



UNIVERSIDADE DO ESTADO DE MATO GROSSO
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CAMPUS UNIVERSITÁRIO DE SINOP
MESTRADO PROFISSIONAL EM MATEMÁTICA EM
REDE NACIONAL PROFMAT



PEDRO PAULO ANACLETO

**ARDUINO NO ENSINO DE LÓGICA MATEMÁTICA: UMA APLICAÇÃO NO
PRIMEIRO ANO DO ENSINO MÉDIO.**

Sinop - MT

2018

PEDRO PAULO ANACLETO

**ARDUINO NO ENSINO DE LÓGICA MATEMÁTICA: UMA APLICAÇÃO NO
PRIMEIRO ANO DO ENSINO MÉDIO.**

Dissertação apresentada como requisito parcial à
obtenção do grau de Mestre em Matemática,
elaborada por Pedro Paulo Anacleto.

Orientador: Prof. Dr. Silvio César Garcia Granja
Coorientador: Prof. Dr. Érico Fernando de Oliveira
Martins

Sinop - MT

2018

A532a ANACLETO, Pedro Paulo .
Arduino no Ensino de Lógica Matemática Uma Aplicação no
Primeiro Ano do Ensino Médio / Pedro Paulo Anacleto – Sinop,
2018.
60 f.; 30 cm.(ilustrações) Il. color. (sim)

Trabalho de Conclusão de Curso
(Dissertação/Mestrado) – Curso de Pós-graduação Stricto Sensu
(Mestrado Profissional) Profmat, Faculdade de Ciências Exatas e
Tecnológicas, Câmpus de Sinop, Universidade do Estado de
Mato Grosso, 2018.
Orientador: Professor Doutor Silvio César Garcia Granja
Coorientador: Professor Doutor Érico Fernando de Oliveira
Martins

1. Arduino. 2. Programação. 3. Ensino Matemático. I. Pedro
Paulo Anacleto. II. Arduino no Ensino de Lógica Matemática:
Uma Aplicação no Primeiro Ano do Ensino Médio.
CDU 51

PEDRO PAULO ANACLETO

“Arduino no ensino de lógica matemática: uma aplicação no primeiro ano do ensino médio”

Dissertação apresentada ao Programa de Mestrado Profissional em Matemática em Rede Nacional da Universidade do Estado de Mato Grosso - UNEMAT no *Campus* Universitário de Sinop, para obtenção do título de Mestre em Matemática.

Aprovado em: 21/06/2018



Prof. Dr. Silvio César Garcia Granja - UNEMAT



Prof. Dr. Marcelo de Carvalho Borba – UNESP



Prof. Dr. Rodrigo Bruno Zanin - UNEMAT

Sinop/Mt – Junho/2018

Dedico este trabalho aos educadores, ávidos por buscar o melhor aos seus alunos e que estão sempre trabalhando pela única ciência capaz de transformar todas as outras.

AGRADECIMENTOS

Agradeço aos colegas de turma por estarem comigo durante essa caminhada e, em especial Diogo Albino de Queiroz e Thiago Mantovani Tonial. À toda equipe docente do Profmat / Unemat – Campus Sinop, pelo suporte e compreensão nos momentos em que as atividades da vida fora da universidade não me possibilitaram ser o aluno esperado, principalmente ao Prof. Dr. Silvio Cesar Garcia Granja que me orientou durante a execução deste trabalho.

Não seria possível participar do programa e realizar este estudo sem a ajuda de toda a EquiPPE, com destaque para Denize e Tuty, que certamente se esforçaram ao máximo para manter tudo funcionando enquanto eu me ausentava para participar das aulas, realizar trabalhos, estudar para provas e escrever a minha dissertação.

À minha esposa Paola Stefanelli, por sempre me apoiar, desde a decisão de nos mudarmos da cidade de São Carlos – SP para Sinop – MT, e ser peça fundamental em minha vida. Tenho certeza de que não conseguiria alcançar meus objetivos sem você, e à toda nossa família, que sente a mesma saudade causada pela distância que nós, mas compreende que o fato de estarmos aqui nos faz felizes.

Obrigado a todos os mestres que passaram por minha vida e me inspiraram a me tornar um dia um educador. Busco sempre em minha memória o melhor de cada um de vocês e procuro, à minha maneira, fazer o mesmo aos meus alunos.

Por fim cumprimento a todos aqueles que um dia foram meus alunos e aos que ainda serão, espero ser sempre um educador a ser lembrado, tanto pelos momentos mais leves, em que tudo ocorreu conforme o esperado, quanto pelas broncas e brigas, espero que na maioria das vezes necessárias, para que fosse possível a vocês de fato estar em um estado de atenção necessário ao melhor entendimento do que lhes era exposto.

"A educação é a arma mais poderosa que você
pode usar para mudar o mundo."

Nelson Mandela

RESUMO

O presente estudo visa demonstrar uma forma de aplicação para a plataforma de prototipagem *Arduino* no ensino de matemática. Emprega-se uma metodologia participativa no ensino de tópicos específicos de matemática em uma turma do primeiro ano do ensino médio na cidade de Sinop, MT, por meio de programação. Esta metodologia utiliza a escolha da plataforma de programação, seja virtual ou física, pelos estudantes após uma apresentação prévia e de uma mediação pelo professor da turma. A plataforma selecionada foi o *Arduino* e elegeu-se uma série de códigos de programação que necessitam de discussão sobre a matemática e grandezas física envolvidas para o adequado controle. A cada etapa de execução da pesquisa, foi executada uma avaliação sobre a motivação em executar e continuar as atividades propostas, assim como foi avaliado o papel do professor como mediador.

Palavras-Chave: *Arduino*, programação, ensino matemático

ABSTRACT

The present study aims to demonstrate a form of application for the Arduino prototyping platform in mathematics teaching. A participatory methodology is used in the teaching of specific topics of mathematics in a class of the first year of high school in the city of Sinop, MT, through programming. This methodology uses the choice of the programming platform, either virtual or physical, by the students after a previous presentation and a mediation by the class teacher. The selected platform was the Arduino and it was chosen a series of programming codes that need discussion about the mathematics and physical quantities involved for the proper control. At each stage of the research, an evaluation was performed on the motivation to execute and continue the proposed activities, as well as the evaluation of the role of the teacher as mediator.

Keywords: Arduino, programming, mathematics teaching

SUMÁRIO

1. INTRODUÇÃO	17
1.1 OBJETIVOS.....	19
1.1.1 OBJETIVO GERAL	19
1.1.2 OBJETIVOS ESPECÍFICOS.....	19
1.2 ORGANIZAÇÃO DO TRABALHO	20
2. REFERENCIAL TEÓRICO	21
2.1 MOTIVAÇÃO DE ALUNOS	21
2.2 CONSTRUTIVISMO	23
2.3 CONSTRUCIONISMO.....	24
2.4 ENSINO DE PROGRAMAÇÃO.....	26
2.5 ARDUINO.....	26
2.6 SCRATCH.....	28
3. METODOLOGIA: MATERIAIS E MÉTODOS	29
3.1 DESCRIÇÃO DA TURMA	29
3.2 BUSCA PELO MÉTODO MAIS ADEQUADO À SITUAÇÃO	30
3.3 DEMONSTRAÇÃO DO MÉTODO À TURMA.....	30
3.4 DETERMINAÇÃO DA SEQUÊNCIA DIDÁTICA	31
3.5 PREPARAÇÃO DA NOVA TURMA	31
3.6 AVALIAÇÃO DA MOTIVAÇÃO	32
4. DESENVOLVIMENTO DO TRABALHO.....	33
4.1 APRESENTAÇÃO E DEMONSTRAÇÃO DOS MÉTODOS À TURMA	33
4.2 A PLATAFORMA SELECIONADA E OS MATERIAIS EMPREGADOS	34
4.3 PREPARAÇÃO DA NOVA TURMA	35
4.4 PROJETOS SELECIONADOS PARA IMPLEMENTAÇÃO NO ARDUINO.....	36
4.4.1 PROJETO 1: A UTILIZAÇÃO DE POTÊNCIAS DE BASE 10	37
4.4.2 PROJETO 2: USO DE CRIPTOGRAFIA E PARÂMETROS DE COMPARAÇÃO	42
4.4.3 PROJETO 3: ORGANIZANDO INFORMAÇÕES NA SEQUÊNCIA CORRETA	49
4.4.4 PROJETO 4: UTILIZANDO VETORES E MATRIZES	51
4.4.5 PROJETO 5: UTILIZANDO FUNÇÕES PERIÓDICAS	53
4.5 CONSIDERAÇÕES SOBRE O TRABALHO	55
5. CONSIDERAÇÕES FINAIS.....	57
5.1 CONTRIBUIÇÕES DO TRABALHO	58
5.2 SUGESTÕES PARA PRÓXIMOS TRABALHOS	58

REFERÊNCIAS59

1. INTRODUÇÃO

Motivar adolescentes a aprender matemática e fazê-los compreender que esses conceitos são importantes não é tarefa fácil, principalmente tendo em vista o desempenho brasileiro em edições de provas como o *PISA* (*Programme for International Student Assessment*), apresentado na Tabela 1, que realiza uma pesquisa trienal para avaliar os sistemas de educação no mundo por meio de um teste com alunos de 15 anos.

Tabela 1: Desempenho brasileiro em matemática no PISA

Ano da prova	2000	2003	2006	2009	2012	2015
Colocação em Matemática	31°	40°	52°	57°	58°	66°

Fonte: INEP (2018).

Os resultados da última avaliação, que ocorreu em 2015, colocam o Brasil na posição 66, entre 72 nações participantes na disciplina de matemática, sendo que a média nacional foi de 377 pontos e a média mundial foi de 490 pontos (OECD, 2018).

É bastante claro o fato de que é necessário buscar formas de melhorar o processo educacional matemático no país. Para tal, é preciso motivar alunos e professores a buscarem novas maneiras de alcançar esse resultado.

Alguns alunos não veem com clareza o porquê de aprender e estudar determinados conteúdos. Muitos, por não enxergarem aplicações práticas de conceitos, dizem que nunca usarão tal conhecimento, e isso, provavelmente, faz com que eles se sintam menos estimulados a estudar.

O ensino de alguns conteúdos, na escola, muitas vezes se mostra abstrato. Fórmulas físicas, matemáticas e químicas, em muitos momentos, são incompreendidas porque falta a aplicação prática de conhecimento.

Para mudar essa realidade, uma solução é a abordagem construcionista, na qual pode-se escolher pela ministração de aulas que mostrem como um conteúdo poderá ser aproveitado no dia a dia de cada estudante. O Construcionismo busca mostrar que é possível aprender através de uma ação concreta que resulte em algo visível (PAPERT, 1985).

O Construcionismo de Seymour Papert (PAPERT, 1985) tem relações estreitas com o Construtivismo de Piaget (AZENHA, 1994); ambos defendem a valorização do cotidiano do aluno para o aprendizado. A teoria construcionista afirma que o conhecimento é “construído” a partir de experiências do cotidiano do aluno, de experiências que fazem sentido para ele, sendo

o professor um simples articulador entre o conhecimento prévio do aluno - aquele que ele traz de casa e das experiências anteriores já vividas - e o conhecimento que ele pode e deve alcançar. Dessa maneira, o aluno, a partir daquilo que lhe interessa, consegue alcançar o pleno conhecimento.

Com o intuito de alcançar esse conhecimento, buscou-se uma forma de trazer algo para a sala de aula que tivesse mais conexão com a vida dos alunos e para tal podem ser utilizados, por exemplo, conceitos básicos de programação.

O ensino de programação em fases da educação básica ainda é pouco difundido no Brasil, o que nos posiciona em um cenário onde o conhecimento sobre o assunto é bastante precário.

De acordo com Tucker (2003), para que os indivíduos do século XXI possam participar da sociedade, esses devem compreender ao menos os princípios da ciência da computação, e como exposto anteriormente não há no Brasil um número adequado de iniciativas com essa finalidade.

Não bastasse o fato de não estarmos preparando nossos alunos para ingressar em carreiras voltadas à tecnologia da informação, mesmo sendo grandes consumidores de novas tecnologias, mas sem grandes expoentes no seu desenvolvimento, também perdemos a oportunidade de desenvolver habilidades em resolução de problemas. Pensar de maneira criativa, organizar os pensamentos de maneira sistêmica e trabalhar colaborativamente são habilidades que podem ser construídas por meio do ensino de programação.

Desta forma o presente estudo busca demonstrar a aplicação de ferramentas que visam uma melhor compreensão matemática por parte dos alunos de uma turma do primeiro ano do ensino médio e descreve uma abordagem para a inserção de tal metodologia no dia a dia das escolas por meio de uma sequência didática que emprega a programação.

Tendo em vista que o ensino de programação não faz parte do currículo escolar em nenhuma das fases da educação básica no Brasil, a apresentação do tema foi feita utilizando a disciplina de matemática com alunos do primeiro ano do ensino médio em uma escola da rede particular de ensino na cidade Sinop-MT. O tema programação tem a intenção de contribuir como uma forma de demonstrar a utilização de conceitos matemáticos em situações reais. Isso pode fazer com que a motivação dos estudantes no estudo da matemática cresça de forma a conseguir um maior engajamento para poder desenvolver os projetos que lhes foram propostos.

Buscou-se na literatura disponível estudos sobre o tema *softwares* e *hardwares* utilizados por escolas do ensino médio no Brasil e no mundo para o ensino de programação. Dentre as alternativas disponíveis escolheram-se dois formatos que seriam passíveis de aplicação na escola: a linguagem *Scratch* e a plataforma *Arduino*, para demonstração.

Após demonstrar as ferramentas previamente selecionadas foi possível escolher, com base nos parâmetros definidos, uma única ferramenta que seria de fato executada com a turma ingressante do ano letivo seguinte.

A escolha pelo *Arduino* se deu principalmente pela possibilidade de os alunos do primeiro ano do ensino médio verificarem os resultados de seus trabalhos fisicamente: *LEDs* acendendo e apagando de acordo com os comandos enviados, e não só por meio da tela de um computador, *tablet* ou *smartphone*.

1.1 OBJETIVOS

Nessa subseção serão apresentados o objetivo geral e os objetivos específicos do presente trabalho.

1.1.1 OBJETIVO GERAL

Analisar o processo de motivação de alunos do primeiro ano do ensino médio, na resolução de problemas lógico-matemáticos por meio da plataforma de prototipagem *Arduino* ou da linguagem *Scratch*.

1.1.2 OBJETIVOS ESPECÍFICOS

- Pesquisar sobre o uso de novas plataformas para o ensino de programação durante a educação básica no Brasil e em outros países.
- Determinar qual metodologia melhor se adequava à escola onde o trabalho se desenvolveu, dadas as condições da instituição.
- Definir qual plataforma utilizar no desenvolvimento de projetos para o ensino de matemática dentre o *Scratch* e a plataforma *Arduino*.
- Escolher projetos a serem desenvolvidos na plataforma escolhida, de forma que o ensino de matemática se torne menos abstrato.
- Desenvolver uma sequência didática de ensino dos conceitos matemáticos supracitados, analisando a motivação gerada nos estudantes com o uso da plataforma escolhida.

1.2 ORGANIZAÇÃO DO TRABALHO

Esta dissertação apresenta no capítulo 2 o Referencial Teórico em que são abordados os conceitos de motivação, construtivismo e Construcionismo, além de explanar sobre as plataformas de programação com potencial de ensino e auxílio em matemática. O capítulo 3 descreve a Metodologia que foi seguida para a aplicação e análise dos procedimentos com a turma. No capítulo 4 são descritos os resultados da escolha e aplicação da sequência didática que foi determinada à partir do momento que a escolha pelo método mais adequado foi feita. Por fim, no capítulo 5, são feitas as Considerações Finais sobre o trabalho, descrição de suas contribuições e são feitas sugestões para trabalhos futuros.

2. REFERENCIAL TEÓRICO

O referencial teórico abaixo visa demonstrar as bases do conhecimento utilizadas para a execução do trabalho.

Para tal, relacionam-se os conceitos da Teoria de Motivação, do Construtivismo e Construcionismo e do ensino de programação por meio das linguagens de programação do *Scratch* e do *Arduino*.

2.1 MOTIVAÇÃO DE ALUNOS

O vocábulo “motivação” de acordo com Dicio (2018) é a reunião de razões que levam alguém a agir de acordo com um determinado comportamento, ou um processo que dá origem a uma ação pensada.

Quando se fala em motivação para a aprendizagem no contexto escolar, é importante, também, entender que existem diferentes realidades. As particularidades do ambiente devem ser levadas em conta, pois um ambiente agradável, limpo e confortável estimula mais o aprendizado do que aquele em que o aluno não tem as mínimas condições para um bom estudo.

Infelizmente, em algumas regiões do Brasil, alunos e professores travam uma batalha contra a falta de infraestrutura – prédio precário, escolas sem mesas, quadros, cadeiras e alunos sem livros e cadernos. Mesmo assim, em algumas comunidades carentes, o aluno se vê estimulado a estudar, percebe que o estudo é, muitas vezes, a sua única chance de mudar de vida. Por isso, estimular a motivação, independentemente do local, é papel do professor, mesmo que essa motivação transmitida seja desprezada por alguns alunos.

Para Godoi (2001), motivação é um processo psicológico, nela influenciam vários aspectos (família, saúde, finanças...), e a alteração da motivação pode ser indício de que algo está errado. Alunos que carregam para as salas de aula problemas familiares ou que são cobrados em extremo pela família podem ser vistos como desmotivados, e, por alguns professores, como incapacitados.

É importante que professores e comunidade escolar mostrem para os alunos o que eles devem fazer, mas, principalmente, o porquê de realizar uma atividade.

A motivação deve ser trabalhada com o aluno, e se o professor mostrar a ele o que ele pode alcançar através de uma boa educação, através de um ótimo currículo, ele será um aluno ainda mais motivado.

Segundo Godoi (2001), existem três tipos de motivação:

- Motivação por impulso cognitivo: provém da necessidade de progredir, passar de nível para evitar o fracasso.
- Motivação aversiva: quando a pessoa se motiva para evitar constrangimentos desnecessários. É forçada por uma pressão psicológica.
- Motivação por engrandecimento do ego: quando o cidadão nota por si só que está tendo progressos em alguma área e se motiva querendo mais.

A motivação tem duas origens: intrínseca (do próprio ser) e extrínseca (do meio que cerca o ser). Na primeira, o indivíduo procura realizar aquilo que ele deseja, é uma motivação interna, uma necessidade da pessoa que lhe causa prazer. Já a motivação extrínseca acontece simplesmente pelo fato do indivíduo querer receber recompensas e evitar as punições (GODOI, 2001).

Dessa forma, para ilustrar a motivação intrínseca, podemos citar o indivíduo de classe social baixa que deseja alcançar um cargo público de destaque. Para isso, ele trabalha e estuda muito, é movido por uma força interior. Esse mesmo indivíduo, em sua caminhada, consegue um estágio em um escritório, estágio necessário para que ele suba mais um degrau em sua carreira, nesse momento, ele passa a ser movido pela motivação extrínseca, pois a expectativa de galgar mais um estágio faz com que ele cumpra suas tarefas as mais corretas possíveis e, assim, receba a sua recompensa.

De acordo com Godoi (2001), as categorias de motivação são:

- Categoria Pulsional: o indivíduo realiza as suas atividades para o seu próprio bem-estar. Acontece com o indivíduo que gosta muito de música e que para satisfação pessoal, busca aprender vários instrumentos, isso lhe traz uma sensação de bem-estar e satisfação pessoal.
- Categoria Cognitiva: o indivíduo realiza as suas atividades pensando nos benefícios que ela pode lhe proporcionar. Assim, um adolescente que está prestes a prestar um exame vestibular, realiza todas as suas atividades escolares sabendo que aquilo poderá lhe trazer benefícios no momento do exame.
- Categoria Social: quando o indivíduo trabalha com o intuito de agradar os que o cercam. Um exemplo que ilustra muito bem essa categoria é a do filho que abdica de seus sonhos para satisfazer os de seus pais, como na escolha da futura profissão. A pessoa que assim o faz tem uma grande necessidade de ser aceita e de agradar aqueles que estão à sua volta.

Portanto, sem motivação o ser humano não vive, não produz, não progride. É importante que o indivíduo se sinta motivado em todas as áreas. Na educação, os alunos precisam ser motivados pelo professor, que precisa estar motivado para ensinar. Assim, com professores e alunos motivados, o aprendizado será efetivo e melhor para todos.

2.2 CONSTRUTIVISMO

De acordo com Azenha (1994), o Construtivismo é uma linha de pensamento que ganhou espaço no campo das teorias pedagógicas e que é baseada nos estudos de Jean Piaget, biólogo, psicólogo e epistemólogo suíço, considerado um dos mais importantes pensadores do século XX. Essa concepção condena a rigidez no ensino, as avaliações padronizadas e a utilização de materiais didáticos estranhos para o aluno e ainda considera o erro como uma forma de crescimento intelectual.

Muitas vezes, é necessário conhecer a vida e obra de uma pessoa para que suas teorias sejam entendidas. De acordo com o Portal da educação (2018), Piaget já demonstrava interesse pela ciência quando criança. Com 10 anos de idade, já estudava moluscos e observava pássaros. Sua primeira publicação foi fruto de uma observação de um pardal parcialmente albino.

Com 11 anos de idade, Piaget era assistente do diretor do museu de história natural de sua cidade natal Neuchâtel. Antes dos 20 anos de idade ele publicou vários estudos sobre a natureza. Em 1918, doutorou-se em ciências naturais (PORTAL DA EDUCAÇÃO, 2018).

De acordo com Azenha (1994) Piaget se debruçou sobre a obra de Freud, visitou clínicas psiquiátricas e participou de cursos na Sorbonne, período altamente produtivo para o pai do construtivismo e de formulações teóricas, como a de que todo sistema “vivo” tem diferentes possibilidades de equilíbrio.

Piaget argumenta que o Construtivismo seria a solução para se conhecer as origens do conhecimento humano.

Cinquenta anos de experiência ensinaram-nos que não existem conhecimentos resultantes de um simples registro de observações, sem uma estruturação devida às atividades do indivíduo. Mas, tampouco, existem (no homem), estruturas cognitivas a priori ou inatas: só o funcionamento da inteligência é hereditário, e só gera estruturas mediante uma organização de ações sucessivas, exercidas sobre os objetos. Daí resulta que uma epistemologia em conformidade com os dados da psicogênese não poderia ser empírica nem pré-formista, mas não pode deixar de ser um Construtivismo, com a elaboração contínua de operações e de novas estruturas. O problema central consiste, pois, em compreender como se efetuam tais criações e por que, ainda que resultem construções não pré-determinadas, elas podem acabar por se tornar logicamente necessárias (Piaget, 1983 apud Azenha, 1994).

Segundo Piaget, as crianças têm estruturas lógicas próprias de raciocínio diferentes dos adultos, que evoluem com a mudança de faixas de idade pré-estabelecidas. A teoria se preocupa em desenvolver um aprendizado próprio para cada estágio do desenvolvimento.

No Construtivismo, o professor é um facilitador da aprendizagem e sua principal função é colocar o aluno frente a problemas práticos ou teóricos para que achem a solução e construam o seu conhecimento. Para que isso aconteça, de forma eficaz, o conhecimento adquirido com a experiência de vida deve ser aproveitado (AZENHA, 1994).

De acordo com Azenha (1994), o Construtivismo prega que o aluno não aprende de forma passiva, sem incentivo. Segundo a teoria construtivista, o aprendizado deve ser dinâmico e não estático como em métodos mais tradicionais. O professor deve provocar a curiosidade de seu aluno, fazê-lo pensar, buscar respostas, provocá-lo para que ele saia da zona de conforto, para que se sinta desafiado. Esse desafio fará com que o aluno cresça em maturidade, melhorando as suas estruturas de pensamento, julgamento, raciocínio lógico e argumentação.

No Construtivismo, erros são oportunidades de crescimento, momento de reflexão para que as imperfeições sejam corrigidas, tanto para professores quanto para os alunos. No meio acadêmico muito se fala que “é errando que se aprende”, para Piaget isso é a pura verdade (AZENHA, 1994).

Os erros cometidos pelas crianças na solução dos problemas propostos eram os aspectos mais especialmente interessantes para Piaget. O que o fascinava era a compreensão lógica subjacente ao erro e à interpretação do percurso intelectual da criança em relação ao seu desenvolvimento cognitivo global.

Portanto, através de uma metodologia que respeita os conhecimentos adquiridos, o professor deve implantar o seu método de trabalho sempre respeitando a individualidade de cada um, assim se cumprirá o ideal construtivista de se aprender interagindo com o meio.

2.3 CONSTRUCIONISMO

Seymour Papert, nascido em 29 de fevereiro de 1928, em Pretória, África do Sul, e falecido em 31 de julho de 2016, foi um matemático, considerado um dos pais da inteligência artificial. Internacionalmente, ficou conhecido por pensar em meios de se desenvolver a aprendizagem com o uso da informática. Nascido na época em que os computadores foram inventados e começaram a se desenvolver, introduziu a linguagem da programação em suas aulas, algo totalmente inovador na época. Isso provocou, no aluno, uma sensação de domínio

sobre a máquina. Para Papert, as máquinas devem auxiliar na construção do conhecimento (DAILY PAPERT, 2018).

O professor precisa compreender a representação da solução do problema adotado pelo aluno; acompanhar a depuração e tentar identificar as hipóteses, os conceitos e os possíveis equívocos envolvidos no programa; e assumir o erro como defasagem ou discrepância entre o obtido e o pretendido. Assim, o professor intervém no processo de representação do aluno, ajuda-o a tomar consciência de suas dificuldades e a superá-las; a compreender os conceitos envolvidos; a buscar informações pertinentes; a construir novos conhecimentos; e a formalizar esses novos conhecimentos (ALMEIDA, 1999).

A frase “instrução ajudada pelo computador” (computer-aided-instruction) significa fazer com que o computador ensine a criança. Pode-se dizer que o computador está sendo usado para “programar” a criança. A criança deve programar o computador e, ao fazê-lo, ela adquire um sentimento de domínio sobre um dos mais modernos e poderosos equipamentos tecnológicos e estabelece um contato íntimo com algumas das ideias mais profundas da ciência, da matemática e da arte de construir modelos intelectuais (PAPERT, 1985).

O Construcionismo permite ao educando a construção de seu próprio conhecimento por intermédio de alguma ferramenta. O computador passa a ser apenas um suporte a ser explorado, investigado e descoberto pelo aluno que o explora. O professor é apenas o mediador. O aluno, na abordagem Construcionista, constrói algo do seu interesse, o que torna o aprendizado mais significativo. Uma linguagem de programação que ilustra como o Construcionismo pode atuar e auxiliar na aprendizagem é a LOGO, idealizada por Papert e Wally Feurzeig (PAPERT, 1985).

Elaborar um programa significa manipular um sistema de palavras e de regras formais, que constituem a sintaxe e a estrutura da linguagem, que dão suporte para se representar os conhecimentos e as estratégias necessários à solução do problema. O conhecimento não é fornecido ao aluno para que ele dê respostas. É o aluno que coloca conhecimento no computador e indica as operações que devem ser executadas para produzir as respostas desejadas (ALMEIDA, 1994).

De acordo com Papert (1985), a linguagem LOGO tem cinco características:

- 1) Amigabilidade: é de fácil acesso e manuseio.

- 2) Modularidade e Extensabilidade: novos comandos podem ser criados usando-se da própria linguagem LOGO.
- 3) Interatividade: respostas imediatas aos questionamentos e informações sobre os comandos aplicados.
- 4) Flexibilidade: pode ser usada por pessoas em diferentes faixas etárias.
- 5) Capacidade: utilizada na criação de programas simples ou mais sofisticados.

A linguagem LOGO é somente um dos recursos que os professores têm. Em um mundo tão informatizado e de respostas rápidas, os alunos precisam aprender a pensar e a construir os seus próprios caminhos, sem respostas ou fórmulas prontas. Assim, seguindo a teoria Construcionista, e com aprendizado dinâmico e prazeroso, os alunos aprenderão mais e melhor, desenvolverão suas capacidades criativas e serão profissionais diferenciados no mercado de trabalho.

2.4 ENSINO DE PROGRAMAÇÃO

Quase todos os currículos de cursos de ciências exatas exigem que os alunos detenham conhecimento sobre programação, mas esta é uma das disciplinas que mais desafiam os estudantes. Um estudo realizado pelo grupo McCracken em vários países e em diferentes instituições mostrou que a nota média dos alunos do primeiro ano dos cursos de ciências exatas era de 23 em 110, além disso percebe-se também que a taxa de desistências nessas disciplinas está entre 30% e 40% segundo Ozoran, Cagiltay e Topalli (2012).

De acordo com Ozoran, Cagiltay e Topalli (2012) a maior causa dos problemas dos adolescentes em aprender programação é a falta de motivação, sendo que para o nível universitário há uma maior variação dessas causas. Muitos dos problemas encontrados na graduação se relacionam à complexidade dos conceitos e sintaxe das linguagens de programação.

Tais observações evidenciam a importância de iniciativas como o Construcionismo de Papert, que buscam aproximar o aluno do conteúdo estudado fazendo com que ele se sinta responsável pela construção de seu próprio conhecimento.

2.5 ARDUINO

De acordo com Arduino (2018b) o *Arduino* é uma plataforma de prototipagem e aprendizado de eletrônica e aplicações *open source* baseada no uso de *hardware* e *software* fáceis de utilizar, sendo possível a ela ler entradas e gerar saídas com base nessas informações.

Para que isso aconteça é necessário utilizar o *Arduino Software (IDE)*, que é um Ambiente de Desenvolvimento Integrado que tem uma linguagem própria baseada em um subconjunto da linguagem C, e enviar essas instruções ao micro controlador da placa.

O fato de a plataforma ser dita *open source*, código aberto em português, significa que algumas premissas devem ser respeitadas. De acordo com o Open Source Initiative (2017), 10 critérios que definem o que é *open source*, e descrevem que ele deve possibilitar a terceiros sua manipulação e utilização em outros projetos sem que isso gere direitos aos criadores da plataforma.

Milhares de projetos já foram criados utilizando o *Arduino* e uma comunidade mundial de estudantes, artistas, programadores e outros profissionais adicionam uma incrível quantidade de conhecimento que está acessível para ajudar novatos e especialistas em seus projetos (ARDUINO, 2018b).

Inicialmente a ideia era criar uma ferramenta para prototipagem rápida e direcionada a estudantes sem experiência em projetos eletrônicos e programação, mas por ter atingido uma ampla comunidade, hoje existem projetos envolvendo uma série de aplicações para *IOT (Internet of Things)*, internet das coisas em português, impressoras 3D, vestimentas e sendo incorporados a vários ambientes (ARDUINO, 2018b).

A placa permite que diversos tipos de componentes eletrônicos, que geram saídas ou captam entradas do ambiente, sejam adicionados, direcionados e programados para uma determinada atividade. Para tal é utilizado um Microcontrolador ATMEGA que tem a função de receber e entregar o fluxo de informações de maneira controlada por uso de software (SILVA, 2014).

Segundo Souza et al. (2011) a placa *Arduino* é baseada em um microcontrolador que vai além de uma simples interface de aquisição de dados, permitindo ao usuário desenvolver aplicações em instrumentação embarcada e robótica, sendo que a comunidade internacional que interage por meio de sites, blogs e eventos patrocinados por diferentes instituições mundo afora gera uma farta documentação sobre projetos e troca de informações entre as pessoas, o que diminui bastante as dificuldades encontradas no início do uso dessa tecnologia.

Souza et al. (2011) salientam que “Outro fato positivo é que atualmente podemos encontrar diferentes versões nacionais da placa *Arduino*, no mercado nacional, com preços muito acessíveis ou mesmo, instruções de montagem a partir dos componentes eletrônicos

básicos, o que pode atender o interesse de professores e alunos com maior capacitação em eletrônica.”

2.6 SCRATCH

O *Scratch* é um projeto criado pelo *Lifelong Kindergarten Group* do MIT (*Massachusetts Institute of Technology*) *Media Lab*, que possibilita a criação de estórias, jogos e animações com o objetivo de ajudar jovens a pensar de maneira criativa, raciocinar sistematicamente e trabalhar em colaboração com outras pessoas por meio de uma comunidade online (SCRATCH, 2018).

Apesar de ter sido desenvolvido para pessoas entre oito e dezesseis anos, o *Scratch* é utilizado por milhões de pessoas de todas as idades para a criação de projetos em uma grande variedade de configurações. Estudantes desde os níveis elementares até o colégio estão aprendendo matemática, ciência da computação, linguagens e estudos sociais utilizando o *Scratch* (SCRATCH, 2018).

SCRATCH (2018) destaca que mais de cento e cinquenta países utilizam a plataforma de programação que está disponível em mais de quarenta línguas, e atualmente existem mais de vinte e oito milhões de projetos compartilhados no site do projeto.

Por se tratar de uma linguagem de programação visual, o *Scratch* torna o aprendizado mais fácil e mais divertido, não sendo necessário digitar nenhum comando complicado, mas sim conectar bloco gráficos para criar programas (MARJI, 2014).

Ouahbia et al. (2015) salienta que muitas plataformas de programação introdutória foram desenvolvidas para evitar que alguns erros comuns aos iniciantes, como erros de sintaxe e lógica acontecessem. Para isso ao invés de digitar comandos usa-se o conceito de programação em blocos, nesse contexto cada bloco é um elemento da linguagem de programação, seja uma estrutura de controle, um operador, uma variável ou uma função por exemplo. A combinação de tais elementos acontece arrastando os blocos para uma formação intuitiva de acordo com certa lógica planejada para gerar um programa de computador.

3. METODOLOGIA: MATERIAIS E MÉTODOS

Para a realização do experimento foi necessário buscar na literatura iniciativas que tivessem sido aplicadas com o intuito de melhorar a compreensão matemática de alunos com o uso de ferramentas computacionais e de programação.

Dentre as opções vislumbradas, duas chamaram atenção e foram melhor estudadas, sendo elas o *Scratch* e o *Arduino*, expostas no capítulo 2. Após entender como as ferramentas poderiam ser utilizadas em sala de aula, resolveu-se pela demonstração desses métodos a uma turma com o intuito de compreender qual seria a reação dos alunos ao que lhes seria exposto.

Após a demonstração somente uma das ferramentas foi escolhida para ser trabalhada junto aos alunos da turma que iniciaria ano letivo no semestre seguinte, tendo em vista a impossibilidade de fazê-lo no ano de 2017.

Com a ferramenta definida foi possível determinar quais projetos seriam executados e para tal o livro *Arduino Básico* de Michael McRoberts de 2011 foi de suma importância, pois a forma de trabalho adotada nos projetos descritos está em consonância com os princípios descritos por Seymour Papert.

Dadas as limitações do tempo e do controle da população, para pesquisas quantitativas e longitudinais, optamos, neste momento, por um estudo de caso, restringindo nossa pesquisa à descrição da aplicação do método educacional a uma turma de alunos, bem como ao registro das impressões da população, em relação à aula.

3.1 DESCRIÇÃO DA TURMA

A turma que realizará os projetos é composta por dezesseis alunos do primeiro ano do ensino médio de uma escola da rede particular de ensino no município de Sinop – MT.

Todos os alunos estão na faixa etária adequada à série cursada, entre quatorze e quinze anos de idade, sendo seis meninas e dez meninos, com doze deles residindo com a família no município de Sinop – MT, um de Sorriso – MT, um de Terra Nova do Norte – MT e dois de Novo Progresso – PA, sendo que dos dezesseis dois já moraram respectivamente seis meses e um ano nos Estados Unidos da América, algo que contribuiu para a realização de um dos projetos.

3.2 BUSCA PELO MÉTODO MAIS ADEQUADO À SITUAÇÃO

Inicialmente vislumbrou-se a possibilidade de trabalhar com uma série de softwares, hardwares e metodologias. De acordo com Medeiros, Silva e Aranha (2013) onze ferramentas se destacam no auxílio do ensino de programação, sendo elas *PyGame*, *RoboMind*, *Lego Mindstorms*, *Takkou*, *Scratch*, *Alice*, *iVprog*, *Escracho*, *Kodu*, *Game Maker* e *Construct 2*, sendo que destas a que mais chamou atenção, devido à existência de uma maior concentração de pessoas utilizando e uma rede social onde é possível publicar trabalhos e verificar o que outros estão fazendo, foi o *Scratch*.

Segundo Rubio, Hierro e Pablo (2013) ao introduzir a plataforma *Arduino* em conjunto com formas de programação como o *Scratch* mais estudantes aprendem a programar e passam a encontrar uma relação física mais forte com os conceitos lógico-matemáticos encontrados na Ciência, Tecnologia, Engenharia e Matemática, que em inglês é chamado de *STEM* (*Science, Technology, Engineering, and Mathematics*).

Com essas informações em mente decidiu-se pela apresentação do *Scratch* e da plataforma de prototipagem *Arduino* a uma turma de quinze alunos concluinte do primeiro ano do ensino médio no ano letivo de 2017 e a partir da reação dos alunos realizar a aplicação da metodologia que mais se mostrasse interessante.

3.3 DEMONSTRAÇÃO DO MÉTODO À TURMA

De acordo com Marji (2014) o *Scratch* além de permitir o desenvolvimento de habilidades relacionadas à resolução de problemas, fornece um *feedback* imediato sobre as ações, o que permite a conferência da programação de maneira rápida e fácil. Para tal, programou-se uma única aula para demonstrar aos alunos como a ferramenta funciona e alguns projetos prontos disponíveis no próprio site do *Scratch*, e para a apresentação seria necessária uma sala de aula, contendo carteiras para os alunos, um notebook e um projetor de imagens.

Durante a execução da aula será apresentado o programa de Introdução ao *Scratch*, disponível como uma das formas para aqueles que querem aprender como utilizar a ferramenta. A ideia é fazer com que o *Sprite1*, personagem do *Scratch*, dance de acordo com os blocos estruturados pelo usuário, e depois apresentar projetos realizados por outros usuários da plataforma principalmente na seção *Games*, onde estão disponíveis os jogos criados por pessoas que utilizam o *Scratch*. Essa apresentação inicial será feita de maneira expositiva, sem que os alunos possam efetivamente utilizar a ferramenta.

Após a apresentação do *Scratch*, será apresentada a plataforma de prototipagem *Arduino*, e para tal necessitaremos de uma sala de aula, contendo carteiras para os alunos, um notebook, um projetor de imagens e de um kit *Arduino* simples.

O kit deve conter uma placa *Arduino UNO R3*, um cabo USB para *Arduino*, cabos jumper macho-macho, uma *protoboard*, LEDs difusos amarelos, LEDs difusos verdes, LEDs difusos vermelhos, LEDs difusos azuis, LEDs difusos brancos, resistores de 150Ω , resistores 200Ω e resistores de 1000Ω . Com esse kit pode-se realizar o projeto LED piscante sugerido por McRoberts (2011), que consiste em fazer um único LED piscar de maneira intermitente, sendo possível alterar o tempo que o LED fica apagado e o tempo que o LED fica aceso. Novamente, a forma utilizada para tal apresentação será a exposição de um único kit, que já estará montado antes do início da aula, e será mostrado ao alunos juntamente do código necessário ao seu funcionamento.

3.4 DETERMINAÇÃO DA SEQUÊNCIA DIDÁTICA

À partir do resultado obtido com as apresentações serão definidos os projetos a serem executados, de acordo com a plataforma escolhida, podendo ser o *Scratch* ou o *Arduino*.

Durante esse processo será levado em consideração a abordagem construcionista descrita por Papert (1985).

3.5 PREPARAÇÃO DA NOVA TURMA

A turma que participou da primeira fase do experimento, durante o ano letivo de 2017, não possuía na grade curricular carga horária suficiente para concluir o projeto, mas findado o ano letivo foi acordado com a coordenação da escola a readequação da carga para que fosse possível trabalhar com a turma do ano seguinte durante os dois primeiros meses de aula e executar os cinco projetos propostos.

Inicialmente será realizada uma avaliação dos conhecimentos em matemática e ciências e sobre programação de computadores dos novos alunos, por meio de prova de múltipla escolha para avaliar o conhecimento matemático e em ciências e um questionário para identificar se havia experiências anteriores às aulas com o uso de programação e prototipagem.

Após essa primeira avaliação se iniciarão os trabalhos com os projetos de programação em *Scratch* ou prototipagem em *Arduino*.

3.6 AVALIAÇÃO DA MOTIVAÇÃO

Serão registrados comentários dos estudantes ao final de cada projeto executado, visando caracterizar e tipificar se ocorre a mobilização destes para um próximo projeto ou para o estudo do tópico relacionado ao tema de matemática selecionado.

Será também verificada a motivação do grupo em mobilizar indivíduos que se desmotivaram durante a realização do projeto.

Os registros serão feitos a partir de entrevistas durante e após a aplicação de projetos da sequência didática.

4. DESENVOLVIMENTO DO TRABALHO

Pensando na introdução de novas tecnologias em sala de aula para o ensino de matemática, buscou-se maiores informações quanto à forma utilizada por outras escolas no Brasil e em outros países para o ensino de programação como uma ferramenta no auxílio do desenvolvimento lógico-matemático dos alunos da educação básica. Tal busca se deve ao fato de que por muitas vezes é possível identificar nos alunos falta de interesse pelas disciplinas de Ciência, Tecnologia, Engenharia e Matemática por conta da abstração muitas vezes necessária para a compreensão dos assuntos.

Após a seleção da linguagem *Scratch* e da plataforma de prototipagem *Arduino*, que se deu devido ao grande número de ações envolvendo esses dois métodos e à comunidade mundial de discussão de temas sobre essas metodologias, foi feita a apresentação à turma.

4.1 APRESENTAÇÃO E DEMONSTRAÇÃO DOS MÉTODOS À TURMA

Houve boa receptividade por parte da turma em relação ao *Scratch* e ao *Arduino*, sendo que a utilização do *Arduino* se mostrou mais interessante.

As duas apresentações foram feitas de maneira expositiva, em aulas distintas e contou com a presença dos dezesseis alunos. Primeiramente foi demonstrado o programa de Introdução ao *Scratch*, seguido de uma navegação pelo portal para que os alunos visualizassem quais seriam as possibilidades de projetos a serem realizados por meio da plataforma.

O que chamou mais atenção foi que eles poderiam criar seus próprios jogos, fazendo com que isso os motivasse a aprender mais sobre o *Scratch*.

Já na apresentação do *Arduino*, somente um kit foi mostrado aos alunos e o professor realizou a montagem do projeto 1: LED piscante, que consiste em colocar um LED para piscar de maneira intermitente, mostrou a plataforma *Arduino* funcionando e projetou as linhas de código para que os alunos pudessem tentar entender como o *software* realizava o controle do *hardware*.

O controle sobre algo físico fez com que os alunos se mostrassem mais interessados e decidissem de maneira unânime pela utilização da plataforma *Arduino* como ferramenta para o estudo de matemática.

Portanto a decisão por trabalhar somente com a plataforma de desenvolvimento de projetos *Arduino* foi tomada com base na motivação dos alunos, que ficaram curiosos para realizar os projetos, havendo inclusive o pedido de um grupo de três alunos para levar os

materiais para casa afim de desenvolver projetos diferentes. Outro fator de motivação mostrou-se associado ao custo da implementação da metodologia usando o *Arduino*.

Cada kit *Arduino* custa R\$68,00 (sessenta e oito reais) e será possível trabalhar com apenas 4 computadores e grupos de 4 alunos, pois cada um deles irá dispor de um kit, conforme mencionado acima, onde realizará a montagem do *hardware* e após fará a programação e testes utilizando um computador compartilhado com o restante da equipe.

Caso optássemos pelo *Scratch*, seria mais interessante que cada aluno tivesse acesso a um computador, uma vez que não havia nenhum outro tipo de *hardware* para que eles trabalhassem enquanto aguardavam o colega terminar de utilizar o computador, fazendo com que o custo de implementação da metodologia de trabalho em sala fosse aumentado, tendo em vista que na escola objeto de estudo não há sala de informática.

4.2 A PLATAFORMA SELECIONADA E OS MATERIAIS EMPREGADOS

Buscou-se na literatura projetos possíveis de serem realizados com o kit adquirido e o livro *Arduino Básico* de Michael McRoberts (2011) chamou atenção pela forma de trabalho.

Os projetos do livro primeiramente descrevem, utilizando figuras ilustrativas e texto, a forma com que o hardware deve ser montado e logo em seguida entregam ao leitor o código que deve ser utilizado para que a placa comece a funcionar, assim o primeiro resultado aparece rapidamente e acaba por prender a atenção dos alunos.

Somente após termos colocado o protótipo em funcionamento é que vamos buscar a compreensão da programação e à partir daí realizar modificações no projeto que podem se estender de acordo com a curiosidade e habilidade dos grupos.

No livro McRoberts (2011) traz 50 projetos que vão aumentando sua complexidade e necessitando de cada vez mais dispositivos que interagem com a placa *Arduino*, como a ideia era inicialmente tratar de projetos mais simples, que podem ser alterados e ter sua complexidade aumentada com a utilização dos mesmos itens, somente foram tratados com os alunos cinco projetos do livro, que trabalham basicamente com LEDs.

A escolha se deu principalmente por haver consonância entre a forma de trabalho do livro e os conceitos de Construtivismo de Piaget, em que o professor é um facilitador da aprendizagem, o Construcionismo de Papert, no qual as máquinas devem auxiliar na construção do conhecimento.

Após a definição dos projetos foram adquiridos os materiais necessários à realização dos projetos, sendo eles: 16 kits contendo uma placa *Arduino uno*, uma *protoboard*, LEDs de várias cores e resistores de diversos valores para que cada aluno pudesse realizar a montagem do hardware individualmente, além disso foram necessários também 4 computadores, que a escola já dispunha, onde foi instalado o software *Arduino IDE*.

4.3 PREPARAÇÃO DA NOVA TURMA

Dos dezesseis integrantes da turma do primeiro ano do ensino médio de 2018 apenas dois já haviam tido algum tipo de contato com linguagens de programação, mas ambos afirmaram ser capazes de ler e compreender códigos de baixa complexidade nas linguagens C++ e Python respectivamente.

Uma prova contendo 90 questões de múltipla escolha, com cinco alternativas cada, foi aplicada em quatro horas e quarenta e cinco minutos a quinze dos dezesseis alunos da turma, sendo que um deles não pode participar da avaliação por estar afastado devido a problemas de saúde.

A aplicação da prova teve a finalidade de verificar os conhecimentos em matemática e em ciências da turma que realizaria os projetos, tendo em vista que a escola onde o trabalho foi desenvolvido não atende alunos do nono ano do ensino fundamental, e portanto, não tinha histórico próprio quanto ao desempenho dos indivíduos nessas áreas.

As questões selecionadas tiveram conteúdos que fazem parte do conteúdo pertinente ao nono ano do ensino fundamental e foram distribuídas como segue:

- 15 questões de biologia
- 15 questões de física
- 15 questões de química
- 45 questões de matemática

Na Tabela 2 verifica-se o desempenho individual dos alunos no teste.

Tabela 2 – Número de acertos por disciplina para cada aluno

ALUNO	BIOLOGIA	FÍSICA	QUÍMICA	MATEMÁTICA	TOTAL
Aluno 1	9	10	10	23	52
Aluno 2	12	5	10	20	47
Aluno 3	11	7	9	18	45
Aluno 4	8	8	9	18	43
Aluno 5	12	8	11	10	41
Aluno 6	11	6	8	15	40
Aluno 7	10	4	10	16	40
Aluno 8	7	8	9	14	38
Aluno 9	7	4	8	17	36
Aluno 10	12	5	7	11	35
Aluno 11	8	4	7	14	33
Aluno 12	8	4	4	13	29
Aluno 13	6	3	1	14	24
Aluno 14	4	7	5	8	24
Aluno 15	7	1	3	9	20
Aluno 16	Ausente				

Fonte: Do autor

É possível perceber os conhecimentos matemáticos dos alunos não é homogêneo, tendo em vista a grande variação no número de acertos. A vida escolar dos alunos antes de iniciarem os estudos na escola onde o projeto foi executado também é bastante diversificada, sendo que temos alunos provenientes de 8 escolas diferentes, cinco no município de Sinop e outras três em outras cidades.

Após essa primeira avaliação iniciou-se o trabalho com os projetos descritos por McRoberts (2011).

4.4 PROJETOS SELECIONADOS PARA IMPLEMENTAÇÃO NO ARDUINO

Com o intuito de demonstrar uma forma diferente de aprender matemática foram executados cinco projetos propostos por McRoberts (2011), sendo que os itens de hardware necessários às execuções são basicamente os mesmos para todos os projetos. Foram selecionados projetos que trabalham o uso de potências de base 10, conceitos de criptografia, organização de informações em sequência correta, uso de vetores e matrizes e funções periódicas.

4.4.1 PROJETO 1: A UTILIZAÇÃO DE POTÊNCIAS DE BASE 10

O primeiro contato da nova turma com a plataforma Arduino se deu pela realização do Projeto 1 proposto por McRoberts (2011), que consiste em colocar um LED para piscar de maneira intermitente.

Para realizar a montagem desse projeto é necessário entender e utilizar potências de base 10 para determinar qual será o resistor adequado ao circuito apresentado, além de compreender o uso dos prefixos do SI (Sistema Internacional de Unidades).

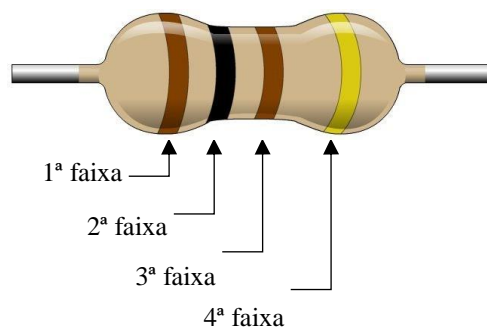
A utilização de potências de base 10 para os alunos do ensino médio muitas vezes não os aproxima de temas cotidianos, e visando trazer uma situação na qual o uso da potência de base 10 seja necessária lhes foi apresentada a Tabela 3 que contém cores e valores respectivos para o cálculo de resistores.

Tabela 3 – Tabela de cores para determinar o valor de resistores

Cor	Primeira faixa	Segunda faixa	Terceira faixa (multiplicador)	Quarta faixa (tolerância)
Preto	0	0	$\times 10^0$	
Marrom	1	1	$\times 10^1$	$\pm 1\%$
Vermelho	2	2	$\times 10^2$	$\pm 2\%$
Laranja	3	3	$\times 10^3$	
Amarelo	4	4	$\times 10^4$	
Verde	5	5	$\times 10^5$	$\pm 0,5\%$
Azul	6	6	$\times 10^6$	$\pm 0,25\%$
Violeta	7	7	$\times 10^7$	$\pm 0,1\%$
Cinza	8	8	$\times 10^8$	$\pm 0,05\%$
Branco	9	9	$\times 10^9$	
Dourado			$\times 10^{-1}$	$\pm 5\%$
Prata			$\times 10^{-2}$	$\pm 10\%$
Nenhuma				$\pm 20\%$

Fonte: Adaptada de McRoberts (2011)

De acordo com PUCRS (2001) a forma de cálculo do valor de resistores segue o exemplo da Figura 1.

Figura 1 – Resistor de $100\Omega \pm 5\%$ 

Fonte: I Am Technical (2018) – Modificada

O valor do resistor é determinado da seguinte maneira, a 1ª faixa define o valor de uma dezena, a 2ª faixa determina o valor de uma unidade, a 3ª faixa mostra o valor do expoente de uma potência de base 10 que multiplicará o valor obtido pela dezena e unidade e a 4ª faixa apresenta o limite de tolerância para o resistor.

Para esse exemplo temos marrom (1), preto (0), marrom (1) e dourado ($\pm 5\%$), portanto o valor do resistor será $10 \times 10^1 \pm 5\%$, ou seja, um resistor de valor nominal igual a 100Ω , variando entre 95Ω e 105Ω .

O projeto pede que seja utilizado um resistor de no mínimo 100Ω , e como o intuito do projeto não é a demonstração e compreensão plena da parte elétrica e eletrônica dos protótipos, somente foi explicado aos alunos que a placa Arduino funciona como uma espécie de bateria que seria ligada aos LEDs. De acordo com Arduino Uno Rev3 (2018a), a placa fornece uma voltagem igual a 5V, e o LED necessita de apenas 3V, nós iríamos precisar ligar um resistor ao circuito, que segundo McRoberts (2011) é um dispositivo que causa uma queda na voltagem, permitindo, portanto, a ligação do LED ao terminal da placa sem que ocorresse a queima de componentes.

Neste momento da montagem ocorre um impasse e o papel do professor se tornou fundamental como mediador: como não havia no kit entregue para as montagens resistores de 100Ω , os alunos tiveram que utilizar um resistor maior, no caso de 120Ω , definido pelas cores marrom, vermelho, marrom e com tolerância de 5%, conforme Figura 2. Tal decisão se deu com o professor atuando como um mediador para a turma, durante a discussão sobre qual resistor utilizar, o professor sugeriu que eles pensassem o que ocorreria caso ligassem uma lâmpada projetada para funcionar em uma rede de 127 V em um bocal de 220 V, em que a resposta foi unanime pela queima da lâmpada devido à voltagem maior, e portanto, com corrente elétrica acima da possível para a lâmpada.

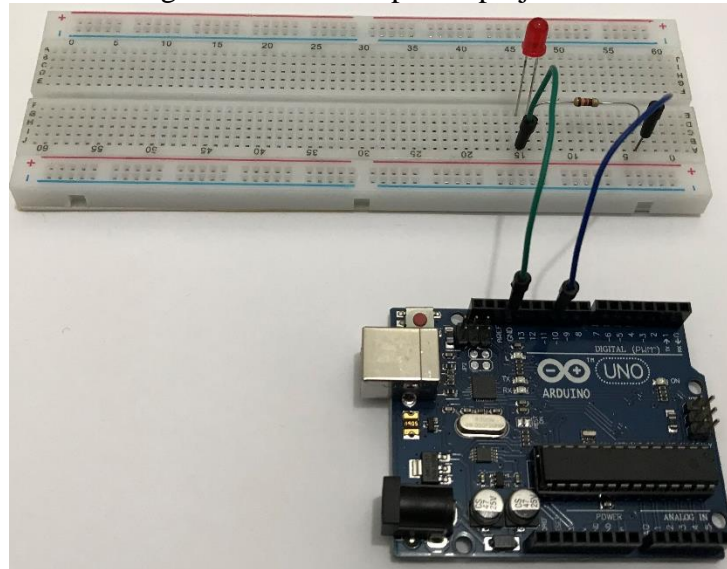
Figura 2 – Resistor de 120Ω



Fonte: Do autor

Na montagem do projeto foi necessário utilizar uma protoboard, um LED, um resistor de 120 Ω e 2 fios de ligação (*jumpers*) conectados ao *Arduino Uno*, a Figura 3 mostra o *hardware* montado.

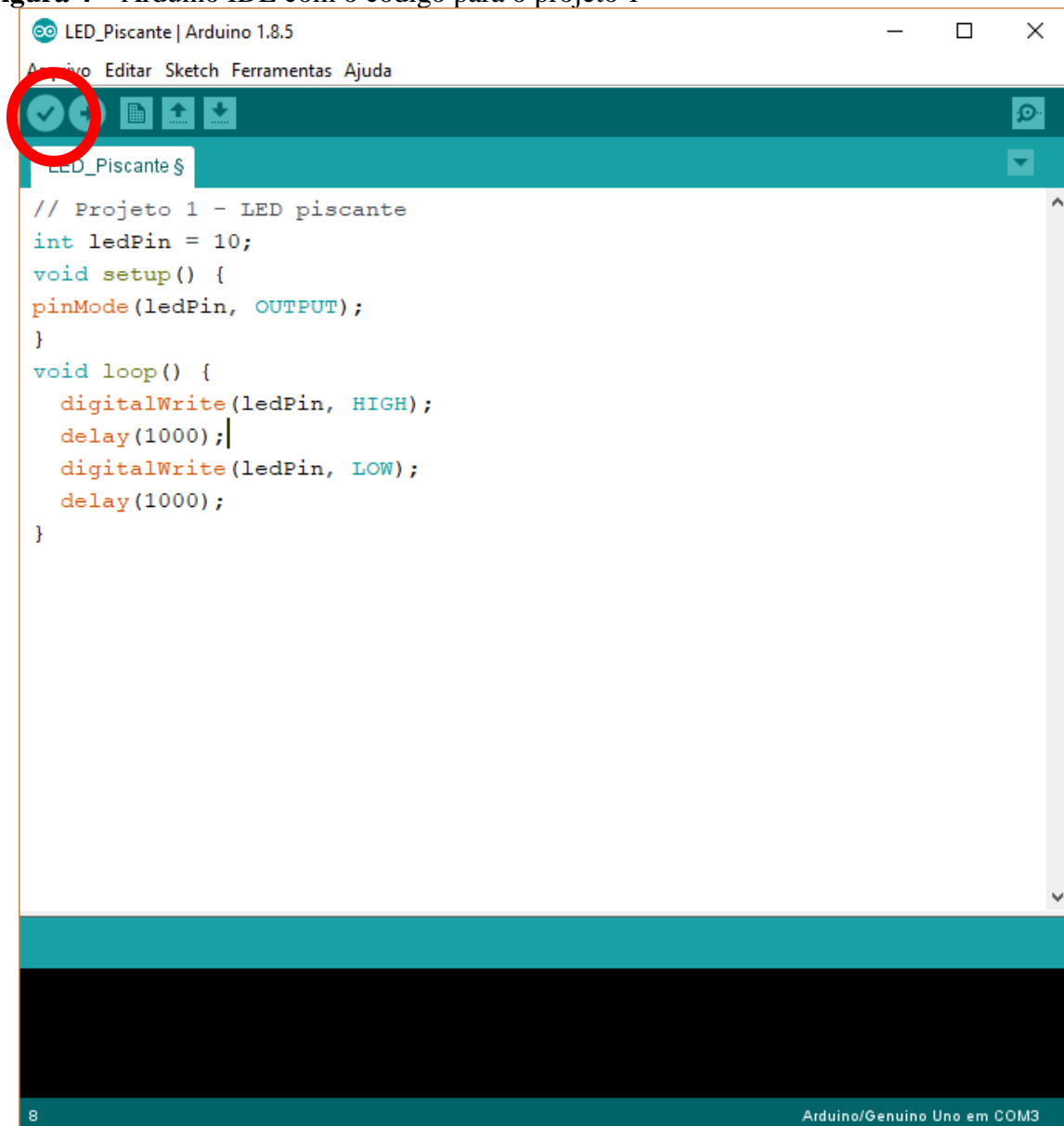
Figura 3 – Esquema de montagem do hardware para o projeto 1



Fonte: Do autor

Um *jumper* ligado ao pino GND, que significa *ground* ou terra em português, foi ligado a um ponto na *protoboard* que fez a ligação com o polo negativo do LED. A *protoboard* fez a ligação do polo positivo do LED a um resistor de 120Ω, e um *jumper* ligou o resistor ao pino de número 10 na placa *Arduino*.

Após a montagem do *hardware* os estudantes tiveram que escrever o código na plataforma, utilizando o *Arduino IDE*, conforme Figura 4, que é um *software* onde é possível escrever os códigos e realizar o *upload* para a placa. O Código 1 mostra as instruções que foram inseridas na *IDE* do *Arduino* para que o projeto fosse implementado.

Figura 4 – Arduino IDE com o código para o projeto 1

Fonte: Do autor

Após digitar ou copiar o Código 1 para o *Arduino IDE* foi necessário clicar em Carregar, que está em destaque na Figura 4. Ao clicar em carregar o *software* fez a compilação das linhas de código e realizou o upload para a placa, abaixo segue o sketch utilizado.

Código 1 – Código para o projeto 1

```
// Projeto 1 - LED piscante
int ledPin = 10;
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
```

```

    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(1000);
}

```

Fonte: McRoberts (2011)

Inicialmente a linha de código determina qual será a porta, ou *pin*, utilizado na ligação do LED, nesse caso a porta de número 10, em seguida há uma linha colocando a porta de número 10 no modo *OUTPUT*, ou saída em português, significando que o pin enviará um sinal. Tendo isso definido inicia-se o código do loop, que nesse caso se repete indefinidamente, em que o pin é definido inicialmente como *HIGH*, que é quando ele se acende e depois aguarda por 1000 milissegundos de acordo com o comando `delay(1000)`, para então entrar no estado de *LOW*, que é quando o LED se apaga e aguardar mais 1000 milissegundos. À partir daí tudo se repete.

Além de mostrar uma situação real onde é necessária a escrita de valores utilizando potências de base 10 e o uso do prefixo mili (m) do SI, o projeto1 foi utilizado na demonstração da plataforma aos alunos e consiste em um loop infinito onde um LED se acende e apaga sempre com os mesmos intervalos de tempo.

Todos os alunos foram capazes de montar seus próprios circuitos, conectá-los ao computador e testá-los, sendo que após o teste inicial lhes foi proposto alterar os intervalos com que o LED permanecia aceso e apagado.

Para realizar a tarefa foi necessário compreender o uso de prefixos do sistema internacional de unidades, já que o comando `delay` determina quantos milissegundos (ms) a placa irá aguardar para executar a próxima linha de código.

De acordo com a Tabela 4 o prefixo mili equivale a 10^{-3} , portanto para que o LED ficasse aceso por um segundo foi necessário utilizar 1000 milissegundos no comando `delay`.

Tabela 4 – Prefixos do SI

Fator	Nome	Símbolo	Fator	Nome	Símbolo
10^{24}	Yotta	Y	10^{-1}	Deci	d
10^{21}	Zetta	Z	10^{-2}	Centi	c
10^{18}	Exa	E	10^{-3}	Mili	m
10^{15}	Peta	P	10^{-4}	Micro	μ
10^{12}	Tera	T	10^{-5}	Nano	n
10^9	Giga	G	10^{-6}	Pico	p
10^6	Mega	M	10^{-7}	Femto	f
10^3	Quilo	k	10^{-8}	Atto	a
10^2	Hecto	h	10^{-9}	Zepto	z
10^1	Deca	da	10^{-10}	Yocto	y

Fonte: Adaptado de Prefixos SI e notação científica (2018)

Durante a execução do projeto a turma questionou sobre a possibilidade de incluir novos LEDs no circuito, demonstrando o interesse em aprofundar o conhecimento no tema, sugerindo, portanto, a existência de motivação por impulso cognitivo, onde os alunos se mostraram interessados em progredir e avançar com o tema.

4.4.2 PROJETO 2: USO DE CRIPTOGRAFIA E PARÂMETROS DE COMPARAÇÃO

O projeto 2 consistia em fazer um LED piscar de acordo com o código Morse, desta forma introduzimos o conceito de criptografia e foi possível a utilização de parâmetros de comparação, maior que, menor que e igual, além de realizar comandos de repetição de sequencias até que uma determinada situação seja atingida.

O projeto 2 utiliza a mesma estrutura física do projeto 1, porém com a alteração das linhas de código. Inicialmente lhes foi demonstrado o código internacional para pedido de socorro, o SOS.

Percebemos pela Figura 5 que a palavra SOS em código Morse é formada por 3 pontos, seguidos por 3 traços e novamente por 3 pontos.

Figura 5 – Alfabeto em código Morse

A ●- -	J ●- - -	S ●●●
B - ●●●	K - ● -	T -
C - ● - ●	L ● - ●●	U ●● -
D - ●●	M - -	V ●●● -
E ●	N - ●	W ● - -
F ●● - ●	O - - -	X - ●● -
G - - ●	P ● - - ●	Y - ● - -
H ●●●●	Q - - ● -	Z - - ●●
I ●●	R ● - ●	

Fonte: InfoEscola (2018).

O Código 2 é utilizado para a palavra SOS.

Código 2 – Código para o projeto 2

```
// LED conectado ao pino 10
int ledPin = 10;
// executa uma vez, quando o sketch inicia
void setup() {
  // define o pino como saída
  pinMode(ledPin, OUTPUT);
}
// executa repetidas vezes
void loop() {
  // 3 pontos
  for (int x=0; x<3; x++) {
    digitalWrite(ledPin, HIGH); // acende o LED
    delay(150); // espera 150ms
    digitalWrite(ledPin, LOW); // apaga o LED
    delay(100); // espera 100ms
  }
  // espera 100ms para marcar o intervalo entre as letras
  delay(100);
  // 3 traços
  for (int x=0; x<3; x++) {
    digitalWrite(ledPin, HIGH); // acende o LED
    delay(400); // espera 400ms
    digitalWrite(ledPin, LOW); // apaga o LED
    delay(100); // espera 100ms
  }
}
```

```
// espera 100ms para marcar o intervalo entre as letras
delay(100);
// 3 pontos novamente
for (int x=0; x<3; x++) {
    digitalWrite(ledPin, HIGH); // acende LED
    delay(150); // espera 150ms
    digitalWrite(ledPin, LOW); // apaga o LED
    delay(100); // espera 100ms
}
// espera 5 segundos antes de repetir o sinal de SOS
delay(5000);
}
```

Fonte: McRoberts (2011)

Em relação ao projeto 1 somente foram alteradas as linhas de código dentro do comando `void loop()`, pois o mesmo pino da placa Arduino foi utilizado para a ligação do LED e ele foi colocado no mesmo modo de operação, que no caso foi *OUTPUT*.

Assim como acontece em outras linguagens de programação, como a linguagem C, ao utilizar o *Arduino IDE* para programar linhas de código sempre que uma linha iniciar com os caracteres `//` isso significa que se trata apenas de uma linha explicativa para uso de quem está programando, sendo que o que seguir na mesma linha não será lido pelo compilador, a utilização desse artifício foi bastante interessante nesse projeto, pois permitiu aos alunos compreender melhor o código.

No início do *loop* temos o comando `for`, que define uma sequência de passos que se repete até que uma determinada condição seja satisfeita, no caso `for (int x=0; x<3; x++)`. O comando se divide em três partes, sendo a primeira, `int x=0`, a definição da variável de controle, `x`, e a associação da mesma ao valor zero, depois em `x<3` temos a condição para que o comando continue a ser executado, que no caso é para valores de `x` menores do que 3. Por fim tem-se o incremento da variável `x` descrito por `x++`, que significa a soma de uma unidade à variável de controle ao final do comando `for`, quando o valor de `x` não for menor do que 3 o comando se encerra e a próxima linha de código é executada.

De acordo com McRoberts (2011) a palavra SOS em código Morse é formada por três pontos, ou sinais curtos, seguidos de três traços, ou sinais longos, seguidos novamente por três pontos. Ao realizar a leitura das linhas de código pudemos perceber que o padrão adotado foi de uma espera de 150 milissegundos para um ponto, intervalo de 100 milissegundos entre um

senal e outro, 100 milissegundos entre as letras, 400 milissegundos para um traço e 5000 milissegundos antes de reiniciar a repetição da palavra.

Após a compreensão da sequência definida para a palavra SOS foi pedido aos grupos que escrevessem as iniciais da escola com o LED piscando em código Morse. A sigla da escola é PPE e de acordo com Hipercultura (2018) a letra P é formada por um ponto, seguido por dois traços e novamente um ponto, já a letra E é formada somente por um ponto.

Foi possível observar que alguns alunos utilizaram estruturas extremamente simples, mas com uma maior quantidade de linhas de código, enquanto outros buscaram uma melhor utilização da estrutura for, para realizar tarefas repetitivas. Houve também aqueles que por falta de compreensão do comando for o utilizaram para realizar uma sequência de operações que não se repetiria, segue abaixo o exemplo mais extenso seguido do mais curto.

O Código 3 mostra o código mais extenso para a sigla PPE, enquanto o Código 4 mostra a implementação mais compacta.

Código 3 – Código mais extenso para a sigla PPE

```
// LED conectado ao pino 10
int ledPin = 10;
// executa uma vez, quando o sketch inicia
void setup() {
  // define o pino como saída
  pinMode(ledPin, OUTPUT);
}
// executa repetidas vezes
void loop() {
  //letra P
  //ponto
  digitalWrite(ledPin, HIGH); // acende o LED
  delay(150); // espera 150ms
  digitalWrite(ledPin, LOW); // apaga o LED
  delay(100); // espera 100ms
  //traço
  digitalWrite(ledPin, HIGH); // acende o LED
  delay(400); // espera 400ms
  digitalWrite(ledPin, LOW); // apaga o LED
  delay(100); // espera 100ms
```

```

//traço
digitalWrite(ledPin, HIGH); // acende o LED
delay(400); // espera 400ms
digitalWrite(ledPin, LOW); // apaga o LED
delay(100); // espera 100ms
//ponto
digitalWrite(ledPin, HIGH); // acende o LED
delay(150); // espera 150ms
digitalWrite(ledPin, LOW); // apaga o LED
delay(100); // espera 100ms
// espera 100ms para marcar o intervalo entre as letras
delay(100);
//segundo P
//ponto
digitalWrite(ledPin, HIGH); // acende o LED
delay(150); // espera 150ms
digitalWrite(ledPin, LOW); // apaga o LED
delay(100); // espera 100ms
//traço
digitalWrite(ledPin, HIGH); // acende o LED
delay(400); // espera 400ms
digitalWrite(ledPin, LOW); // apaga o LED
delay(100); // espera 100ms
//traço
digitalWrite(ledPin, HIGH); // acende o LED
delay(400); // espera 400ms
digitalWrite(ledPin, LOW); // apaga o LED
delay(100); // espera 100ms
//ponto
digitalWrite(ledPin, HIGH); // acende o LED
delay(150); // espera 150ms
digitalWrite(ledPin, LOW); // apaga o LED
delay(100); // espera 100ms
// espera 100ms para marcar o intervalo entre as letras
delay(100);
//letra e

```



```

    //ponto
    digitalWrite(ledPin, HIGH); // acende LED
    delay(150); // espera 150ms
    digitalWrite(ledPin, LOW); // apaga o LED
    delay(100); // espera 100ms
// espera 5 segundos antes de repetir o sinal de SOS
delay(5000);
}

```

Fonte: Código adaptado por um estudante durante a aplicação.

Percebemos que o grupo responsável pelo Código 3 se absteve de utilizar o comando `for`, fazendo com que cada linha seja executada somente uma vez a cada loop.

No Código 4 temos uma quantidade de linhas menor para se atingir o mesmo objetivo, pois enquanto no Código 3 temos 45 linhas com instruções, já desconsiderando as linhas de comentários, e no Código 4 apenas 28 linhas de instrução.

Código 4 – Código mais curto para a sigla PPE

```

// LED conectado ao pino 10
int ledPin = 10;
// executa uma vez, quando o sketch inicia
void setup() {
// define o pino como saída
pinMode(ledPin, OUTPUT);
}
// executa repetidas vezes
void loop() {
    // 2 vezes a letra P
    for (int y=0; y<2; y++) {
        //ponto
        digitalWrite(ledPin, HIGH); // acende o LED
        delay(150); // espera 150ms
        digitalWrite(ledPin, LOW); // apaga o LED
        delay(100); // espera 100ms
        // 2 traços
        for (int x=0; x<2; x++) {
            digitalWrite(ledPin, HIGH); // acende o LED

```

```

    delay(400); // espera 400ms
    digitalWrite(ledPin, LOW); // apaga o LED
    delay(100); // espera 100ms
  }
  //ponto
  digitalWrite(ledPin, HIGH); // acende o LED
  delay(150); // espera 150ms
  digitalWrite(ledPin, LOW); // apaga o LED
  delay(100); // espera 100ms
  // espera 100ms para marcar o intervalo entre as letras
  delay(100);
}
//letra e
digitalWrite(ledPin, HIGH); // acende LED
delay(150); // espera 150ms
digitalWrite(ledPin, LOW); // apaga o LED
delay(100); // espera 100ms
// espera 5 segundos antes de repetir o sinal de SOS
delay(5000);
}

```

Fonte: Código adaptado por um estudante durante a aplicação.

O grupo responsável pelo Código 4, o mais compacto, foi o que mais questionou como a estrutura funcionava e solicitou auxílio para melhor compreensão das linhas código. Inicialmente eles haviam utilizado a estrutura *for* somente para repetir os traços da letra P, e, quando indagados sobre a possibilidade de reduzir o número de linhas, chegaram à conclusão de utilizar o comando para a letra P inteira, que aparece duas vezes na sequência, sendo que aí novamente aparece uma abordagem construtivista em que o professor atua como mediador do conhecimento e incentiva o desenvolvimento dos alunos.

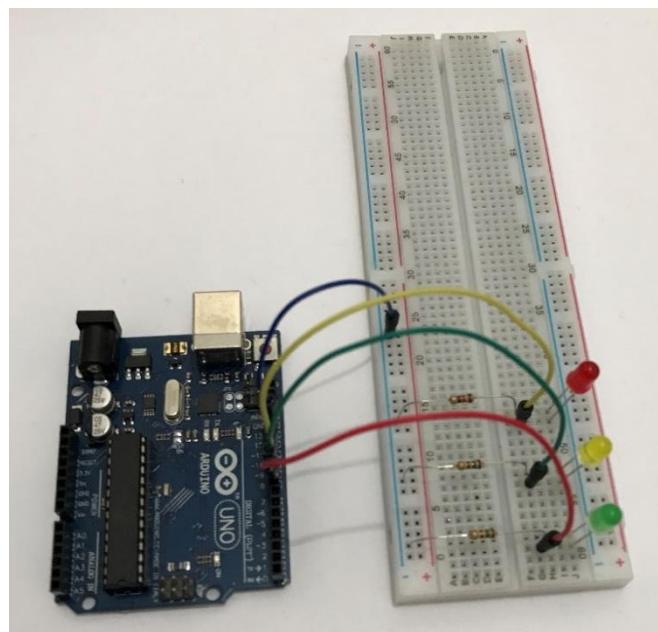
Somente um aluno dos dezesseis não se mostrou interessado em escrever as iniciais da escola utilizando código Morse, mas os próprios colegas acabaram por convencê-lo de realizar a tarefa, sugerindo, portanto, motivação aversiva, onde o indivíduo se motiva a realizar algo para evitar constrangimentos desnecessários.

4.4.3 PROJETO 3: ORGANIZANDO INFORMAÇÕES NA SEQUÊNCIA CORRETA

No projeto 3 realizamos a montagem de um sinal de trânsito, utilizando o sistema de quatro estados do Reino Unido, e para tal foi necessário fazer a ligação de 3 LEDs à placa *Arduino*, sendo que o aumento no número de LEDs ligados à placa foi algo sugerido pelos alunos já durante a execução do projeto 1.

Para tal foi necessário definir três pinos como saída, um para cada LED, e determinar quando cada um deles se acenderia e por quanto tempo, conforme Figura 6.

Figura 6 – Montagem do sinal de trânsito



Fonte: Do autor

Abaixo está demonstrado o Código 5, que foi utilizado para o projeto do sinal de trânsito.

Código 5 - O projeto do sinal de trânsito

```
// Projeto 3 - Semáforo
int ledDelay = 10000; // espera entre as alterações
int redPin = 10;
int yellowPin = 9;
int greenPin = 8;
void setup() {
    pinMode(redPin, OUTPUT);
    pinMode(yellowPin, OUTPUT);
    pinMode(greenPin, OUTPUT);
}
```

```
}  
void loop() {  
  
    digitalWrite(redPin, HIGH); // acende a luz vermelha  
    delay(ledDelay); // espera 10 segundos  
  
    digitalWrite(yellowPin, HIGH); // acende a luz amarela  
    delay(2000); // espera 2 segundos  
  
    digitalWrite(greenPin, HIGH); // acende a luz verde  
    digitalWrite(redPin, LOW); // apaga a luz vermelha  
    digitalWrite(yellowPin, LOW); // apaga a luz amarela  
    delay(ledDelay); // espera 10 segundos  
  
    digitalWrite(yellowPin, HIGH); // acende a luz amarela  
    digitalWrite(greenPin, LOW); // apaga a luz verde  
    delay(2000); // espera 2 segundos  
  
    digitalWrite(yellowPin, LOW); // apaga a luz amarela  
    // agora nosso loop se repete  
}
```

Fonte: McRoberts (2011)

Após a montagem e o teste do projeto 3 foi sugerido aos alunos que buscassem informações sobre ciclos de semáforos e que eles reproduzissem o que encontrassem utilizando a plataforma.

No padrão do Reino Unido a luz amarela se acende entre todas alterações do semáforo, seja na abertura, acendimento da luz verde, ou no fechamento, acendimento da luz vermelha, já o padrão utilizado no Brasil não acende a luz amarela para a abertura, sendo que para o sinal ficar aberto ele apaga a luz vermelha e acende a verde sem que a luz amarela seja acionada. A luz amarela fica acessa quando o sinal está indo de aberto, verde, para fechado, vermelho.

O intuito desse projeto foi demonstrar a importância da organização de informações na sequência correta para se obter os resultados desejados, pois por vezes é possível perceber que a falta de organização do raciocínio lógico-matemático dificulta na resolução dos problemas encontrados principalmente nas disciplinas de matemática e ciências.

Foi interessante observar a reação de alunos que já estiveram em outros países, no caso em tela nos Estados Unidos da América, e que nesse momento ficaram atentos e compartilharam a experiência vivida com os colegas sobre os sinais de trânsito observados durante a viagem.

4.4.4 PROJETO 4: UTILIZANDO VETORES E MATRIZES

A utilização de vetores e matrizes para a realização de tarefas auxilia na organização e sequenciamento de ações, para tal foi apresentado o projeto 4 que consiste em acender 10 LEDs em sequência.

Foi demonstrado que pelo uso de um comando que organiza valores em sequência, como uma matriz, é possível reduzir muito o número de comandos para chegar ao resultado esperado, que no caso era ter os LEDs acendendo e apagando de maneira sequencial.

Ao comparar o número de LEDs acessos no projeto 3 e no projeto 4 os alunos imaginaram uma quantidade de linhas de código bastante extensa, já que houve um aumento de três para dez LEDs, mas ao se deparar com o Código 6, fornecido por McRoberts (2011), ficaram curiosos para compreender como os comandos conseguiriam definir quando cada um dos LEDs se acenderia.

O código 6 foi utilizado para a realização desse projeto.

Código 6 – Utilizando vetores e matrizes

```
// Projeto 4 - Efeito de iluminação sequencial com LEDs
byte ledPin[] = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13}; // cria um
array para os pinos dos LEDs
int ledDelay(65); // intervalo entre as alterações
int direction = 1;
int currentLED = 0;
unsigned long changeTime; // declaração da variável changeTime
como unsigned long
void setup() {
    for (int x=0; x<10; x++) { // define todos os pinos como
saída
        pinMode(ledPin[x], OUTPUT); }
    changeTime = millis(); // define changeTime como sendo o
tempo desde o início do programa na placa Arduino
}
void loop() {
```

```

    if ((millis() - changeTime) > ledDelay) { // verifica se já
transcorreram ledDelay

                                // ms desde a última
                                alteração

        changeLED();
        changeTime = millis();
    }
}

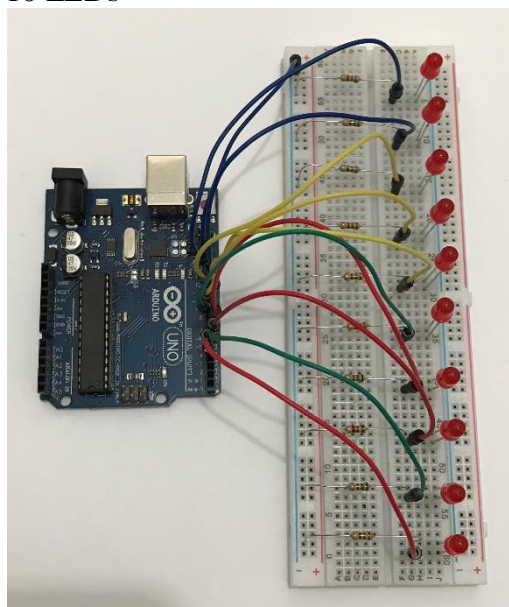
void changeLED() {
    for (int x=0; x<10; x++) {
        digitalWrite(ledPin[x], LOW); // apaga todos os LEDs
    }
    digitalWrite(ledPin[currentLED], HIGH); // acende o LED
atual
    currentLED += direction; // incrementa de acordo com o valor
de direction
    // altera a direção se tivermos atingido o fim
    if (currentLED == 9) {direction = -1;}
    if (currentLED == 0) {direction = 1;}
}

```

Fonte: McRoberts (2011)

Na Figura 7 vemos o arranjo físico do hardware utilizado no projeto.

Figura 7 – Montagem dos 10 LEDs



Fonte: Do autor

O projeto 4 foi importante para demonstrar a utilização de matrizes e para que fosse possível compreender que o fato de conhecermos novas técnicas e ferramentas matemáticas nos permite chegar ao resultado de problemas de maneiras mais simples e rápidas, algo também negligenciado pelos alunos ao serem apresentados a novas formas de resolução de exercícios.

Apesar de o aumento no número de LEDs ser algo sugerido já no início dos projetos, não houve grande motivação, comparando com os projetos anteriores, por parte dos alunos para a realização do projeto 4.

Uma possível causa para isso pode ser o fato de que até esse momento a teoria sobre matrizes e vetores ainda não havia sido trabalhado na disciplina de matemática.

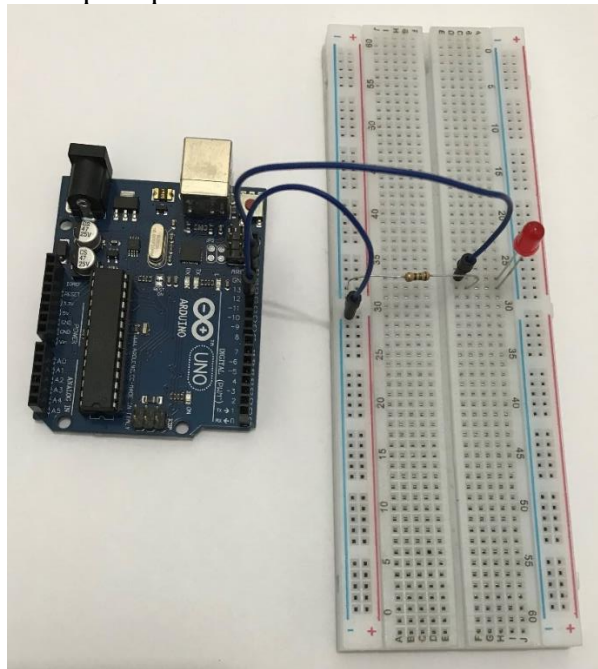
4.4.5 PROJETO 5: UTILIZANDO FUNÇÕES PERIÓDICAS

O projeto 5 consiste em alterar o nível do brilho de um LED entre o máximo e o mínimo passando por valores intermediários, sendo que até o momento somente havíamos trabalhado com a possibilidade de termos um determinado LED acesso ou apagado.

Para tal foi necessário simular uma saída analógica em um pino digital, no caso utilizamos a saída de número 11, que é do tipo *PWM* (*Pulse Width Modulation*) ou Modulação por Largura de Pulso em português, que segundo McRoberts (2011) é uma forma utilizada pela placa *Arduino* para obter resultados analógicos, que podem assumir valores diferentes de 0 e 1, por meio de portas digitais.

Sendo assim utilizamos uma função periódica, no caso em tela a função seno, variando entre 0 e 1, para determinar o nível de luminosidade do LED, que pode variar entre 0 e 255, ao multiplicar o valor do seno por 255.

Na Figura 8 temos a montagem do hardware do projeto.

Figura 8 – Montagem da lâmpada pulsante

Fonte: Do autor

O Código 7 foi utilizado junto ao arranjo físico da Figura 8.

Código 7 – Utilizando funções periódicas

```
// Projeto 5 - Lâmpada pulsante
int ledPin = 11;
float sinVal; // define a variável sinVal como sendo do tipo float
int ledVal;

void setup() {
    pinMode(ledPin, OUTPUT);
}

void loop() {
    for (int x=0; x<180; x++) {
        // converte graus para radianos e, então, obtém o valor do seno
        sinVal = (sin(x*(3.1416/180))); // calcula o valor do seno de x
        ledVal = int(sinVal*255);
        analogWrite(ledPin, ledVal);
        delay(25);
    }
}
```


}

Fonte: McRoberts (2011)

Além de compreender uma utilização para funções periódicas, foi necessário também realizar a conversão de ângulos em graus para radianos, já que a função `for` está variando entre 0 e 179 graus, que é o intervalo de ângulos em que a função seno varia de zero a um, e a função `sin` realiza o cálculo do valor do seno para ângulos em radianos.

Houve nesse momento uma adesão menor ao projeto 5 do que o verificado nos projetos anteriores. Nesse instante da execução do projeto 5 um dos grupos comentou sobre a utilização de sensores e motores com a plataforma *Arduino* e, apesar de lhes ter sido apresentado um novo projeto, a semelhança do projeto físico com os dois projetos iniciais acabou por desmotivá-los, já que eles gostariam de acrescentar peças aos projetos e não reduzi-las.

4.5 CONSIDERAÇÕES SOBRE O TRABALHO

Ao final da aplicação foi possível perceber que poderíamos alterar a sequência didática para melhorar a motivação dos alunos. Algumas alternativas seriam:

- Realizar o projeto 5 após o projeto 2, fazendo com que o nível de complexidade de montagem e o número de componentes ligados à *protoboard* fossem gradativamente aumentados.
- Além de realizar a alteração da sequência, projeto 5 após o projeto 2, alterar o projeto 3 para utilizar o conceito do LED pulsante junto do sinal de trânsito.

5. CONSIDERAÇÕES FINAIS

Apesar de termos realizado um número pequeno de projetos foi possível verificar que há viabilidade para a implementação da metodologia em sala de aula, pois mesmo não tendo um computador para cada aluno e não tendo muito tempo para discutir os projetos, percebeu-se que boa parte dos alunos se mostrou motivada a entender o que estava sendo proposto.

O ideal construtivista de se aprender interagindo com o meio e respeitando a individualidade dos alunos, permitindo por exemplo que eles realizassem as atividades de maneiras distintas, traz consigo esse engajamento.

O uso dos projetos sugeridos por McRoberts (2011) facilita o trabalho do professor como um facilitador do aprendizado, conforme sugerido por Papert (1985) em sua descrição da teoria construtivista, percebeu-se que os alunos se sentiram possuidores de suas soluções e que o domínio sobre o que a máquina executava estava em suas mãos, fazendo com alguns alunos se interessassem em levar os materiais para casa e trabalhar com projetos mais complexos do que os apresentados em sala de aula.

Ao entrevistar os alunos de maneira individual obteve-se boa receptividade da parte de todos, porém nem todos compreenderam plenamente do que se tratava o projeto.

O testemunho mais negativo foi de um aluno que diz não ter entendido como a participação nos projetos seria vantajosa para ele naquele momento, ele relatou não ter se sentido motivado a realizar as atividades por não ter pretensão de seguir nenhuma carreira na área de tecnologia e que, portanto, não chegará a utilizar tal conhecimento. Há, portanto, uma sugestão de motivação aversiva durante a execução dos projetos, já que o referido aluno não deixou de realizar nenhuma atividade durante as aulas

Em contrapartida o testemunho de uma aluna que diz querer cursar medicina veterinária foi oposto a esse, sendo que ela compreendeu que a participação nos projetos com o *Arduino* foram importantes no sentido de melhorar a compreensão de problemas matemáticos e de acordo com ela o que mais a motivou foi o fato de ela se sentir desafiada pelos projetos propostos, o que sugere motivação por impulso cognitivo e motivação por engrandecimento do ego.

Um outro aluno diz ter compreendido os projetos e sua execução, se sentiu motivado a realizá-los, mas não soube dizer exatamente o porquê desse resultado. Nesse caso específico trata-se de um aluno que normalmente realiza suas tarefas de maneira rápida e assim que o faz expõe o fato aos colegas da turma, mesmo durante aulas expositivas com a realização de exercícios.

Durante as entrevistas percebeu-se que os alunos que se sentiram menos motivados foram justamente aqueles que tiveram a menor compreensão do tema, comprovando a ideia de que a motivação para aprender está relacionada a quanto os alunos acabam absorvendo das informações que lhes são passadas.

5.1 CONTRIBUIÇÕES DO TRABALHO

A maior contribuição buscada pelo autor durante o desenvolvimento do trabalho foi a de tentar deixar claro neste estudo de caso que é possível trabalhar com esse tipo de metodologia mesmo em situações não totalmente adequadas, como com um número reduzido de computadores, poucas aulas por semana e sem ter pleno conhecimento da ferramenta. Isso se deve ao fato de termos buscado a utilização dos conceitos descritos por Seymour Papert, em que o professor atua como um mediador do conhecimento e buscar direcionar o aprendizado, ao invés de se colocar como alguém que detém o conhecimento para si e compartilha com os outros à forma que melhor lhe convier.

5.2 SUGESTÕES PARA PRÓXIMOS TRABALHOS

Pensando em alterações na forma de execução dos projetos para o futuro, aquela que parece ser a mais adequada é o aumento no número de computadores disponíveis, uma vez que o fato de termos tido todos os projetos realizados em grupo acaba por limitar a individualidade dos alunos. Justamente os mais tímidos e que têm maiores dificuldades para se expressar perante os colegas foram os que se sentiram mais isolados e menos motivados a trabalhar, sendo possível a eles trabalhar de fato somente com o hardware, que foi disponibilizado a todos individualmente.

Apesar de ser algo que possivelmente trará melhores resultados, essa ação acaba por limitar ainda mais o uso da ferramenta por outros professores pelo país, devido à dificuldade de acesso a recursos para equipar um laboratório de informática com computadores e placas *Arduino* para uma turma inteira poder trabalhar.

REFERÊNCIAS

- ARDUINO. *Arduino Uno Rev3*. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acesso em 09 abr 2018a.
- ARDUINO. *What is Arduino?*. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction#>>. Acesso em 27 jan 2018b.
- AZENHA, M. G. *Construtivismo: de Piaget a Emília Ferreiro*. São Paulo: Ática, 1994.
- DAILY PAPERT. *About Seymour Papert*. Disponível em: <<http://dailypapert.com/about-seymour-papert/>>. Acesso em: 10 abr 2018.
- DICIO. *Motivação – Dicio, Dicionário Online de Português*. Disponível em <<https://www.dicio.com.br/motivacao/>>. Acesso em 28 de abr 2018.
- GODOI, C. K.; *Categorias da motivação na aprendizagem*. Florianópolis, Universidade Federal de Santa Catarina, 2001.
- HIPERCULTURA. *Mas afinal para que serve o código Morse?*. Disponível em <<https://www.hipercultura.com/codigo-morse/>>. Acesso em 09 abr 2018.
- I AM TECHNICAL. *100Ω Resistor Color Code*. Disponível em: <<http://www.iamtechnical.com/100-ohm-resistor-color-code>>. Acesso em: 09 abr 2018.
- INEP. *Resultados – INEP*. Disponível em: <<http://inep.gov.br/web/guest/acoes-internacionais/pisa/resultados>>. Acesso em 16 de mai 2018.
- INFOESCOLA. *Código Morse – Comunicação – InfoEscola*. Disponível em: <<https://www.infoescola.com/comunicacao/codigo-morse/>>. Acesso em 20 mai 2018.
- MARJI, M.; *Aprenda a programar com scratch: uma introdução visual à programação com jogos, arte, ciência e matemática*. São Paulo, Novatec Editora, 2014.
- MCROBERTS, M.; *Arduino básico*. São Paulo, Novatec Editora, 2011.
- MEDEIROS, T. J.; SILVA, T. R.; ARANHA, E. H. S.; *Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura*. *RENOTE - Revista Novas Tecnologias na Educação*, Rio Grande do Sul, v. 11, n. 3, dezembro, 2013.
- OECD. *PISA*. Disponível em: <<http://www.oecd.org/pisa/>>. Acesso em 04 de mai 2018.
- OPEN SOURCE INITIATIVE. *The open source definition*. 2007. Disponível em: <<https://opensource.org/osd>>. Acesso em 27 jan 2018.
- OUAHBIA, I. et al. *Learning Basic Programming Concepts By Creating Games With Scratch Programming Environment*. *Procedia - Social and Behavioral Sciences*, 2015.
- OZORAN, D.; CAGILTAY, N. E.; TOPALLI, D. *Using scratch in introduction to programming course for engineering students*. *Educational Technologies & Distance Education in Engineering*, 2nd International Engineering Education Conference. Antalya, 2012.
- PAPERT, S. *Logo: Computadores e educação*. São Paulo, Editora brasiliense S.A., 1985.

PREFIXOS SI E NOTAÇÃO CIENTÍFICA. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/1118013/mod_resource/content/0/PrefixosSI%2BNotacaoCientifica.pdf>. Acesso em 09 abr 2018.

PUCRS. *Código de Cores Para Resistores*. Disponível em: <https://www.inf.pucrs.br/~calazans/undergrad/laborg/cod_cores_res.html>. Acesso em 09 abr 2018.

RUBIO, M. A.; HIERRO, C. M.; PABLO, A. P. M.; *Using arduino to enhance computer programming courses in science and engineering*. Proceedings of EDULEARN13 Conference, Barcelona, julho, 2013.

SCRATCH. *About Scratch*. Disponível em: <<https://scratch.mit.edu/about>>. Acesso em 27 jan 2018.

SILVA, J. L. S. et al. *Plataforma Arduino integrado ao PLX-DAQ: Análise e aprimoramento de sensores com ênfase no LM35*. XIV Escola Regional de Computação Bahia, Alagoas e Sergipe (ERBASE). Feira de Santana, BA, 2014.

SOUZA, A. R. et al. *A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC*. In: Revista Brasileira de Ensino de Física, v. 33, n. 1, 2011.

TUCKER, A.; *A Model Curriculum for K–12 Computer Science: Final Report of the ACM K–12 Task Force Curriculum Committee*. Nova Iorque, 2003.