÍSICA. 2010.
- PEREIRA, P. de S.; MEDEIROS, M.; MENEZES, J. W. M. (2012) Análise do Scratch como Ferramenta de Auxílio ao Ensino de Programação de Computadores. In: XL Congresso Brasileiro de Educação em Engenharia, Belém - Pará.
- Qedu. (2013) Disponível em: <<http://www.qedu.org.br/escola/82156-eeiefm-guilherme-da-silveira/sobre>>¹. Acessado em: 07 Dezembro 2014.
- Qedu. Disponível em: <<http://www.qedu.org.br/escola/76049-eeiefm-dro-jose-lobes-ribeiro/sobre>>². Acessado em: 07 Dezembro 2014.
- REATEGUI, E; BOFF, B; FINCO, M. D. Proposta de Diretrizes para Avaliação de Objetos de Aprendizagem Considerando Aspectos Pedagógicos e Técnicos. WEI. 2007.
- RESNICK, M; MALONEY, J; MONROY-HERNÁNDEZ, A; RUSK, N; EASTMOND, E; BRENNAN, K; MILLNER, A; ROSENBAUM, E; SILVER, J; SILVERMAN, B; KAFAI, Y. 2011. “Scratch: Programming for all”. Comm. ACM 52, 11, 60–67.

Scratch (2011). Disponível em: <<http://scratch.mit.edu/about>>. Acessado em: 07 dez 2014.

SEITER, Linda; FOREMAN, Brendan. Modeling the learning progressions of computational thinking of primary grade students. In: Proceedings of the ninth annual international ACM conference on International computing education research. ACM, 2013. p. 59-66. APA

WING J. Computational Thinking. In: Communications of the ACM. v. 49. Março, 2006. p. 33-35.

WING J. Computational thinking and thinking about computing. In: Philosophical Transactions of the Royal Society, 2008. p. 3717–3725

APÊNDICE A

PLANO DE AULA 01

<u>DADOS DE IDENTIFICAÇÃO</u>	
Primeira Aula: motivação e introdução ao ambiente <i>Scratch</i>	
Professor (a): Hévellyn de Moraes Rabelo	E-mail: hevellyn.morais@dce.ufpb.br
Auxiliares: Wênio Kelson Francisco de Oliveira Luan de Luna Santos	E-mail: wennio.kelson@dce.ufpb.br luan.luna@dce.ufpb.br
Número de auxiliares: 02	Número de participantes: 30 alunos
Carga Horária: 4 horas	Data: 22/11/2014

<u>OBJETIVO GERAL DA AULA</u>
Apresentar aos alunos o ambiente de desenvolvimento e a linguagem de programação <i>Scratch</i> . Introduzir conceitos iniciais de algoritmos e laço de repetição. Desenvolver atividades práticas com os alunos.

<u>CONTEÚDO PROGRAMÁTICO</u>
<ul style="list-style-type: none">▪ Conteúdo 1: Conceito de algoritmo e linguagem de programação▪ Conteúdo 2: Familiarização do ambiente do <i>Scratch</i>▪ Conteúdo 3: Plano cartesiano▪ Conteúdo 4: Laços de repetição

<u>ROTEIRO DA AULA</u>	
Número - AULA : 01	Duração da aula: 4 horas
Objetivo da Aula: <ul style="list-style-type: none">• Motivar• Apresentar o conceito de criação no contexto do Scratch;• Capacitar para imaginar possibilidades de criação no ambiente Scratch;• Familiarizar-se com os recursos do ambiente;• Apresentar o site do Scratch;• Apresentar os elementos e comandos mais importantes para a construção de um projeto no Scratch;• Compartilhar projeto produzido no Scratch;	
Passo a Passo:	Tempo
5. QUIZ – Inicial - Este QUIZ inicial tem o intuito de coletar as características socioculturais dos alunos.	(10 minutos)

<p>6. Motivação</p> <ul style="list-style-type: none"> - Apresentação - O que é ser programador; - Onde eles podem atuar; - Exemplos de sucesso da programação; 	<p>(15 minutos)</p>
<p>7. Conceituar e exemplificar Algoritmo</p> <ul style="list-style-type: none"> - Pedir que um deles explique uma cerimônia indígena; - Definição do que é algoritmo; - Exemplo de algoritmo: ir à escola; - Explicar a relação entre algoritmo e a computação. 	<p>(15 minutos)</p>
<p>8. Conceituar e exemplificar Linguagem de Programação</p> <ul style="list-style-type: none"> - Definição de Programação; - Exemplos do significado de linguagem; - Definição de linguagem de programação. 	<p>(10 minutos)</p>
<p>9. Relação de Linguagem de Programação com o Scratch</p> <ul style="list-style-type: none"> - Definição da linguagem <i>Scratch</i>; - Relacionar o <i>Scratch</i> com o jogo LEGO. 	<p>(10 minutos)</p>
<p>10. Familiarização com o ambiente Scratch</p> <ul style="list-style-type: none"> - Tela inicial do <i>Scratch</i>; - Nomes dos campos; - Categoria de comandos; - Abas para opções comandos, trajes e sons; - Área de edição (depende da aba selecionada no comando anterior); - Botões para editar o objeto no palco: duplicar, apagar, crescer ou encolher objeto; - Botão para alterar tamanho de visualização do palco; - Botões para iniciar (bandeira verde) e parar o script (círculo vermelho); - Palco: local de visualização do Sprite; - Área que disponibiliza a seleção/criação dos <i>Sprites</i> e do palco. - Explicar como funciona o Sprite. 	<p>(15 minutos)</p>
<p>11. Entender o sistema de coordenadas</p> <ul style="list-style-type: none"> - Definição de coordenadas; - Exemplos de localização de um ponto na tela; - Exemplos de aplicação; - Como funciona o plano cartesiano no palco do <i>Scratch</i>. 	<p>(15 minutos)</p>
<p>12. Entender os laços de repetição</p> <ul style="list-style-type: none"> - Definição de laços de repetição - Exemplos usando laço de repetição - Mostrar exemplos de laço de repetição 	<p>(30 minutos)</p>
<p>13. Prática 01 – Movimentação através de coordenadas</p>	<p>(20 minutos)</p>
<p>14. Prática 02 – Entendendo <i>Sprites</i> com seu nome</p>	<p>(20 minutos)</p>
<p>15. Exercício em sala – Key Move</p>	<p>(30 minutos)</p>

16. Criar conta no site do <i>Scratch</i> e compartilhar as animações feitas em sala - Abrir o site; - Fazer o cadastro; - Mostra como compartilhar a prática 02 e o exercício de sala.	(30 minutos)
TOTAL EM MINUTOS	240 minutos

Softwares utilizados
Nome do Software: Microsoft <i>PowerPoint</i>
Versão do Software: Microsoft Office Professional Plus 2010
Nome do Software: <i>Scratch</i>
Versão do Software: <i>Scratch 1.4</i>

Prática Utilizada:
Nome da Prática: Movimentação através de coordenadas
Objetivos da Prática: Entender o problema a ser resolvido. Implementar comandos no <i>Scratch</i> que maneira que o gato (<i>sprite</i>) pule em 4 pontos específicos no palco.
Passo a passo: - Comece selecionando o bloco de Controle e arraste o comando [<i>quando bandeira clicado</i>] para à área de comando; - Arraste o comando [<i>repita</i>] para à área de comando colocando-o abaixo do comando [<i>quando bandeira clicado</i>]; - Arraste o comando [<i>espere x segundos</i>] para dentro do comando [<i>repita</i>], isto deve ser feito quantas vezes o <i>sprite</i> for mudar de posição; - Arraste o comando [<i>vá para x: y:</i>] para dentro do comando [<i>repita</i>] abaixo do comando anterior. Isso deve ser feito quantas vezes o <i>sprite</i> for mudar de posição;
Prática Utilizada:
Nome da Prática: <i>Entendendo Sprites</i>
Objetivos da Prática: Um estudo aprofundado dos <i>Sprites</i> analisando suas funcionalidades, como por exemplo, a sua aparência, os seus botões de uso, a sua lista de <i>Sprite</i> e o estilo de rotação.
Passo a passo: - Escolha os <i>Sprites</i> referentes às letras do seu nome no banco de <i>Sprites</i> ; - Para cada <i>Sprite</i> da letra: - Selecione o bloco de Controle arraste o comando [<i>quando bandeira clicado</i>] para à área de comando; - Arraste o comando [<i>sempre</i>] para baixo do comando [<i>quando bandeira clicado</i>]; - Selecione o bloco de Aparência ou movimento e arraste uma ação para dentro do comando [<i>repita</i>] para dar o efeito desejado ao <i>Sprite</i> .
Exercício de Sala:
Nome do Exercício: <i>Key Move</i>
Objetivos do Exercício: - Praticar os comandos dos elementos Movimento, Controle e Aparência;

- Mostrar que os projetos em *Scratch* constroem-se juntando comandos, assim como se constroem diferentes objetos no mundo físico com peças de um quebra-cabeça.

- Construir o algoritmo que faz o gato dançar.

Passo a passo:

- Comece por arrastar o comando **move 10 passos** da categoria Movimentos para a área de comandos. Cada vez que se clicar no comando, o gato anda 10 passos. Pode-se mudar o número de passos para fazer o gato andar distâncias maiores ou menores.

- Arraste o comando **toca tambor** da galeria Sons. Clique no comando xxx para ouvir o som do tambor. Junte este comando abaixo do comando **move**. Quando clicar neste bloco, o gato move-se e o tambor toca.

- Copie este bloco de comandos (ou usando o item **Duplicar** da barra de ferramentas ou clicando no bloco com o botão direito do rato e selecionando duplicar) e junte a cópia aos comandos já colocados.

- Alterar para **-10 passos** o valor do segundo move. Cada vez que clicar no bloco de quatro comandos, o gato fará uma pequena dança para frente e para trás.

- Arraste o comando **repete 10 vezes** da categoria Controle de forma a envolver os outros comandos, na respectiva área. Quando clicar no bloco, o gato dança 10 vezes para frente e para trás.

- Finalmente, arraste o comando **quando clicar** no Sprite da galeria Controle para o topo do bloco de comandos. Clique no gato (em vez de clicar no bloco de comandos), para fazer o gato dançar.

RECURSOS MATERIAIS

- Data Show;
- Caneta (30 unidades);
- Folha A4;
- Pincel para Quadro Branco;
- Apagador de quadro branco;

BIBLIOGRAFIA

MARJI, M. Aprenda a programar com Scratch. [S.l.]: novatec, 2014.

APÊNDICE B

PLANO DE AULA 02

<u>DADOS DE IDENTIFICAÇÃO</u>	
Segunda Aula: laço de repetição e decisão	
Professor (a): Hévellyn de Moraes Rabelo	E-mail: hevellyn.morais@dce.ufpb.br
Auxiliares: Wênnio Kelson Francisco de Oliveira Luan de Luna Santos	E-mail: wennio.kelson@dce.ufpb.br luan.luna@dce.ufpb.br
Número de auxiliares: 02	Número de participantes: 30 alunos
Carga Horária: 4 horas	Data: 29/11/2014

<u>OBJETIVO GERAL DA AULA</u>
Após familiarizados com a Linguagem <i>Scratch</i> , esta aula servirá para o aprofundamento nos Blocos de Comando – movimento, controle e aparência. A partir disso será feita uma introdução a lógica de programação utilizando exemplos de laço de repetição e decisão contidos nas práticas desenvolvidas junto com os alunos.

<u>CONTEÚDO PROGRAMÁTICO</u>
17. Conteúdo 1: Revisão da primeira aula 18. Conteúdo 2: Laços de decisão 19. Conteúdo 3: Variáveis

<u>ROTEIRO DA AULA</u>	
Número - AULA : 02	Duração da aula: 4 horas
Objetivo da Aula: <ul style="list-style-type: none">▪ Revisar conteúdos vistos na aula passada;▪ Familiarizar a turma com mais efeitos/animações que o <i>Scratch</i> oferece;▪ Fazer um aprofundamento do estudo dos elementos: movimento, controle e aparência;▪ Desenvolver práticas de pequenos projetos para diferenciar os efeitos existentes e utilizar variáveis e laços de decisão e repetição.▪ Criar um banco de dados em áudio com palavras ou expressões indígenas para utilizar nas atividades das próximas aulas.	
Passo a Passo: <ul style="list-style-type: none">• Revisão da aula passada<ul style="list-style-type: none">- Relembrar a definição de coordenadas;- Relembrar laços de repetição;- Utilizar alguns exemplos de localização de um ponto na tela;	(20 minutos)

<ul style="list-style-type: none"> • Explicar o Bloco de Comandos – Movimento <ul style="list-style-type: none"> - Exibir no Data show a tela com Bloco de Comando - movimento - Detalhar cada item explicando para que servem. - Utilizar o exemplo de deslocar o Gato ('sprite') na tela ('palco') 	(20 minutos)
<ul style="list-style-type: none"> • Explicar o Bloco de Comandos - Controle (15 minutos) <ul style="list-style-type: none"> - Exibir no data show a tela com bloco de comando - controle - Detalhar cada item explicando para que servem. 	(20 minutos)
<ul style="list-style-type: none"> • Explicar o Bloco Comandos – Aparência <ul style="list-style-type: none"> - Exibir no Data show a tela com Bloco de Comando - aparência - Detalhar cada item explicando para que servem. 	(10 minutos)
<ul style="list-style-type: none"> • Explicar laços de decisão <ul style="list-style-type: none"> - Explicar o que são laços de decisão e quando são usados no algoritmo. 	(20 minutos)
<ul style="list-style-type: none"> • Explicar variáveis <ul style="list-style-type: none"> - Explicar o que são variáveis e quando são usadas no algoritmo. 	(20 minutos)
<ul style="list-style-type: none"> • Relembrar o que são laços de repetição <ul style="list-style-type: none"> - Relembrar o conceito de laços de repetição e quando são usados de acordo com a atividade da aula anterior. 	(20 minutos)
<ul style="list-style-type: none"> • Prática 01 – Pegue o dinheiro 	(20 minutos)
<ul style="list-style-type: none"> • Prática 02 – Key Move 	(20 minutos)
<ul style="list-style-type: none"> • Acessar a sua conta no Scratch 	(10 minutos)
<ul style="list-style-type: none"> • Compartilhar Projeto 	(10 minutos)
<ul style="list-style-type: none"> • Criar o banco de dados (em áudio) <ul style="list-style-type: none"> - Gravar palavras e expressões indígenas sugeridas pelo professor de Arte e Cultura em formato mp3 (compatível com Scratch) e salvar em uma pasta para ser disponibilizado nas aulas seguintes para a criação do diálogo entre o personagem índio e o personagem não índio. 	(40 minutos)
<ul style="list-style-type: none"> • Responder o formulário 	(10 minutos)
TOTAL em minutos	240 minutos

Softwares utilizados
Nome do Software: Microsoft PowerPoint
Versão do Software: Microsoft Office Professional Plus 2010
Nome do Software: Scratch
Versão do Software: Scratch 1.4

Prática Utilizada:
Nome da Prática: Pegue o dinheiro
Objetivos da Prática: Desenvolver um jogo simples, no qual o jogador deve mover o Sprite utilizando as

teclas de direção para coletar o máximo de sacos de ouro possível. Cada saco aparecerá em lugares aleatórios e o jogador terá três segundos para pegar o saco, senão este será movido novamente para outro lugar. Com essa prática ele reforçou a noção de espaço, que geralmente é trabalhada nas matérias de matemática, pois para que o *Sprite* possa se movimentar dentro do palco, o aluno teria que ter noções de localização no plano cartesiano, e esse mesmo conceito iria ajudá-lo a limitar a área em que o objeto poderia aparecer.

Dinâmica passo a passo:

Crie o *Sprite jogador*

- Selecione o comando [quando bandeira clicado] do bloco de controle;
- Arraste o comando [vá para x: e y:] do bloco movimento, encaixe abaixo do comando anterior, e coloque os valores do par ordenado como (-30, -30);
- Ainda no bloco movimento, selecione e arraste o comando [aponte para] e selecione a direção 90;
- No bloco controle, selecione o comando [quando tecla pressionada] e selecione seta para esquerda. Não encaixe este comando abaixo dos demais;
- No bloco movimento, selecione o comando [aponte para a direção], selecione o valor -90 e encaixe abaixo do comando anterior;
- No bloco som, selecione o comando [toque o som] e encaixe no código anterior. Na aba som, importe o som *pop* da pasta *effects*;
- No bloco movimento, arraste e encaixe no comando anterior o comando [mova 60 passos] e em seguida, arraste o comando [se tocar na borda volte];

- Duplique este segundo bloco de código mais 3 vezes, e mude as direções de cada uma das teclas pressionadas para direita, baixo e cima;

- Mude os valores das direções respectivamente para: 90, 180, 0, respectivamente;

Crie o *Sprite dinheiro*

- Selecione no bloco de controle o comando [quando bandeira clicado];
- Selecione o bloco de variáveis, nele crie uma variável pontos e selecione o comando [mude pontos para 0] e encaixe abaixo do comando anterior;
- Selecione no bloco de controle o comando [repita e selecione 20];
- Selecione no bloco de movimento selecione o comando [mude x para] e coloque dentro do repita, vá até o bloco sensores e arraste a opção do [operador de soma], no primeiro espaço coloque o valor -210, no segundo valor preencha com o comando [sorteie um número entre], este comando terá mais dois valores para serem preenchidos, no primeiro preencha 0 e no segundo preencha com o comando de [multiplicação] e coloque os valores 7 e 60;
- Selecione no bloco de movimento selecione o comando [mude x para] e coloque dentro do repita embaixo do comando anterior, vá até o bloco sensores e arraste a opção do [operador de soma], no primeiro espaço coloque o valor -150, no segundo valor preencha com o comando [sorteie um número entre], este comando terá mais dois valores para serem preenchidos, no primeiro preencha 0 e no segundo preencha com o comando de [multiplicação] e coloque os valores 5 e 60;
- No bloco sensores selecione o comando [zere temporizador] e coloque dentro do repita abaixo do comando anterior;
- No bloco de controle, selecione o comando [espere até] e coloque dentro do repita abaixo do comando anterior, preencha seu valor com o comando [ou] do bloco de operadores, preencha o primeiro valor com o comando [tocando em jogador] do bloco de sensores, o segundo valor preencha com o comando [menor que] do bloco de operadores, o primeiro valor preencha com temporizador e o segundo com 3;

- No bloco de controle selecione o comando [se então] e coloque dentro do repita abaixo do comando anterior, preencha seu valor com o comando [tocando em jogador] do bloco de sensores;
- Dentro do comando [se então] anterior, coloque o comando [toque o som] do bloco som e na aba sons, importe o som [WaterDrop] da pasta effects e selecione este som no comando em questão;
- Abaixo do comando anterior e dentro do comando [se então] coloque o comando [mude pontos por 1] do bloco de variáveis.

Prática Utilizada:

Nome da Prática: *Key Move*

- | Objetivos | do | Exercício: |
|---|-----------|-------------------|
| <ul style="list-style-type: none"> - Um estudo aprofundado dos comandos dos elementos Movimento, Controle e Aparência; - Os projetos constroem-se juntando comandos, assim como se constroem diferentes objetos no mundo físico com peças de um quebra-cabeça. - Apenas como exemplo, vamos fazer o gato dançar. | | |

Passo a passo:

- Comece por arrastar o comando move 10 passos da categoria Movimentos para a área de comandos. De cada vez que se clica no comando, o gato anda 10 passos. Pode mudar este número para fazer andar o gato distâncias maiores ou menores.
- Arraste o comando toca tambor da galeria Sons. Clique no comando para ouvir o som do tambor. Junte este comando abaixo do comando move. Quando clica neste bloco de dois comandos, o gato move-se e o tambor toca.
- Copie este bloco de comandos (ou usando o item Duplicar da barra de ferramentas ou clicando no bloco com o botão direito do rato e selecionando duplicar) e junte a cópia aos comandos já colocados.
- Alterar para -10 passos o valor do segundo move. De cada vez que clicar o bloco de quatro comandos, o gato fará uma pequena dança para frente e para trás.
- Arraste o comando repete 10 vezes da categoria Controle por forma a envolver os outros comandos, na respectiva área. Quando clicar no bloco, o gato dança 10 vezes para frente e para trás.
- Finalmente, arraste o comando quando clicar no Sprite da galeria Controle para o topo do bloco de comandos. Clique no gato (em vez de clicar no bloco de comandos), para fazer o gato dançar.

Exercício 01 de Sala Utilizado:

Nome da Prática: *Seja criativo*

Objetivos da Prática:

Fazer eles pensarem em uma lenda, dança, música, (temática livre indígena) para criar o projeto final.

Dinâmica passo a passo:

- Pensar nas frases, palavras ou expressões indígenas que vão usar;
- Gerar um áudio para estas palavras, expressões;
- Anotar estas ideias na folha do seja criativo.

RECURSOS MATERIAIS

- Data show
- Material de apoio (apostila)
- Computador com acesso à internet.

BIBLIOGRAFIA

MARJI, M. Aprenda a programar com Scratch. [S.l.]: novatec, 2014.

APÊNDICE C

PLANO DE AULA 03

<u>DADOS DE IDENTIFICAÇÃO</u>	
Terceira Aula: Construindo diálogos com <i>Scratch</i>	
Professor (a): Hévellyn de Morais Rabêlo	E-mail: hevellyn.morais@dce.ufpb.br
Auxiliares: Wênnio Kelson Francisco de Oliveira Luan de Luna Santos	E-mail: wennio.kelson@dce.ufpb.br luan.luna@dce.ufpb.br
Número de auxiliares: 02	Número de participantes: 30 alunos
Carga Horária: 4 horas	Data: 06/12/2014

<u>OBJETIVO GERAL DA AULA</u>
Está aula servirá para o aprofundamento nos Blocos de Comando – som, aparência, controle, movimento. A partir disso será feito prática, onde é feito um projeto junto com os alunos e em seguida um exercício para reforçar o que foi aprendido.

<u>CONTEÚDO PROGRAMÁTICO</u>
<ul style="list-style-type: none">▪ Conteúdo 1: Revisão dos comandos: aparência, movimento e controle▪ Conteúdo 2: Bloco de Comando som

<u>ROTEIRO DA AULA</u>	
Número - AULA : 03	Duração da aula: 4 horas
Objetivo da Aula: <ul style="list-style-type: none">• Concluir e revisar conteúdos vistos na aula passada (Bloco de comandos Movimento, Controle e Aparência);• Familiarizar a turma com mais efeitos do <i>Scratch</i>;• Fazer um aprofundamento do estudo dos elementos: Som;• Desenvolver práticas de pequenos projetos para diferenciar os efeitos existentes.	
Passo a Passo:	
1. Revisão da aula passada - Movimento, controle, aparência;	(30 minutos)
2. Entender o Bloco de Comandos – Som - Exibir no Data show a tela com Bloco de Comando - som - Detalhar cada item explicando para que servem.	(30 minutos)
3. Entender o Bloco de Comandos – Caneta - Exibir no Data show a tela com Bloco de Comando - caneta - Detalhar cada item explicando para que servem.	(30 minutos)
4. Prática 01 – Diálogo entre dois Sprites	(40)

	<i>minutos)</i>
5. Exercício 01 - Animação básica corrigir o código - Os alunos devem encontrar os erros da animação, detectar qual parte do código se encontra com bugs e corrigir os mesmos.	<i>(30 minutos)</i>
6. Exercício 02 – Seja Criativo - Os alunos vão começar a descrever como farão o projeto na aula seguinte baseado no exemplo feito na aula atual (diálogo entre dois <i>sprites</i>);	<i>(30 minutos)</i>
7. Formulário Aula 03	<i>(30 minutos)</i>
TOTAL em minutos	240 minutos

Softwares utilizados
Nome do Software: Microsoft <i>PowerPoint</i>
Versão do Software: Microsoft Office Professional Plus 2010
Nome do Software: <i>Sracth</i>
Versão do Software: <i>Scracth 1.4</i>

Prática Utilizada:
Nome da Prática: Diálogo entre dois <i>Sprites</i>
Objetivos da Prática: Exercitar o comando som, movimento, aparência, caneta, controle;
Dinâmica passo a passo: - Primeiro selecione o fundo de tela (<i>school1</i>) para o palco; - Adicione o <i>sprite</i> (professora); - Adicione o <i>sprite</i> (aluno) e seus dois trajes; - Para o <i>Sprite</i> (professora) faça: - Selecione o comando “quando bandeira clicado” do bloco de controle e arraste até a área de comando; - No bloco de aparência selecione o comando “mude o tamanho para: %” arraste e encaixe abaixo do comando anterior, preencha o espaço com o valor 150; - No bloco de controle selecione o comando “espere: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com o valor 1; - No bloco de aparência selecione o comando “diga por: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com os respectivos valores: “Boa tarde Joãozinho!!!” e “2”; - No bloco de controle selecione o comando “espere: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com o valor 3; - No bloco de aparência selecione o comando “diga por: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com os respectivos valores: “O que nós aprendemos hoje?” e “3”; - No bloco de controle selecione o comando “espere: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com o valor 4; - No bloco de aparência selecione o comando “diga por: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com os respectivos valores: “Muito bem Joãozinho!” e “2”;

- Para o *sprite*(aluno) faça:

- Selecione o comando “quando bandeira clicado” do bloco de controle e arraste até a área de comando;
- No bloco de aparência selecione o comando “mude para o traje:” arraste e encaixe abaixo do comando anterior e escolha o traje1;
- No bloco de controle selecione o comando “espere: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com o valor 4;
- No bloco de aparência selecione o comando “diga por: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com os respectivos valores: “Boa tarde querida professora!” e “2”;
- No bloco de controle selecione o comando “espere: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com o valor 0.5;
- No bloco de aparência selecione o comando “mude para o traje:” arraste e encaixe abaixo do comando anterior e escolha o traje2;
- No bloco de controle selecione o comando “espere: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com o valor 0.5;
- No bloco de aparência selecione o comando “mude para o traje:” arraste e encaixe abaixo do comando anterior e escolha o traje1;
- No bloco de controle selecione o comando “espere: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com o valor 3;
- No bloco de aparência selecione o comando “diga por: segundos” arraste e encaixe abaixo do comando anterior, preencha o espaço com os respectivos valores: “Aprendemos a fazer diálogos professora!” e “2”;
- No bloco de aparência selecione o comando “mude o tamanho para: %” arraste e encaixe abaixo do comando anterior, preencha o espaço com o valor 70;

Exercício 01 de Sala Utilizado:

Nome do exercício: Animação básica corrigir o código

Objetivos da Prática:

Trabalhar a percepção dos alunos de modo que eles possam encontrar os erros do exercício e modificar ou acrescentar os códigos necessários para que a animação possa ser executada sem erros.

Passo a passo:

- Encontrar o erro que faz o (*sprite*) bat, voar de cabeça para baixo, modificando o que for preciso para que ele voe sempre de cabeça pra cima.
- Encontrar o erro que faz o (*sprite*) objeto1 bater na tela e não voltar, modificando o código de forma que o (*sprite*) fique se movimentando sempre dentro do palco.

Exercício 02 de Sala Utilizado:

Nome do exercício: Seja Criativo

Objetivos da Prática:

Fazer com que os alunos comecem a pensar em termos de algoritmo e estruturar as ideias do seja criativo da aula anterior em códigos com *Scratch* para executar na aula seguinte como projeto final.

Passo a passo:

- Planeje seu algoritmo (Pense no que você deseja criar no *Scratch*);
- Utilize todos os blocos de comandos vistos até agora em sala de aula (Movimento, Controle, Aparência, Som, Caneta e variáveis);
- Explore bem cada bloco de comando, pois quanto mais comandos você utilizar melhor será o resultado final do seu projeto;
- Seja criativo, use e abuse da imaginação.

RECURSOS MATERIAIS

- Data show
- Material de apoio (apostila)
- Computador com acesso à internet.

BIBLIOGRAFIA

MARJI, M. Aprenda a programar com Scratch. [S.l.]: novatec, 2014.

APÊNDICE D

PLANO DE AULA 04

<u>DADOS DE IDENTIFICAÇÃO</u>	
Terceira Aula: Construindo diálogos com <i>Scratch</i> projeto final	
Professor (a): Hévellyn de Moraes Rabêlo	E-mail: hevellyn.morais@dce.ufpb.br
Auxiliares: Wênnio Kelson Francisco de Oliveira Luan de Luna Santos	E-mail: wennio.kelson@dce.ufpb.br luan.luna@dce.ufpb.br
Número de auxiliares: 02	Número de participantes: 30 alunos
Carga Horária: 4 horas	Data: 13/12/2014

<u>OBJETIVO GERAL DA AULA</u>
Está aula servirá para revisar de maneira geral os conteúdos vistos ao longo do curso, bem como desenvolver a atividade deste módulo, sendo ela o projeto final.

<u>CONTEÚDO PROGRAMÁTICO</u>
<ul style="list-style-type: none">▪ Conteúdo 1: Revisão dos blocos de comandos: aparência, movimento, controle, variáveis, caneta, operadores, sensores e som.▪ Conteúdo 2: Revisão dos conceitos computacionais: sequencia, <i>loops</i>, paralelismo, eventos, condições, operadores e dados.▪ Conteúdo 3: Revisão das práticas condicionais: atividades interativas e incrementais, testes e correções de erros, reuso, modularização e abstração.▪ Conteúdo 4: Revisão das perspectivas computacionais: questionar, conectar e expressar.

<u>ROTEIRO DA AULA</u>	
Número - AULA : 03	Duração da aula: 4 horas
Objetivo da Aula: <ul style="list-style-type: none">• Revisar os conteúdos vistos no curso;• Desenvolver o projeto final utilizando a ferramenta Scratch e uma temática indígena livre;• Responder ao formulário final.	
Passo a Passo:	
8. Revisão dos blocos de comando <ul style="list-style-type: none">- Movimento, controle, aparência, variáveis, caneta, operadores, sensores e som;	(30 minutos)
9. Entender os conceitos computacionais <ul style="list-style-type: none">- Explicar o que os conceitos computacionais têm haver com os comandos presentes em cada bloco do S.E.	(15 minutos)
10. Entender as perspectivas computacionais <ul style="list-style-type: none">- Explicar o que são e onde foram utilizadas as perspectivas computacionais ao longo do curso.	(15 minutos)

11. Entender as práticas computacionais - Explicar o que são e onde foram utilizadas as práticas computacionais ao longo do curso.	(15 minutos)
12. Entender melhor o pensamento computacional, a programação visual, habilidades cognitivas e a computação criativa.	(30 minutos)
13. Projeto final – Seja Criativo - Os alunos vão desenvolver na ferramenta <i>Scratch</i> um diálogo com uma temática indígena livre, podendo ser uma continuidade da atividade “Seja criativo” da aula anterior onde os presentes descreveram ideias para a construção do projeto desse módulo.	(60 minutos)
14. Formulário Aula 04	(30 minutos)
TOTAL em minutos	240 minutos

Softwares utilizados

Nome do Software: Microsoft *PowerPoint*

Versão do Software: Microsoft Office Professional Plus 2010

Nome do Software: *Scratch*

Versão do Software: *Scratch 1.4*

Projeto final:

Nome do exercício: Seja Criativo

Objetivos da Prática:

Fazer com que os alunos desenvolvam um diálogo de no mínimo dois *Sprites* no ambiente *Scratch* utilizando uma temática indígena livre, com o maior número de comandos aprendidos no curso, onde os alunos podem se basear nas atividades já desenvolvidas no curso bem como na atividade da aula anterior “Seja criativo” onde eles começaram a pensar em termos de algoritmo e estruturar as ideias que queriam conceber nessa atividade do projeto final.

Passo a passo:

- Planeje seu algoritmo (Pense no que você deseja criar no *Scratch* e transforme em uma sequência de passos utilizando os comandos do S.E.);
- Utilize todos os blocos de comandos vistos até agora em sala de aula (Movimento, Controle, Aparência, Som, Caneta, Sensores, Operadores e variáveis);
- Explore bem cada bloco de comando, pois quanto mais comandos você utilizar melhor será o resultado final do seu projeto;
- Seja criativo, use e abuse da computação criativa e da programação visual.













RECURSOS MATERIAIS





- Data show.
- Material de apoio (apostila).
- Computador com acesso à internet.

BIBLIOGRAFIA






MARJI, M. Aprenda a programar com Scratch. [S.l.]: novatec, 2014.

ANEXOS






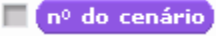






MOVIMENTO	
	Move o <i>Sprite</i> para frente e para trás na direção pré-definida.
	Roda o <i>Sprite</i> no sentido dos ponteiros do relógio.
	Roda o <i>Sprite</i> no sentido contrário ao dos ponteiros do relógio.
	Aponta o <i>Sprite</i> para uma determinada direção (0 = para cima; 90 = direita; 180 = para baixo; -90 = esquerda; ou outro valor angular introduzido através do teclado).
	Aponta o <i>Sprite</i> na direção do <i>mouse</i> ou para outro <i>Sprite</i> .
	Move o <i>Sprite</i> para uma posição específica no Palco, correspondente às coordenadas X e Y.
	Move o <i>Sprite</i> para a localização do rato ou de um <i>Sprite</i> .
	Move o <i>Sprite</i> suavemente para uma posição específica, num determinado período de tempo.
	Altera a posição do <i>Sprite</i> de acordo com o valor especificado.
	Define a posição do <i>Sprite</i> de acordo com o valor especificado.
	Altera a posição do <i>Sprite</i> de acordo com o valor especificado.
	Define a posição do <i>Sprite</i> de acordo com o valor especificado.









	O <i>Sprite</i> volta-se para a posição oposta ao tocar numa das margens do Palco.
	Indica a posição X do <i>Sprite</i> (de -240 a 240).
	Indica a posição Y do <i>Sprite</i> (de -180 a 180).
	Indica a direção para onde o <i>Sprite</i> aponta.

MOTOR (MOVIMENTO)


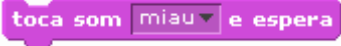



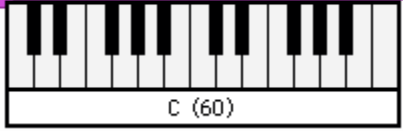






	Liga o motor durante o período de tempo indicado.
	Liga o motor.
	Desliga o motor.
	Define a potência do motor (varia entre 0 e 100) e liga-o.
	Define ou muda a direção do motor (nesta direção = sentido dos ponteiros do relógio; nessa direção = sentido contrário aos do ponteiro do relógio; inverter = muda de direção).



APARÊNCIA

	<p>Muda para o traje cujo nome é especificado.</p>
	<p>Muda para o traje seguinte, de acordo com a lista de trajes (<i>chegando ao final da lista, volta ao início</i>).</p>
	<p>Indica o número do traje, de acordo com a lista de trajes.</p>
	<p>Muda para o cenário cujo nome é especificado.</p>
	<p>Muda para o cenário seguinte de acordo com a lista de cenários (<i>chegando ao final da lista, volta ao início</i>).</p>
	<p>Indica o número do cenário, de acordo com a lista de cenários.</p>
	<p>Mostra o balão de discurso do <i>Sprite</i>, por um determinado período de tempo.</p>
	<p>Mostra o balão de discurso do <i>Sprite</i> (<i>para remover este balão, usar este comando sem qualquer texto</i>).</p>
	<p>Mostra o balão de pensamento do <i>Sprite</i>, por um determinado período de tempo.</p>
	<p>Mostra o balão de pensamento do <i>Sprite</i> (<i>para remover este balão, usar este comando sem qualquer texto</i>).</p>
	<p>Altera o efeito visual num <i>Sprite</i>, pelo valor especificado (usa o menu <i>pull-down</i> para escolheres o efeito).</p>
	<p>Define um efeito visual num <i>Sprite</i> para um determinado valor (<i>a maioria dos efeitos visuais varia entre 0 e 100</i>).</p>

	Termina todos os efeitos visuais num <i>Sprite</i> .
	Altera o tamanho do <i>Sprite</i> pela percentagem especificada.
	Define o tamanho do <i>Sprite</i> para a percentagem indicada.
	Indica o tamanho do <i>Sprite</i> em percentagem do tamanho original.
	Faz aparecer o <i>Sprite</i> no Palco.
	Faz desaparecer o <i>Sprite</i> do Palco (<i>quando um Sprite está escondido, os outros não podem detectá-lo com o comando "a tocar em"</i>).
	Move um <i>Sprite</i> para frente de todos os outros.
	Move o <i>Sprite</i> para trás, por camadas, de forma a ficar por detrás de outros <i>Sprite</i> .








SOM







	<p>Começa a tocar o som selecionado no menu <i>pull-down</i>, e avança para o comando seguinte sem esperar que o som termine.</p>
	<p>Toca um som e espera que o mesmo termine, antes de passar para o comando seguinte.</p>
	<p>Para de tocar todos os sons.</p>
	<p>Toca um som de percussão, escolhido no menu <i>pull-down</i>, durante o tempo especificado.</p>
 	<p>Toca uma nota musical durante o tempo especificado (<i>quanto maior o número, mais aguda a nota, 60 = Dó Central</i>).</p>
	<p>Para de tocar durante o período indicado.</p>
	<p>Escolhe um instrumento musical (<i>pode escolher-se um instrumento diferente para cada Sprite e pode alterar-se sempre que se quiser</i>).</p>
	<p>Altera o volume, num valor percentual especificado.</p>
	<p>Define o volume do som para a percentagem especificada.</p>
	<p>Indica o volume do som.</p>
	<p>Altera o ritmo num valor especificado (<i>tempo musical</i>).</p>














	Define o ritmo para o valor indicado, em batidas por minuto (<i>tempo musical, em bpm, afeta os tempos dos comandos "toca... durante"</i>).
	Indica o ritmo (<i>tempo musical, em bpm</i>).


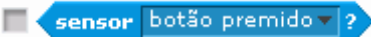
CANETA	
	Limpa do Palco as marcas da caneta e dos carimbos.
	Desce a caneta para que assim desenhe enquanto se move.
	Sobe a caneta para que não desenhe enquanto se move.
	Define a cor da caneta, de acordo com a cor especificada.
	Muda a cor da caneta, pelo valor especificado.
	Define a cor da caneta, de acordo com o valor especificado.
	Muda o tom da caneta, pelo valor especificado.
	Define o tom da caneta (<i>tom=0 muito escuro; tom=100 muito claro</i>).
	Altera a espessura da caneta, pelo valor especificado.
	Define a espessura da caneta, de acordo com o valor especificado.
	Carimba a imagem do <i>sprite</i> no Palco.

CONTROLE

	<p>Executa o bloco de comandos quando a bandeira verde for clicada.</p>
	<p>Executa o bloco de comandos quando a tecla especificada for premida.</p>
	<p>Executa o bloco de comandos quando o <i>Sprite</i> for clicado.</p>
	<p>Espera o número de segundos indicado e então continua no comando seguinte.</p>
	<p>Executa os comandos contidos no seu interior e repete indefinidamente (<i>até haver um comando explícito de paragem ou um clique em STOP</i>).</p>
	<p>Repete a execução de todos os comandos contidos no seu interior, tantas vezes quantas as especificadas.</p>
	<p>Envia uma mensagem e aguarda que todos os <i>Sprite</i> executem os blocos de comandos acionados por essa mensagem.</p>
	<p>Envia uma mensagem para todos os <i>Sprites</i> e prossegue de imediato para a execução do comando seguinte.</p>
	<p>Chapéu de bloco de comandos que aguarda pela emissão de uma mensagem para executar os comandos desse bloco.</p>
	<p>Executa os comandos contidos no seu interior e repete indefinidamente, sempre que a condição for verdadeira; suspende a execução se a condição não se verificar (<i>termina com um comando explícito de paragem ou um clique em STOP</i>).</p>

	<p>Executa os comandos contidos no seu interior uma única vez, se a condição for verdadeira.</p>
	<p>Se a condição for verdadeira, corre o bloco contido no interior de “se”. Se tal não acontece, corre os blocos que se encontrem no interior de “senão”.</p>
	<p>Espera até que uma determinada condição seja verdadeira e só depois avança para a execução do comando seguinte.</p>
	<p>Executa os comandos contidos no seu interior e repete indefinidamente até que a condição se torne verdadeira; se a condição já for verdadeira, avança para o comando seguinte sem executar nenhuma vez.</p>
	<p>Para a execução do bloco de comandos em que estiver inserido.</p>
	<p>Para a execução de todos os blocos de comandos de todos os <i>Sprites</i>; equivale a um clique no botão STOP.</p>










SENSORES	
	Indica o valor da posição X do mouse.
	Indica o valor da posição Y do mouse.
	"Verdadeiro" quando o botão do mouse está pressionado.
	"Verdadeiro" quando a tecla especificada está pressionada.
	"Verdadeiro" se o <i>Sprite</i> estiver a tocar num dado <i>Sprite</i> , na margem ou no cursor/rato (a seleção é feita no <i>menupull-down</i>).
	"Verdadeiro" se o <i>Sprite</i> estiver a tocar na cor especificada (clica na cor, para depois escolheres a que desejas).
	"Verdadeiro" se a primeira cor, contida no <i>Sprite</i> , tocar na segunda (outro <i>Sprite</i> ou cor de fundo) (clica na cor, para depois escolheres a que desejas).
	Indica o valor da distância até ao mouse ou até ao <i>Sprite</i> especificado.
	O relógio interno do <i>Scratch</i> nunca para, mas recomeça a contar do zero com este comando.
	Indica, em décimas de segundo, o tempo decorrido desde a última inicialização do relógio.
	Assume o valor de uma determinada propriedade do <i>Sprite</i> especificado.
	Indica o volume (de 1 a 100) dos sons detectados pelo microfone do computador.
	"Verdadeiro" se o microfone detectar um som cujo volume seja superior a 30 (numa escala de 1 a 100).








	<p>Indica o valor do sensor especificado (para usar este comando é necessário um PicoBoard ligado ao computador. Mais informações em http://www.playfullinvention.com/picoboard.html)</p>
	<p>"Verdadeiro", se for verificada a condição indicada no sensor especificado (para usar este bloco é necessário ter um PicoBoard ligado ao computador. Mais informações em http://www.playfullinvention.com/picoboard.html)</p>

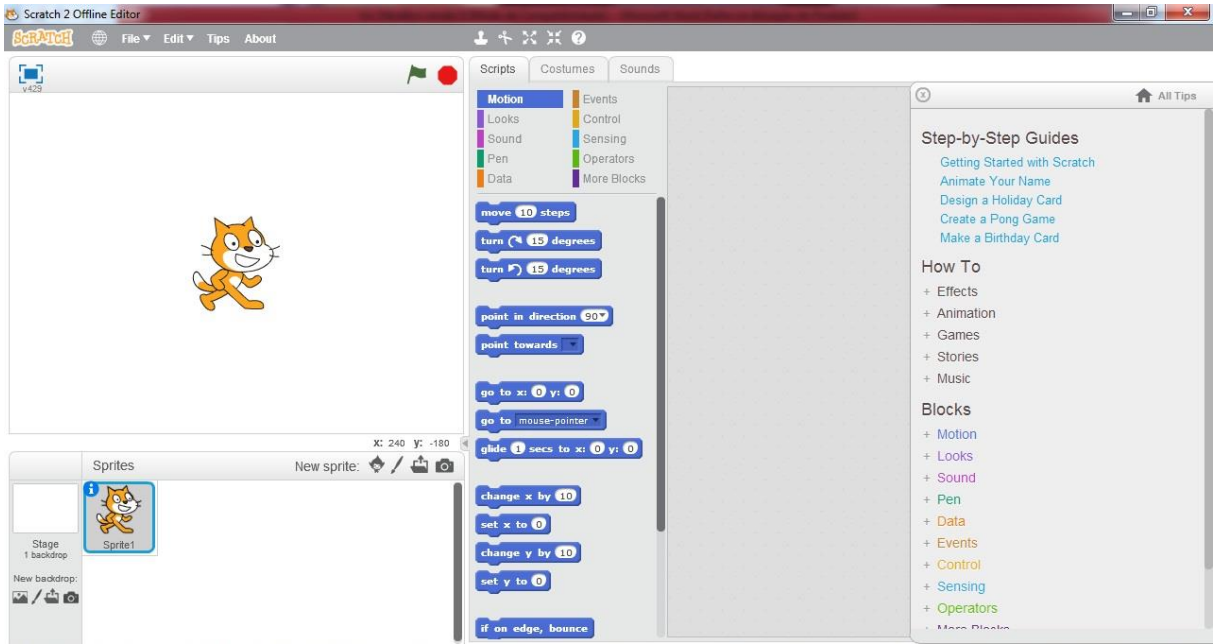
OPERADORES

	Soma os dois números.
	Subtrai o segundo número do primeiro.
	Multiplica os dois números.
	Divide o primeiro número pelo segundo.
	Escolhe aleatoriamente um número dentro do intervalo especificado.
	"Verdadeiro" se o primeiro valor for inferior ao segundo.
	"Verdadeiro" se os dois valores forem iguais.
	"Verdadeiro" se o primeiro valor for superior ao segundo.
	"Verdadeiro" se ambas as condições forem simultaneamente verdadeiras.
	"Verdadeiro" se ambas ou apenas uma das condições forem verdadeiras.
	"Verdadeiro" se a condição for falsa. "Falso" se a condição for verdadeira.
	Toma o valor do Resto da Divisão inteiro do primeiro número pelo segundo.
	Indica o valor inteiro mais próximo.
	Indica o resultado da função escolhida aplicado ao número especificado (<i>abs</i> , <i>raíz quadrada</i> , <i>sin</i> , <i>cos</i> , <i>tan</i> , <i>asin</i> , <i>acos</i> , <i>atan</i> , <i>ln</i> , <i>log</i> , e^x , 10^x).

VARIÁVEIS

	<p>Cria uma variável com o nome que se quiser e faz aparecer os comandos destinados a manipulá-la (<i>uma variável pode ser válida para todos os Sprites ou apenas para aquele onde é definida</i>).</p>
	<p>Permite apagar uma variável (<i>apagar uma variável não elimina a sua aplicação nos comandos, pelo que enquanto estiver em utilização será regenerada sempre que se guardar o projeto</i>).</p>
	<p>Valor de uma variável (<i>neste exemplo chamado "resultado"</i>).</p>
	<p>Atribui um valor à variável especificada.</p>
	<p>Altera a variável pelo valor especificado (<i>se tiveres mais do que uma variável, usa o menupull-down para escolher qual</i>).</p>
	<p>Mostra o valor da variável ecrã, durante a execução do projeto.</p>
	<p>Esconde o valor da variável no ecrã, durante a execução do projeto.</p>
	<p>Cria uma lista com o nome que se quiser e faz aparecer os comandos destinados a manipulá-la (<i>uma lista pode ser válida para todos os sprites ou apenas para aquele onde é definida</i>).</p>
	<p>Permite apagar uma lista (<i>apagar uma lista não elimina a sua aplicação nos comandos, pelo que enquanto estiver em utilização será regenerada sempre que se guardar o projeto</i>).</p>

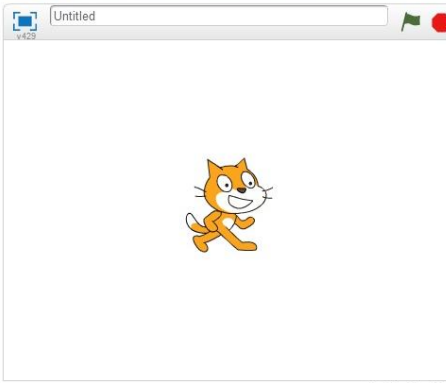
	<p>Conteúdo da lista, isto é, valor de todos os itens de que a lista se compõe (<i>neste exemplo chamado "minha lista"</i>).</p>
	<p>Adiciona o item especificado no final da lista (<i>pode ser uma frase, um número, ou qualquer outro conjunto de caracteres</i>).</p>
	<p>Apaga um ou todos os itens da lista (<i>podes escolher através do menupull-down ou usar um número para indicar o item a eliminar; "tudo" apaga todos os itens da lista</i>).</p>
	<p>Insere um item numa determinada posição na lista (<i>podes escolher através do menupull-down ou usar um número para especificar a posição do item a adicionar; "último" coloca-o no fim; "qualquer" insere-o numa qualquer posição da lista</i>).</p>
	<p>Substitui um item da lista pelo valor indicado (<i>podes escolher através do menupull-down ou usar um número para especificar qual o item a substituir; "último" substitui o último item; "qualquer" substitui aleatoriamente um item qualquer da lista</i>).</p>
	<p>Indica o valor do item localizado na posição indicada (<i>podes colocar este comando dentro de outros</i>).</p>
	<p>Indica quantos itens se encontram na lista.</p>



scratch.mit.edu/projects/editor/?tip_bar=getStarted

Scratch Arquivo Editar Dicas Sobre Autentique-se para salvar Autenticar

Untitled



Roteiros Fantasia Sons

Movimento

- Aparência
- Som
- Caneta
- Variáveis
- Eventos
- Controle
- Sensores
- Operadores
- Mais Blocos

mova 10 passos

gire 15 graus

gire 15 graus

aponte para a direção 90 graus

aponte para

vá para x: -1 y: 0

vá para ponteiro do mouse

deslize por 1 segundos até x:

adicione 10 a x

mude x para 0

adicione 10 a y

mude y para 0


Atores Novo ator:

Sprite1

Palco 1 pano de fundo Novo pano de fundo

Getting Started with Scratch

Want to get started with Scratch? Try this!



Start moving

STEP INDEX

- 1 Start Moving
- 2 Add a Sound
- 3 Start a Dance
- 4 Again and Again
- 5 Say Something
- 6 Green Flag
- 7 Change Color