

**UNIVERSIDADE FEDERAL DO PAMPA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENSINO DE CIÊNCIAS**

JANUÁRIO DIAS RIBEIRO

**EXPLORANDO AS POSSIBILIDADES DE INSERÇÃO DA PLATAFORMA
ARDUINO NO ENSINO DE CIÊNCIAS DA EDUCAÇÃO BÁSICA**

**Bagé
2018**

JANUÁRIO DIAS RIBEIRO

**EXPLORANDO AS POSSIBILIDADES DE INSERÇÃO DA PLATAFORMA
ARDUINO NO ENSINO DE CIÊNCIAS DA EDUCAÇÃO BÁSICA**

Dissertação apresentada ao Programa de Pós-Graduação Stricto Sensu em Ensino de Ciências da Universidade Federal do Pampa, como requisito parcial para a obtenção do Título de Mestre Profissional em Ensino de Ciências.

Orientador: Prof. Dr. Pedro Fernando Teixeira Dorneles

Coorientadora: Profa. Dra. Diana Paula Salomão de Freitas

**Bagé
2018**

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais).

R484e

Ribeiro, Januário Dias
EXPLORANDO AS POSSIBILIDADES DE INSERÇÃO DA
PLATAFORMA ARDUINO NO ENSINO DE CIÊNCIAS DA EDUCAÇÃO
BÁSICA / Januário Dias Ribeiro.

168 p.

Dissertação (Mestrado) -- Universidade Federal do Pampa, MESTRADO
PROFISSIONAL EM ENSINO DE CIÊNCIAS, 2018.

"Orientação: Pedro Fernando Teixeira Dorneles".

1. Plataforma Arduino. 2. Ensino de Programação. 3. Ensino de Eletrônica.
4. Ensino de Ciências. I. Título.

JANUÁRIO DIAS RIBEIRO

**EXPLORANDO AS POSSIBILIDADES DE INSERÇÃO DA PLATAFORMA
ARDUINO NO ENSINO DE CIÊNCIAS DA EDUCAÇÃO BÁSICA**

Dissertação apresentada ao Programa de Pós-Graduação Stricto Sensu em Ensino de Ciências da Universidade Federal do Pampa, como requisito parcial para a obtenção do Título de Mestre Profissional em Ensino de Ciências.

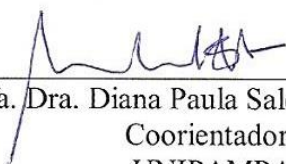
Área de concentração: Ensino de Ciências.

Dissertação defendida e aprovada em: 22 de dezembro de 2017.

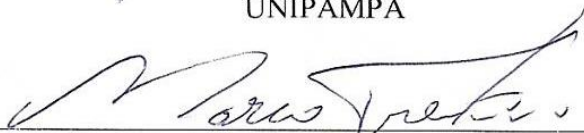
Banca examinadora:



Prof. Dr. Pedro Fernando Teixeira Dorneles
Orientador
UNIPAMPA



Profa. Dra. Diana Paula Salomão de Freitas
Coorientadora
UNIPAMPA



Prof. Dr. Marco Antônio Sandini Trentin
UPF



Prof. Dr. Paulo Henrique Guadagnini
UNIPAMPA

*Dedico este trabalho a meus pais, João Antônio
Ribeiro (in memoriam) e Maria de Lurdes Ribeiro.*

*E à minha noiva, Sharon Guedes, grande inspiração
em todos os momentos, pois sem o seu apoio e
incentivo nada seria possível.*

AGRADECIMENTOS

A meus pais, que, apesar de todas as dificuldades, nunca me deixaram faltar o necessário para que eu chegasse até aqui.

À minha noiva Sharon Geneviéve Araujo Guedes, por ser minha grande incentivadora. Pela compreensão nos momentos distantes que o trabalho ocasionou. Pelo amor e carinho que me dedica. Minha amiga, futura esposa e grande responsável por minha trajetória.

Ao meu orientador Prof. Dr. Pedro Fernando Teixeira Dorneles, pela amizade de todos esses anos e pela ajuda efetiva e indispensável na elaboração deste trabalho. Sempre disposto a auxiliar nos momentos de incertezas, sempre me motivando a acreditar no meu potencial, desde minha graduação.

À minha coorientadora Profa. Dra. Diana Paula Salomão de Freitas, pela ajuda e dedicação, contribuindo com excelentes sugestões, sempre com bom humor e gentileza que lhe são características.

Aos meus colegas TAEs da UNIPAMPA, em especial a todos da sala 3110, pelos momentos descontraídos e pelo incentivo para que eu realizasse este trabalho. São os “melhores colegas que você respeita”.

Aos meus parceiros de pedal, em especial ao Adair “Alemão”, amigo de longa data e grande entusiasta da Ciência, que muito me ensinou sobre eletrônica. Com certeza, um pouco de nossas longas conversas está aqui.

A meus ex-colegas do Laboratório Camilo Gomes, em especial à Ana Ojeda e Beatriz Oliveira que muito me ajudaram para que eu conseguisse me formar em Física. Com quem também aprendi muito.

A todos os colegas e professores do Mestrado Profissional em Ensino de Ciências que fizeram parte da minha formação.

A todos meus amigos e familiares que de uma forma ou de outra contribuíram na minha trajetória.

Muito Obrigado!

RESUMO

O presente trabalho é o resultado de uma intervenção pedagógica realizada em duas etapas denominadas Estudo Piloto e Curso para Professores e Profissionais da Educação, entre os meses de maio e outubro de 2016, com alunos do segundo ano do Ensino Médio, professores e outros profissionais que trabalham na educação formal. O objetivo da pesquisa foi avaliar as possibilidades de inserção da Plataforma Arduino no Ensino de Ciências da Educação Básica através de um enfoque nos conceitos introdutórios de lógica de programação e montagens de circuitos eletrônicos. A pesquisa faz uma avaliação da aprendizagem embasada na Teoria da Aprendizagem Significativa de David Ausubel e verifica duas condições importantes que são: a construção de um material instrucional potencialmente significativo e a predisposição dos aprendizes para incorporar um novo conhecimento. A fundamentação metodológica utilizada foi do tipo intervenção pedagógica, na acepção de Damiani et al., com abordagem predominantemente qualitativa e está dividida em: método de intervenção pedagógica, composta por uma descrição detalhada sobre a intervenção e o método de avaliação da intervenção, que detalha como ocorreu o processo de intervenção através da análise dos dados e dos instrumentos utilizados para obtenção destes. O Estudo Piloto contou com a participação de alunos do segundo ano do Ensino Médio e o Curso para Professores e Profissionais da Educação com dez participantes. A descrição detalhada dos encontros realizados na intervenção pedagógica; a verificação dos objetivos de aprendizagem e os achados da intervenção propriamente dita, conforme discutido por Damiani et al., compõe os resultados e discussões. A experiência didática realizada nas duas etapas ainda integra a elaboração de um produto educacional que poderá servir de base para outros professores em atividades introdutórias sobre a Plataforma Arduino e dos conceitos que ela demanda, seja para a aquisição automática de dados em experimentos de baixo custo ou para formação de professores que necessitam de subsídios para começar a trabalhar com programação, eletrônica ou automação na Educação Básica. Conclui-se ao final deste estudo que as atividades desempenhadas têm forte influência dos conhecimentos prévios relacionados à lógica de programação e ao entendimento de circuitos elétricos. Sem isso, as práticas utilizando tecnologias baseadas na integração de programação e eletrônica continuarão a serem reproduzidas nas escolas, mas pouco compreendidas. Como perspectivas futuras, é apresentado um conjunto de onze recomendações para embasarem novas investigações, principalmente as que tenham como cerne de pesquisa os meios de inserir conceitos introdutórios de programação e eletrônica em currículos da Educação Básica.

Palavras-chave: Plataforma Arduino; Ensino de Programação; Ensino de Eletrônica, Ensino de Ciências.

ABSTRACT

The present work is the result of a pedagogical intervention carried out in two stages called Pilot Study and Course for Teachers and Professionals of Education, between May and October 2016, with second year high school students, teachers and other professionals who formal education. The objective of the research was to evaluate the possibilities of insertion of the Arduino Platform in the Teaching of Sciences of Basic Education through a focus on the introductory concepts of programming logic and assemblies of electronic circuits. The research evaluates learning based on David Ausubel's Significant Learning Theory and verifies two important conditions: the construction of a potentially significant instructional material and the predisposition of learners to incorporate new knowledge. The methodology used was pedagogical intervention, in the sense of Damiani et al., With a predominantly qualitative approach and is divided into: a pedagogical intervention method, composed of a detailed description of the intervention and the evaluation method of the intervention, which details as the intervention process occurred through the analysis of the data and the instruments used to obtain them. The pilot study counted on the participation of second year high school students and Course for Teachers and Professionals of Education with ten participants. The detailed description of the meetings held in the pedagogical intervention; verification of learning objectives and the findings of the intervention itself, as discussed by Damiani et al., compose the results and discussions. The didactic experience realized in the two stages still integrates the elaboration of an educational product that can be the basis for other teachers in introductory activities on the Arduino Platform and the concepts that it demands, either for the automatic acquisition of data in experiments of low cost or for training teachers who need subsidies to start working with programming, electronics or automation in Basic Education. It is concluded at the end of this study that the activities performed have a strong influence of the previous knowledge related to the logic of programming and to the understanding of electric circuits. Without this, practices using technologies based on the integration of programming and electronics will continue to be reproduced in schools, but little understood. As a future perspective, a set of ten recommendations is presented to support new research, especially those that have as a background the means to introduce introductory concepts of programming and electronics in curricula of Basic Education.

Keywords: Arduino Platform; Programming Teaching; Teaching of Electronics, Teaching of Sciences.

LISTA DE FIGURAS

Figura 1 – Laboratório de Física da UNIPAMPA onde ocorreram os encontros presenciais do CPPE.....	29
Figura 2 - <i>Kit</i> Arduino	31
Figura 3 – Planilha eletrônica do Microsoft Excel com as fórmulas utilizadas na simulação das portas digitais.	38
Figura 4 – Fórmulas utilizadas para cada tipo de porta digital.....	39
Figuras 5a e 5b – Circuito digital representado por células com fórmulas. Saídas: falsa (A) e verdadeira (B).	40
Figura 6 - Relação entre o operador lógico, porta digital e circuito elétrico correspondente. .	41
Figura 7 – Janela do IDE com o código exemplo.	43
Figura 8 - Imagem do computador do Aluno 6 com o código (à esquerda) e a janela do monitor serial (à direita) mostrando a temperatura nas três escalas.	48
Figura 9 – Imagens das atividades na escola.	51
Figura 10 – Sequência de passos do exemplo “ <i>blink</i> ”.	62
Figura 11 – Sequência de passos para ligar o LED (atuador) com botão (sensor). Agora não depende apenas da programação feita, mas também do estado do sensor (botão externo).	63
Figura 12 – Circuito da questão 1.....	65
Figura 13 – Circuito da questão 3.....	65
Figura 14 – Questão 2 do questionário 1.	66
Figura 15 – Circuito lógico da questão 4 contendo duas portas AND e uma porta OR.....	66
Figura 16 – Código da questão 5 contendo quatro erros.	67
Figura 17 – Circuito divisor de tensão.....	69
Figura 18 – Divisor de tensão com LDR. A variação da intensidade de luz varia a tensão V_{out}	70
Figura 19 – Circuito típico para medir temperatura com o LM35.	73
Figura 20 – Circuito para medir temperatura utilizando uma bateria e um voltímetro.	74
Figura 21 – Circuitos da questão 1.	77
Figura 22 – Circuito digital da questão 2.	77
Figura 23 – Divisor de tensão com LDR e resistor fixo em duas configurações possíveis.....	78
Figura 24 – Código para responder as questões 4, 5 e 6.	79
Figura 25 – Questão 6 do questionário 2.....	80
Figura 26 – Questão 8 do questionário 3 (<i>online</i>).	82

Figura 27 – LDR (1); LED (2); sensor de umidade do solo (3); relé (4); válvula solenoide de máquina de lavar (5) e placa Arduino (6).....	87
Figura 28 – À esquerda, os primeiros testes com os sensores e à direita a maquete da horta construída pelos alunos.....	87
Figura 29 – Janela de programação do S4A com o código em blocos do sistema de irrigação feito pelos alunos.	88
Figura 30 – Garrafas contendo terra para demonstração utilizando sensor de umidade ligado ao Arduino.	88
Figura 31 – Alunos apresentando o trabalho à comunidade escolar.	89
Figura 32 – Painel solar fotovoltaico.....	90
Figura 33 – Esquema eletrônico do circuito utilizado para medir tensão e potência elétrica do painel solar. A tensão elétrica era medida pelo divisor de tensão ligado à porta A0, e as portas de 8 a 12 acionavam os relés via transistores Q1 a Q5.....	91
Figura 34 – Circuito montado com relés para adicionar carga sequencialmente.	92
Figura 35 – Painel solar no pátio da escola direcionado para o Sol.	92
Figura 36 – Gráfico da medida de potência elétrico do painel em 5 min.....	93
Figura 37 – Gráfico da medida de potência elétrico do painel em 100 min.....	93
Figura 38 – Gráfico das pontuações dos PPEs comparando com os valores médios, máximos e mínimos.	100
Figura 39 – Gráfico da pontuação de cada objetivo.	102

LISTAS DE QUADROS

Quadro 1 - Distribuição dos dias e horas presenciais dos encontros do CPPE realizado.	29
Quadro 2 – Resumo das Atividades do Estudo Piloto.....	54
Quadro 3 – Distribuição dos PPEs em relação à formação e local de atuação.....	56
Quadro 4 – Respostas por PPE da questão 7 e 8.	83
Quadro 5 – Avaliação dos PPEs dentro dos objetivos de aprendizagem	97
Quadro 6 – Facilidades e dificuldades encontradas no CPPE.....	106

LISTA DE TABELAS

Tabela 1 - Respostas por aluno do questionário 1.	45
Tabela 2 - Respostas individuais e por duplas, referente ao questionário 2.	50
Tabela 3 – Respostas por PPE do questionário 1.	65
Tabela 4 – Respostas por PPE do questionário 2.	76
Tabela 5 – Respostas por PPE do questionário 3 (questões de 1 a 6).	81
Tabela 6 – Pontuação dos PPEs com relação às classificações de aprendizagem (7 menor pontuação e 28 a maior pontuação possível).	99
Tabela 7 – Pontuação dos objetivos de aprendizagem com relação às classificações de aprendizagem (9 menor pontuação e 36 a maior pontuação possível).	101

LISTA DE ABREVIATURAS E SIGLAS

AC	Atingiu Completamente
AL	Atingiu Largamente
AM	Atingiu Minimamente
AP	Atingiu Parcialmente
CNC	Comando numérico computadorizado
CPPE	Curso para Professores e Profissionais da Educação
CRE	Coordenadoria Regional de Educação
DC	<i>Direct Current</i> (corrente contínua)
EP	Estudo Piloto
Eq.	Equação
Fig.	Figura
GND	<i>Ground</i> (terra do circuito)
HTML	<i>HyperText Markup Language</i> (Linguagem de Marcação de Hipertexto)
IDE	<i>Integrated Development Environment</i> (Ambiente de Desenvolvimento Integrado)
LDR	<i>Light Dependent Resistor</i> (resistor dependente de luz)
LED	<i>Light Emitting Diode</i> (diodo emissor de luz)
LM35	Sensor de temperatura formado por circuito integrado
p.	Página
PC	Parcialmente correta
PI	Parcialmente incorreta
PIBID	Programa Institucional de Bolsas de Iniciação à Docência
PPE	Professores e Profissionais da Educação
S4A	<i>Scratch for Arduino</i>
SMED	Secretaria Municipal de Educação de Bagé
TAS	Teoria da Aprendizagem Significativa
TC	Totalmente correta
TI	Totalmente incorreta
TIC	Tecnologias de Informação e Comunicação
UNIPAMPA	Universidade Federal do Pampa

LISTA DE SÍMBOLOS

A	Ampère (intensidade de corrente elétrica)
mV	milivolt (submúltiplo do Volt)
°C	Graus Celsius
V	Volt (unidade de diferença de potencial elétrico)
V _{in}	Tensão de entrada
V _{out}	Tensão de Saída
W	Watts (unidade de potência)
Ω	Ohm (unidade de resistência elétrica)
&&	Símbolo do operador lógico AND na linguagem de programação do Arduino
	Símbolo do operador lógico OR na linguagem de programação do Arduino

SUMÁRIO

1 INTRODUÇÃO	17
2 REFERENCIAL TEÓRICO	21
2.1 TEORIA DA APRENDIZAGEM SIGNIFICATIVA (TAS).....	21
2.1.1 Aprendizagem Mecânica	21
2.1.2 Organizadores Prévios.....	22
2.1.3 Condições para Ocorrência da Aprendizagem Significativa.....	22
2.1.4 Diferenciação Progressiva e Reconciliação Integrativa	22
3 ESTUDOS RELACIONADOS	24
4 PROCEDIMENTOS METODOLÓGICOS	27
4.1 MÉTODO DE INTERVENÇÃO PEDAGÓGICA	27
4.1.1. Estudo Piloto (EP)	27
4.1.2 Curso para Professores e Profissionais da Educação (CPPE)	27
4.1.2.1 <i>Módulo I: Formação</i>	28
4.1.2.2 <i>Módulo II: Aplicação nas Escolas</i>	32
4.2 MÉTODO DE AVALIAÇÃO DA INTERVENÇÃO.....	32
4.2.1 Instrumentos de Coleta de Dados	34
5 RESULTADOS E DISCUSSÕES	36
5.1 ESTUDO PILOTO (EP).....	36
5.1.1 Primeiro Encontro (09 mai 2016).....	37
5.1.2 Segundo Encontro (16 mai 2016).....	41
5.1.3 Terceiro Encontro (23 mai 2016)	43
5.1.4 Quarto Encontro (30 mai 2016).....	46
5.1.5 Quinto Encontro (06 jun 2016).....	47
5.1.6 Sexto Encontro (20 jun 2016).....	49
5.1.7 Considerações sobre o Estudo Piloto: achados relativos à intervenção propriamente dita	52
5.2 CURSO PARA PROFESSORES E PROFISSIONAIS DA EDUCAÇÃO (CPPE).....	55
5.2.1 Modulo I - Formação.....	55
5.2.1.1 <i>Perfil dos Participantes</i>	55
5.2.1.2 <i>Primeiro Encontro (17 set 2016)</i>	58
5.2.1.3 <i>Segundo Encontro (24 set 2016)</i>	63
5.2.1.4 <i>Terceiro Encontro (08 out 2016)</i>	68
5.2.1.5 <i>Grupo Focal 1</i>	71

5.2.1.6 Quarto Encontro (22 out 2016).....	72
5.2.1.7 Grupo Focal 2	75
5.2.1.8 Atividade a Distância: Questionário 3 (Online)	80
5.2.2 Modulo II – Aplicação nas Escolas	86
5.2.2.1 Aplicação do PPE 9.....	86
5.2.2.2 Aplicação do PPE 8.....	90
5.2.3 Considerações sobre o CPPE: achados relativos aos efeitos da intervenção sobre seus participantes.....	94
5.2.4 Considerações sobre o CPPE: achados relativos à intervenção propriamente dita	102
6 REFLEXÕES FINAIS	109
REFERÊNCIAS	113
APÊNDICE A – Cartaz de divulgação do curso	115
APÊNDICE B – Apostila utilizada no curso.....	116
APÊNDICE C – Questionário 1	149
APÊNDICE D – Questionário 2.....	151
APÊNDICE E – Formulário online de inscrição	154
APÊNDICE F – Questionário 3 (online)	159
APÊNDICE G – Código de programação para medir o painel solar	163
APÊNDICE H – Anotações dos participantes sobre o Curso	166
ANEXO A – Pôster feito pelo PPE 9 e seus alunos.....	167
ANEXO B – Termo de consentimento livre e esclarecido	168

1 INTRODUÇÃO

O mundo atual está cada vez mais rodeado de informação e tecnologia. “[...] o surgimento de um novo tipo de sociedade tecnológica é determinado principalmente pelos avanços das tecnologias digitais de informação e comunicação e pela microeletrônica” (KENSKI, 2007, p. 22). Considerando as diferentes tecnologias conhecidas, podemos buscar exemplos mais atualizados para trabalhar Ciências com os alunos na escola. Entendemos que uma forma de motivar o estudante é aproximar o Ensino de Ciências ao seu contexto. A tentativa de envolver e mobilizar os estudantes no trabalho educativo pode despertar o seu engajamento em sala de aula. Por isso, alguns tópicos que se aproximam mais das atuais tecnologias, podem facilitar a motivação e, conseqüentemente, a aprendizagem, pois coloca o aluno frente ao que ele vê e conhece, mas que poucas vezes associa ao que está sendo trabalhado em sala de aula. Conforme Kenski (2007), para as tecnologias trazerem alterações no processo educativo, devem ser compreendidas e incorporadas pedagogicamente. Sabemos que essas tecnologias já estão presentes nas escolas, como: computadores, *tablets*, celulares, etc. O que nos perguntamos é: os professores estão preparados para utilizar isso em benefício de melhorias na sua prática docente? O computador nas aulas está sendo usado de maneira a ser um diferencial na aprendizagem ou se limita a um mero editor de textos ou “pesquisador” de informação através de sites de busca? Com base nessas questões relacionadas ao uso de Tecnologias de Informação e Comunicação (TIC) em sala de aula, buscaremos, neste trabalho, desenvolver atividades que incentivem os professores a empregarem novos métodos de ensino utilizando a Plataforma Microcontrolada Arduino¹, com a qual podemos criar uma variedade de projetos educacionais dando ênfase à programação e automação de sistemas através do uso de sensores e atuadores integrados aos conceitos de Física, como o estudo de alguns tópicos de eletricidade.

Durante a formação de Licenciado em Física me deparei² com dificuldades no ensino de diversos conceitos, entre eles os de eletricidade. Ao mesmo tempo em que constatei o potencial da utilização dos computadores, como, por exemplo, das simulações computacionais e outras formas de interagir com os alunos de maneira diferente da tradicional. Vivenciei, também, algumas experiências com o uso da programação no ensino de Física. Tudo isso foi

¹ Plataforma de prototipagem eletrônica *open-source* que se baseia em hardware e software flexíveis e fáceis de usar. Disponível em: <<http://playground.arduino.cc/Portugues/HomePage>> (Acesso em: 14 nov. 2017).

² Para dar sentido e clareza ao texto, o autor escreve a introdução em primeira pessoa. A partir do próximo capítulo se mantém em terceira pessoa.

uma importante contribuição para elaborarmos as primeiras ideias que originaram este trabalho.

Trabalhando desde o ano 2014 como Técnico de Laboratório e atuando diretamente no Laboratório de Instrumentação em Ensino de Física do curso de Licenciatura em Física da Universidade Federal do Pampa, observo que há um diferencial na formação inicial de professores quando ele se utiliza de tecnologias em sala de aula, pois os licenciandos levam experiências diversas aos alunos do Ensino Básico, fazendo uma importante interação Universidade-Escola. Assim, proporcionando um contato direto entre professores e alunos, além da motivação em ensinar e aprender Física pelas mais variadas formas, principalmente com o uso de recursos tecnológicos atuais. Os professores que não fizeram uso destes recursos em suas formações acadêmicas, muitas vezes, se encontrarão entre o querer fazer algo diferente e a falta de suporte ou conhecimento. A tentativa desses professores, de elaborar uma aula mais interessante e motivadora, utilizando tecnologias, muitas vezes esbarrara na falta de conhecimento e pouco tempo para aprender coisas novas. A falta de conhecimento e tempo pode ser a causa de muitas práticas utilizando Arduino e projetos de eletrônica serem apenas reproduzidas de forma idêntica por alunos e professores. Na maioria das vezes, não há preocupação com o entendimento de códigos de programação e de circuitos eletrônicos, no máximo, os estudantes modificam códigos prontos, copiados de sites³ da *internet*. Se esses projetos fossem construídos desde o início e entendidos de forma mais ampla, os resultados dessas práticas poderiam propiciar outras potencialidades, tais como criatividade, raciocínio lógico, planejamento e habilidade para montar experimentos.

O ensino de iniciação à programação e eletrônica pode ser uma oportunidade de modificar esse quadro de discrepância entre as tecnologias atuais e o Ensino de Ciências. Nesse sentido, Rosado e Carmona (2005) argumentam que há razões epistemológicas e didáticas que justificam a inclusão da eletrônica nos currículos de Ciências, para que os alunos confrontem com êxito as mudanças de uma sociedade científica e tecnologicamente avançada. Da mesma forma, outros autores já reconhecem a necessidade de introduzir conceitos de Lógica de Programação nas séries iniciais, já que a mesma pode auxiliar no desenvolvimento cognitivo das crianças. A programação desenvolve o raciocínio lógico e

³ Alguns dos sites mais acessados por usuários do Arduino como: <<http://labdegaragem.com/>>; <<https://playground.arduino.cc/Portugues/Learning>>; <<https://br-arduino.org/>> entre outros, trazem exemplos de códigos e montagens de projetos detalhados que muitas vezes são reproduzidos por alunos. (Acesso em: 18 nov. 2017).

matemático, relação entre causa e consequência, entre outros. A programação aliada à compreensão da lógica desenvolve o pensamento computacional (ARAÚJO et al., 2015).

Frente ao exposto, o objetivo principal deste trabalho é argumentar a cerca de possibilidades do uso da plataforma Arduino na Educação Básica, a partir da investigação de uma intervenção pedagógica realizada na forma de um Estudo Piloto (EP) e de um Curso de Formação de Professores e Profissionais da Educação (CPPE). Também será elaborado, como resultado da intervenção realizada, um produto educacional contendo teoria, atividades práticas e exemplos de projetos com intuito de servir de auxílio a professores da Educação Básica que pretendem desenvolver atividades de programação e automação com a plataforma Arduino, mas que não possuem muita experiência.

O estudo foi embasado na Teoria da Aprendizagem Significativa (TAS) de David Ausubel (2003) que leva em conta a forma de interação entre o conhecimento prévio do aluno e uma nova informação apresentada. No âmbito motivacional, esperamos que, a partir de nossas atividades, vinculando teoria e prática, o aluno possa alcançar uma das condições necessárias para haver a aprendizagem. Segundo Moreira (1999) o aprendiz deve manifestar predisposição para relacionar o novo conhecimento a sua estrutura cognitiva. De outra forma ele apenas irá memorizar o conteúdo de maneira mecânica. Buscamos, também, desenvolver um material instrucional que seja potencialmente significativo, ou seja, que leve em conta o conhecimento prévio do aprendiz. Obtendo essas duas condições preconizadas na TAS, esperamos que este trabalho mostre o potencial educacional que acreditamos ter os conteúdos de programação e automação na aprendizagem de alunos da Educação Básica.

Este trabalho é dividido em duas etapas principais envolvendo atividades de ensino sobre programação e eletrônica, visando à busca por subsídios à iniciação do uso da Plataforma Arduino. Na primeira etapa desenvolvemos um EP com alunos de uma escola estadual de Ensino Médio e Técnico na cidade de Bagé/RS, que transformou o andamento da pesquisa dando outra perspectiva ao trabalho. Essa etapa foi composta por quatro encontros realizados às segundas-feiras, em turno inverso ao turno regular de alunos do segundo ano de um Curso Técnico em Mecânica Industrial integrado ao Ensino Médio. A etapa final do trabalho tratou de um curso realizado com um grupo de sete professores e dois profissionais

da educação⁴, compostos por dois módulos: i) módulo I (de formação) desenvolvido no Laboratório de Física da Universidade Federal do Pampa – UNIPAMPA, Campus Bagé, com um total de seis encontros que aconteciam aos sábados nos turnos da manhã e tarde; ii) módulo II (de aplicação nas escolas) em que acompanhamos as atividades desenvolvidas por dois participantes do curso junto aos seus alunos nas escolas.

Nos primeiros capítulos do trabalho faremos uma descrição sobre o referencial teórico utilizado e sobre os estudos relacionados de alguns autores que realizaram uma proposta semelhante, mostrando os principais diferenciais de nossa pesquisa. O capítulo quatro contém os procedimentos metodológicos utilizados, detalhando as fases da intervenção pedagógica realizada (EP e CPPE – Módulos I e II), os métodos de avaliação da intervenção, materiais de ensino utilizados, instrumentos de coleta de dados e como foram realizadas as análises dos resultados obtidos. No capítulo cinco serão descritos detalhadamente os encontros de cada fase da intervenção pedagógica, relatando atividades, observações sobre os participantes e demais informações relevantes à pesquisa. No capítulo seis serão feitas as conclusões finais do trabalho discutindo o que foi observado e analisado ao longo da intervenção, destacando os pontos relevantes dos resultados encontrados e apontadas contribuições para trabalhos futuros.

⁴ Embora os professores também sejam profissionais da educação, fazemos essa diferenciação entre os profissionais que ministram aulas regularmente (professores) e os profissionais que trabalham em escolas e universidades e participam de atividades de ensino, mas não ministram aulas regularmente (profissionais da educação).

2 REFERENCIAL TEÓRICO

Este capítulo se destina a fazer uma apresentação da Teoria da Aprendizagem Significativa de David Ausubel, na qual nossa intervenção pedagógica está embasada.

2.1 TEORIA DA APRENDIZAGEM SIGNIFICATIVA (TAS)

Um das teorias bastante estudadas quando se trata de Ensino de Ciências é a Teoria da Aprendizagem Significativa de David Ausubel. Conforme Moreira (1999) a aprendizagem significativa é aquela em que o significado do novo conhecimento resulta da interação de maneira não-arbitrária e não-literal entre uma nova informação e um aspecto especificamente relevante da estrutura de conhecimento do aprendiz. A aprendizagem significativa ocorre quando um novo conhecimento ancora-se em outros conhecimentos prévios do aprendiz. Esses conhecimentos prévios são definidos como subsunçores. Ausubel apud Moreira (1999, p. 45), argumenta que:

Se tivesse que reduzir toda a psicologia educacional a um só princípio, diria o seguinte: o fator isolado mais importante que influencia a aprendizagem é aquilo que o aprendiz já sabe. Averigüe isso e ensine-o de acordo.

Quando a nova informação interage com um conceito já existente na estrutura cognitiva do aprendiz ocorre a modificação do conceito subsunçor, tornando-o mais elaborado e inclusivo e este servirá de base para novas informações mais abrangentes.

2.1.1 Aprendizagem Mecânica

Em contraposição à aprendizagem significativa há a aprendizagem mecânica definida por Ausubel como as novas informações que são aprendidas sem ter interação com conceitos presentes na estrutura cognitiva do aprendiz, em que a nova informação é armazenada de maneira arbitrária e literal, não interagindo com a já existente. Embora a aprendizagem significativa seja a mais indicada para facilitar a aquisição de significados, em certas situações, a aprendizagem mecânica pode ser necessária no caso de uma fase inicial da aquisição de um novo conhecimento (MOREIRA, op. cit.). Segundo Ausubel (2003), por não estarem bem ancorados, os materiais aprendidos por memorização são mais suscetíveis ao esquecimento e possuem uma capacidade de retenção inferior àqueles aprendidos de maneira não literal e não arbitrária.

2.1.2 Organizadores Prévios

Segundo Moreira (2011) Ausubel recomenda que se usem organizadores prévios para servirem de ancoradouro para um novo conhecimento. Esses organizadores prévios são constituídos por materiais introdutórios que devem ser usados antes do ensino dos conceitos a serem aprendidos, favorecendo a ligação entre o que o aprendiz já sabe com o que se quer ensinar. Segundo Ausubel (1978, p. 171 apud MOREIRA, 1999, p. 51) “a principal função do organizador prévio é servir de ponte entre o que o aprendiz já sabe e o que ele precisa saber para que possa aprender significativamente a tarefa com que se depara”. Os organizadores prévios vêm preencher uma lacuna entre o que o aluno sabe e o que ele precisa saber, a fim de que o novo conhecimento possa ser aprendido de forma significativa.

2.1.3 Condições para Ocorrência da Aprendizagem Significativa

A teoria trata também das condições para a ocorrência da aprendizagem significativa que são: a predisposição do aluno para relacionar o novo material à sua estrutura cognitiva; e que o material seja potencialmente significativo, ou seja, o novo material deve ser relacionável ou incorporável à estrutura cognitiva do aprendiz. Essas duas condições devem ocorrer simultaneamente no processo de aprendizagem, porque mesmo que o material seja potencialmente significativo ao aluno, se sua intenção for de memorizá-lo mecanicamente não haverá a aprendizagem significativa. Da mesma forma, se o aluno estiver predisposto a incorporar o novo conhecimento de forma não arbitrária, mas o material não for potencialmente significativo a aprendizagem se tornará mecânica (MOREIRA, op. cit.).

2.1.4 Diferenciação Progressiva e Reconciliação Integrativa

De acordo com Moreira (2011) Ausubel argumenta como o conteúdo deve ser apresentado aos alunos, definindo os princípios de “Diferenciação Progressiva” e “Reconciliação Integrativa”. Na Diferenciação Progressiva o material deve ser introduzido a partir de conceitos mais gerais e inclusivos para depois serem diferenciados progressivamente em conceitos mais específicos. Por fim, há o princípio da Reconciliação Integrativa chamada assim por Ausubel, em que a instrução deve também explorar relações entre ideias, apontar similaridades e diferenças importantes e reconciliar discrepâncias reais ou aparentes separando ideias.

Com base no que foi exposto a respeito do referencial teórico que embasa a pesquisa, faremos uma reflexão sobre todo o processo de intervenção pedagógica que busca explorar princípios da TAS, como: a importância de conhecimentos prévios para a aquisição de novos conhecimentos; avaliação das condições de ocorrência da aprendizagem significativa por meio do material potencialmente significativo e da predisposição para incorporar os novos conhecimentos; e os fatores que facilitaram e outros que dificultaram a intervenção pedagógica na busca dos indícios de aprendizagem significativa.

3 ESTUDOS RELACIONADOS

A pesquisa dos trabalhos relacionados foi realizada no catálogo de teses e dissertações da CAPES, através da Plataforma Sucupira; no site SciELO⁵; Revista Brasileira de Ensino de Física; e através do Google Acadêmico⁶. Ao todo foram selecionados oito estudos. Uma breve descrição destas publicações é apresentada a seguir.

Desde o início dos anos de 2000, Araujo, Veit e Moreira (2004) já mostrava uma tendência para o uso de computadores no Ensino de Física, principalmente para modelagem e simulação seguido da coleta e análise de dados em tempo real. Os autores apresentam uma revisão de literatura sobre tecnologias computacionais no Ensino de Física do Ensino Médio e Universitário, na qual o conteúdo que está mais presente nas publicações é o de mecânica geral seguido de eletromagnetismo. Quanto aos estudos que tratam do uso da experimentação com a aquisição de dados utilizando o Arduino temos os trabalhos de Silveira e Girardi (2017); Souza et al. (2011); Cavalcante, Tavolaro e Molisani, (2011). No mesmo sentido, de utilizar o Arduino como ferramenta auxiliar para a aprendizagem de conceitos de Física, encontramos o trabalho sobre análise e interpretação de gráficos de cinemática (DWORKOWSKI, 2015); conceitos básicos de eletricidade através de uma sequência didática para o Ensino Médio (FERNANDES, 2015) e aprendizagem de conceitos físicos de eletricidade em regime de corrente alternada (NETO, 2013).

Filho (2015) sugere a utilização da placa Arduino para experimentos de baixo custo para o ensino de Física do nível médio. São feitos vários aparatos experimentais com a aquisição de dados em tempo real integrando o Arduino a outros *softwares* para geração de gráficos. São utilizados guias para orientar os professores na aplicação, além de vídeos com explicações de como montar os equipamentos. No trabalho de Luciano (2014) é feita uma sequência didática com seis alunos do Ensino Médio de uma escola pública. Esses alunos participaram de oficinas de programação e montagem de circuitos para depois construir um robô com matérias de baixo custo. O objetivo do trabalho foi investigar a contribuição da

⁵ SciELO (*Scientific Electronic Library Online*) é uma biblioteca eletrônica que abrange uma coleção selecionada de periódicos científicos brasileiros. Tem por objetivo o desenvolvimento de uma metodologia comum para a preparação, armazenamento, disseminação e avaliação da produção científica em formato eletrônico. Disponível em: <<http://www.scielo.br/>> (Acesso em: 25 nov. 2017).

⁶ Versão do Google para buscas de informações científicas. Disponível em: <<https://scholar.google.com.br/>> (Acesso em: 25 nov. 2017). Fornece uma maneira simples de pesquisar amplamente a literatura acadêmica, permitindo que qualquer documento acadêmico publicado na Web esteja acessível (MUGNAINI e STREHL, 2008).

robótica educacional na aprendizagem de conceitos físico como: de torque e circuitos elétricos.

No estudo de Rodrigues (2014) foram feitas atividades com alunos de um curso técnico de informática, no qual construíram uma estação meteorológica com aquisição de dados pelo Arduino, utilizando-o como uma ferramenta mediadora no Ensino de Física. No contexto onde foi aplicado, vemos que os alunos já possuíam uma boa base de programação, visto que no currículo havia disciplinas afins como a de Técnicas de Programação e Manutenção de *Hardware*.

Cavalcante; Tavolaro e Molisani (2011) apresentam um experimento de carga e descarga de capacitores utilizando aquisição de dados pelo Arduino e gerando gráficos em tempo real. São mostrados os códigos usados e a montagem dos circuitos, mas não há um foco na programação e sim na viabilidade de utilização, embora o título seja “Física com Arduino para Iniciantes” requer alguma experiência em programação.

Souza et al. (2011) demonstram alguns experimentos que podem ser desenvolvidos com a Plataforma Arduino com baixo custo em relação a experimentos utilizando aquisição de dados de forma tradicional utilizando no PC. Destaca os altos custos que precisam ser disponibilizados por *kits* comerciais para a aquisição dados, porque são geralmente produzidos fora do país e importados por empresas nacionais. Segundo os autores, isso inviabiliza o uso para professores e alunos do Ensino Médio. Mostram dois experimentos de Física: o oscilador amortecido e transferência radiativa de calor.

Nos trabalhos citados, normalmente, supõe-se que o leitor possua um conhecimento mínimo em programação e eletrônica, o que se torna, muitas vezes, um empecilho para o professor ou aluno começar a utilizar o Arduino. Então, surgiu a preocupação de fazer um material instrucional com foco nas questões que achamos essenciais para os que desejam iniciar um trabalho sobre o uso de plataformas como o Arduino.

Com base nos estudos aqui apresentados podemos inferir que atualmente há inúmeras publicações sobre o tema, mas são escassas as que envolvem o ensino de como se trabalhar com a programação e a montagem de circuitos, de modo que se procure dar condições aos alunos para programarem e construírem seus próprios projetos e entenderem o funcionamento dos circuitos relacionados. Cabe destacar que não foram encontrados trabalhos que explorem cursos de formação de professores ou similares. Assim, observamos que os trabalhos a

respeito do uso da Plataforma Arduino não têm, em sua maioria, a preocupação com o entendimento da lógica de programação. Nota-se que os trabalhos pesquisados, embora utilizem a mesma Plataforma, possuem outros objetivos em relação à interação dessas tecnologias com o Ensino de Ciências. Esses autores estão preocupados com as formas de utilização da Plataforma Arduino como aporte à aprendizagem de conceitos estudados em sala de aula e não têm, como objetivo principal, ensinar programação e/ou eletrônica. Nosso trabalho consiste em explorar a utilização dessa tecnologia aliada a conceitos de Física, de programação, de eletrônica, mas com a intenção de dar embasamento teórico e prático aos que pretendem levar a utilização do Arduino à sala de aula e, ao mesmo tempo, desenvolver uma cultura nas escolas, para que os projetos executados por professores e alunos sejam, de fato, construídos e compreendidos por eles, não apenas reproduzidos.

4 PROCEDIMENTOS METODOLÓGICOS

Neste capítulo serão apresentadas todas as etapas referentes à metodologia da pesquisa realizada. A pesquisa é do tipo intervenção pedagógica, na acepção de Damiani et al. (2013), com abordagem predominantemente qualitativa e está dividida em: método de intervenção pedagógica, composta por uma descrição detalhada sobre a intervenção e o método de avaliação da intervenção, que detalha como será avaliado o processo de intervenção através da análise dos dados e dos instrumentos utilizados para obtenção destes.

4.1 MÉTODO DE INTERVENÇÃO PEDAGÓGICA

O processo de intervenção pedagógica foi dividido em duas etapas: Estudo Piloto (EP) e Curso para Professores e Profissionais da Educação (CPPE). Cada uma das etapas será descritas a seguir.

4.1.1. Estudo Piloto (EP)

Na primeira etapa de nosso trabalho, realizamos um Estudo Piloto com alunos do 1º e 2º ano do Ensino Médio, em uma escola estadual de Ensino Médio, localizada no município de Bagé - RS, que oferece a modalidade de Educação Tecnológica, articulando o Curso Técnico em Mecânica Industrial com o Ensino Médio regular. A realização deste estudo preliminar se deu a partir da ideia de levar o ensino de programação e conhecimentos de automação ao alcance de alunos do Ensino Básico, articulado com os conteúdos trabalhados em sala de aula. Uma forma de nos colocarmos frente a uma diversidade de situações, nos levou a propor um Estudo Piloto.

Os objetivos específicos do EP foram:

- i. Avaliar o material didático elaborado;
- ii. Verificar as possibilidades e dificuldades encontradas durante a inserção deste tema na Educação Básica; e
- iii. Levantar questões norteadoras para a etapa do CPPE.

4.1.2 Curso para Professores e Profissionais da Educação (CPPE)

Esta etapa se dividiu em módulos I e II. O módulo I tratou das atividades de formação com um grupo de professores e profissionais da educação, com o objetivo de introduzi-los nos principais conceitos necessários à iniciação para o uso da Plataforma Arduino,

especificamente, no ambiente escolar. O módulo II configurou-se enquanto aplicação do que foi aprendido no curso, através de atividades propostas pelos próprios participantes e colocados em prática juntamente com seus alunos, nas escolas onde trabalham.

4.1.2.1 Módulo I: Formação

No Módulo I, foi elaborado e executado um curso voltado a professores e profissionais da educação que atuam nos Ensinos Fundamental, Médio e Superior. A aplicação do curso proposto, especificamente para este público, foi consequência dos resultados obtidos no Estudo Piloto (EP) realizado com os alunos do Ensino Médio. A divulgação do curso foi realizada através de cartazes (apêndice A) em escolas e *e-mails* enviados para a Secretaria Municipal de Educação de Bagé (SMED) e para a 13ª Coordenadoria Regional de Educação do Estado do RS (CRE), além de ampla divulgação através de redes sociais. As inscrições e todo o processo de identificação dos interessados foram feitas por meio de formulário eletrônico⁷. Não houve seleção porque tivemos 12 inscritos e, mesmo tendo disponibilizado dez vagas inicialmente, aceitamos a inscrição de todos. O curso foi estruturado em aulas presenciais e atividades a distância. As presenciais foram realizadas no Laboratório de Física da Universidade Federal do Pampa (UNIPAMPA - Campus Bagé, fig. 1), onde possuíamos um espaço amplo e ideal para as atividades práticas. A carga horária deste módulo de formação foi de 30 horas na modalidade semipresencial (ou seja, presencial e a distância). O conteúdo da formação era composto por: conceitos iniciais sobre a Plataforma Arduino, noções de lógica de programação, noções de eletrônica e automação utilizando sensores e atuadores. O curso foi dividido em quatro encontros presenciais, realizados aos sábados, com duração de seis horas cada, conforme quadro 1.

⁷ Cf. apêndice E. Disponível em: <<https://goo.gl/forms/ez7OfHD7oJ4k3kzH2>> (Acesso em: 21 nov. 2017).

Quadro 1 - Distribuição dos dias e horas presenciais dos encontros do CPPE realizado.

Encontros	Dia (sábados)	Manhã	Tarde
1º	17/09	9h às 12h	13h30min às 16h30min
2º	24/09		
3º	08/10		
4º	22/10		

Fonte: autoria própria

Figura 1 – Laboratório de Física da UNIPAMPA onde ocorreram os encontros presenciais do CPPE.



Fonte: acervo do autor.

Quanto aos objetivos do módulo I do CPPE, estão divididos em objetivos específicos e de aprendizagem. Os objetivos específicos se relacionam com a finalidade da pesquisa de intervenção pedagógica, são eles:

- i. Fornecer conhecimentos introdutórios sobre a Plataforma Arduino e automação para professores e profissionais ligados à Educação Básica;
- ii. Desenvolver discussões sobre as forma de integração dessas tecnologias com os conteúdos trabalhados nas escolas; e
- iii. Motivar e incentivar professores e profissionais da educação a levarem os conhecimentos adquiridos no CPPE aos alunos do Ensino Básico.

Os objetivos de aprendizagem se relacionam aos “achados relativos aos efeitos da intervenção sobre seus participantes” (DAMIANI et al., 2013, p. 62) pois, a partir da

investigação sobre os conhecimentos adquiridos pelos participantes, procuramos verificar evidências de aprendizagem, baseadas nos objetivos de aprendizagem traçados, apresentados a seguir:

- i. Compreender os conceitos básicos da programação (funções, operadores lógicos e entradas e saídas) e funcionamento da plataforma Arduino;
- ii. Compreender conceitos de automação utilizando sensores e atuadores;
- iii. Identificar erros em códigos de programação;
- iv. Identificar e relacionar as portas digitais com os circuitos elétricos correspondentes;
- v. Montar circuitos elétricos utilizando a *protoboard*;
- vi. Construir códigos e montar circuitos para obtenção de medidas elétricas pelas portas analógicas; e
- vii. Compreender os conceitos de diferença de potencial, resistência e corrente elétrica através da análise de circuitos divisores de tensão utilizados nas montagens com sensores.

Durante a realização do curso, utilizamos como materiais⁸, *kits* Arduino, *notebooks*, *datashow* e quadro branco. Todos os participantes receberam uma apostila⁹ (apêndice B) elaborada por nós, com a sequência discutida no curso, contendo toda a informação necessária para a realização das atividades propostas; com explicações teóricas, práticas de montagem dos circuitos e exercícios. Os *kits* Arduino (fig. 2) foram entregues por duplas e devolvidos no final de cada encontro ou levados para casa por um dos integrantes das duplas. O *kit* era composto por:

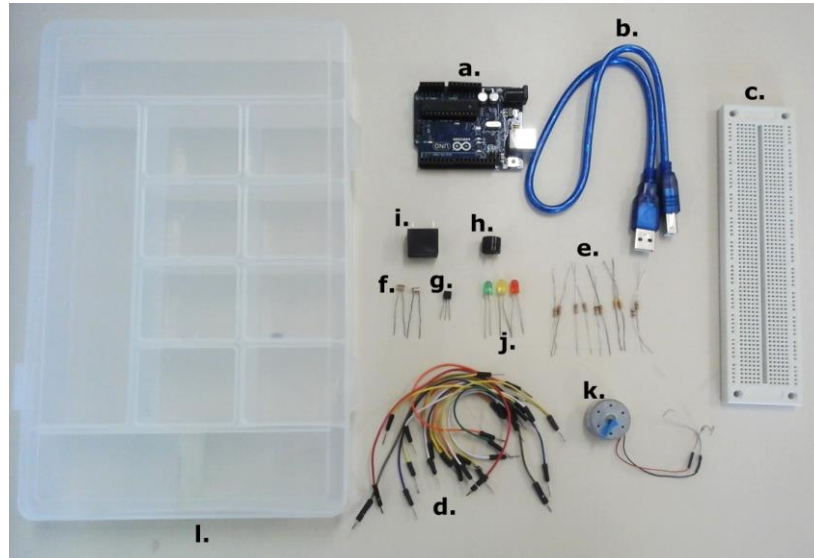
- a. 1 placa microcontrolada Arduino;
- b. 1 cabo USB (conecta a placa ao computador);
- c. 1 *protoboard* (matriz de contato);
- d. 20 *jumpers* para conexão;
- e. 10 resistores (330 Ω ; 1k Ω ; 1,5k Ω ; 5,6k Ω e 10k Ω - 2 de cada valor);
- f. 2 LDRs (sensor de luminosidade);
- g. 1 sensor de temperatura LM35;
- h. 1 *buzzer* (pequeno alarme ou campainha);
- i. 1 relé;
- j. 3 LEDs;

⁸ Todos os materiais utilizados no curso, com exceção da apostila, foram disponibilizados pela UNIPAMPA.

⁹ A apostila foi elaborada, impressa e encadernada com recursos próprios do autor.

- k. 1 pequeno motor 5V DC (corrente contínua de 5V); e
- l. 1 caixa organizadora.

Figura 2 - *Kit Arduino*



Fonte: acervo do autor.

A apostila (apêndice B) é um guia para os participantes que contém os conteúdos como: a introdução à lógica de programação, os aspectos principais para a utilização do Arduino e como utilizar os sensores e atuadores. Esses guias são focados nas atividades práticas com explicações de montagem do Arduino, através de sequências que facilitam a compreensão, aumentando pouco a pouco o nível de complexidade dos conteúdos. De acordo com Moreira (2011), os materiais disponibilizados devem ser potencialmente significativos, pois precisam ser organizados, estruturados, conter exemplos, linguagem adequadas e, principalmente, levar em conta o conhecimento prévio dos participantes do curso. Nesta compreensão, a apostila foi elaborada conforme observação dos encontros do EP, especificamente nos dias anteriores a cada encontro, para que fosse a mais estruturada possível, de acordo com o andamento das práticas em de cada encontro. Assim, procuramos desenvolver um material com atividades que realmente fossem aplicáveis em sala de aula e que possuíssem exemplos práticos e detalhados.

As atividades a distância envolveram todas as tarefas realizadas fora dos horários dos encontros. Incluem-se as horas utilizadas para responder questionários *online*, fóruns de discussão sobre os conteúdos dos encontros e as dúvidas dos participantes enviadas por *e-*

mail. A carga horária das atividades a distância foi de 6 horas e as atividades presenciais de 24 horas, perfazendo um total de 30 horas.

4.1.2.2 Módulo II: Aplicação nas Escolas

Neste módulo, houve o acompanhamento de dois professores participantes do curso, os quais aceitaram o desafio de aplicarem em suas turmas algumas atividades que envolvessem práticas desenvolvidas no curso as quais foram aplicadas logo após o término do módulo de formação. Dos participantes deste módulo de aplicação temos: um professor de Física do Ensino Médio, da cidade de Candiota/RS e um professor de Tecnologias Educacionais do Ensino Fundamental, da cidade de Bagé/RS.

4.2 MÉTODO DE AVALIAÇÃO DA INTERVENÇÃO

O método da avaliação da intervenção envolve os instrumentos de pesquisa utilizados para a coleta e análise de dados da intervenção pedagógica. A avaliação do processo propicia o caráter investigativo da intervenção pedagógica. Segundo Damiani et al. (2013, p. 62) a avaliação possui dois elementos principais: “os achados relativos aos efeitos da intervenção sobre seus participantes e os achados relativos à intervenção propriamente dita”, considerando que:

O primeiro grupo de achados volta-se à análise das mudanças observadas nos sujeitos participantes. Os dados referentes a essas mudanças, coletadas com o auxílio de diferentes instrumentos, são examinados à luz do referencial teórico que embasou a intervenção [...]. Os achados relativos aos efeitos da intervenção devem ser expostos por meio de descrições densas e interpretações detalhadas, incluindo exemplos retirados do *corpus* de dados empíricos (DAMIANI et al., 2013, p. 63).

Os achados desse primeiro grupo citado relacionam-se diretamente com as evidências observadas, baseada nos objetivos de aprendizagem¹⁰ traçados para o CPPE. Sobre os dados referentes às mudanças verificadas nos participantes, estes foram coletados por meio de questionários, interpretação de grupos focais e observações feitas ao longo das atividades realizadas, caracterizando individualmente cada participante. A análise do conjunto de dados obtidos foi feita pela perspectiva da Teoria da Aprendizagem Significativa (AUSUBEL, 2003). A partir dessas evidências, de caráter qualitativo, os achados sobre os participantes serão descritos nas considerações do CPPE de forma detalhada e organizada por meio de um

¹⁰ Cf. objetivos de aprendizagem p. 20.

quadro avaliativo em que cada objetivo de aprendizagem é classificado conforme o grau de entendimento que o participante atingiu, são eles:

- **Atingiu Completamente (AC)** – quando o objetivo é atingido de forma ampla no qual todos os conceitos envolvidos são compreendidos completamente. Neste caso podemos dizer que não ficaram dúvidas sobre o que foi trabalhado. O avaliado consegue realizar a atividade com êxito sem auxílio.
- **Atingiu Largamente (AL)** – quando o objetivo é atingido de forma quase completa, mas alguns conceitos ainda precisam ser mais trabalhados para que tenha uma completa compreensão. O avaliado consegue realizar a atividade, mas eventualmente poderá precisar de auxílio para ter êxito.
- **Atingiu Parcialmente (AP)** – quando o objetivo foi atingido em parte, mas há muitos conceitos que ainda não foram bem compreendidos. O avaliado possui um entendimento suficiente para desempenhar a atividade, mas ainda precisa de um acompanhamento razoável para ter êxito.
- **Atingiu Minimamente (AM)** – quando o objetivo foi atingido de maneira muito superficial, embora alguns conceitos tenham sido assimilados. O avaliado não tem um entendimento suficiente para realizar a atividade com êxito sem o devido acompanhamento.

No segundo grupo temos:

Os achados relativos à avaliação da intervenção propriamente dita enfocam a análise da(s) característica(s) da intervenção responsável(eis) pelos efeitos percebidos em seus participantes. Tal análise discute os pontos fracos e fortes da intervenção, com relação aos objetivos para ela traçados e, caso se aplique, julga as modificações que foram introduzidas durante seu curso, frutos das constantes reflexões realizadas durante o processo interventivo (DAMIANI et al., 2013, p. 63).

Os pontos fracos e fortes citados estão expostos através de um quadro comparativo de “facilidades e dificuldades” encontradas durante a intervenção, constituindo os nossos “achados relativos à intervenção propriamente dita”. Esses achados fazem parte do conjunto de análises e reflexões de todo nosso processo interventivo.

4.2.1 Instrumentos de Coleta de Dados

Os instrumentos utilizados para coletar dados da intervenção foram:

i. Gravações de áudio - utilizadas em todas as etapas da intervenção (EP e CPPE). Os encontros foram gravados para que nenhuma informação, por menor que seja, fosse perdida.

ii. Observação não estruturada e participante - utilizadas em todas as etapas da intervenção (EP e CPPE). Na observação não estruturada, segundo Vianna (2007), o observador procura se integrar pela perspectiva do observado, apenas observando e registrando o que acontece. Sendo, também, o observador parte do evento pesquisado tem-se como participante. No caso de nossas observações, feitas no decorrer da intervenção, podemos caracterizar, então, como não estruturada e participante.

iii. Grupo focal - utilizada apenas no CPPE - Módulo I da nossa intervenção, pois no EP, consideramos que os alunos de Ensino Médio não teriam a mesma desenvoltura em participar de um grupo focal, obtendo assim menos dados. Já no CPPE, entendemos que os professores e profissionais da educação seriam mais participantes se nos utilizássemos deste instrumento, gerando mais dados. Notadamente esses profissionais têm maior facilidade em expressar suas opiniões, experiências profissionais e, até mesmo, discorrer sobre aquilo que não concordam ou que gostariam que fosse diferente. O uso do grupo focal foi utilizado em vez de entrevistas individuais, também, porque se torna menos formal, no qual são colocadas, durante as falas, questões ao grupo em geral para que os participantes falassem livremente sobre o assunto instigado. Segundo Morgan (1997, apud GONDIM, 2003, p.151) grupos focais são definidos como “uma técnica de pesquisa que coleta dados por meio das interações grupais ao se discutir um tópico especial sugerido pelo pesquisador. Como técnica, ocupa uma posição intermediária entre a observação participante e as entrevistas em profundidade”.

iv. Questionários - utilizados no EP e no CPPE - Módulo I foram utilizados para avaliar a intervenção por parte dos participantes, a fim de verificar seus conhecimentos prévios sobre algumas partes dos conteúdos que tratados e de verificar o andamento da sua aprendizagem no decorrer das atividades. Foram constituídos com perguntas objetivas e subjetivas.

No capítulo de resultados e discussões faremos uma descrição detalhada de como foi realizada a intervenção pedagógica (descrevendo cada aula separadamente), discutindo-se os objetivos dos EP e CPPE e apontando quando foram atingidos, através de evidências observadas nas atividades. Os possíveis motivos das facilidades e dificuldades de inserção de nossa proposta na Educação Básica (que é o foco principal da intervenção pedagógica realizada) serão discutidos na seção 5.2.4 denominada de “considerações sobre o CPPE: achados relativos à intervenção propriamente dita”.

5 RESULTADOS E DISCUSSÕES

No presente capítulo, faremos uma descrição detalhada de como aconteceram os encontros do Estudo Piloto (EP) e do Curso para Professores e profissionais da Educação (CPPE), módulos I e II. Além da apresentação dos resultados obtidos, faremos uma discussão sobre os elementos percebidos ao longo das atividades realizadas que serão analisados de forma predominantemente qualitativa. Serão observados, principalmente, os efeitos da intervenção sobre os participantes e as facilidades e dificuldades encontradas na intervenção pedagógica realizada.

5.1 ESTUDO PILOTO (EP)

O primeiro contato com os participantes do estudo foi por meio de uma palestra aberta a todos os alunos do Ensino Médio, realizada no auditório da escola. A palestra aconteceu dia 6 de maio de 2016 e foi realizada por apresentação de *slides*. No total, participaram desta palestra 37 alunos de todos os anos, pois foi um meio de reforçar a divulgação do nosso estudo piloto que se daria na forma de um curso sobre programação e Plataforma Arduino, em que os alunos puderam ter uma primeira noção sobre questões básicas a respeito do tema, tais como: o que é e o que podemos fazer com o Arduino; principais formas de utilização; modelos de placa existentes; funções das portas de entradas e saídas; comunicação da plataforma com o computador e exemplos com códigos e montagens que podem ser feitas. Após uma breve apresentação sobre nossa trajetória e os objetivos que nos levaram a estar ali, começamos a interagir com os presentes perguntando: quem já conhece ou já viu falar sobre o Arduino?

Dos 37 presentes na palestra, apenas três alunos relataram que já tinham assistido alguns vídeos, pela *internet*, de projetos utilizando a placa Arduino, mas que não nunca tinham usado. Em seguida passamos a mostrar as principais características da placa e o que poderíamos fazer com essa (em um nível bem básico de funcionalidades). Demonstrados os conceitos mais teóricos passamos a fazer algumas demonstrações práticas como acender um LED¹¹ controlado por um sensor LDR¹². Também realizamos uma demonstração utilizando um sensor de temperatura, com o qual era possível visualizar as variações em graus Célsius na tela do computador. Em seguida, abrimos espaços para perguntas e os alunos, demonstrando

¹¹ *Light Emitting Diode* – diodo emissor de luz (em português).

¹² *Light Dependent Resistor* – resistor dependente de luz (em português).

interesse, realizaram vários questionamentos sobre alguns pontos abordados. Ao final, convidamos os alunos para participar do curso que denominamos “Curso de Arduino e Programação”.

A maioria dos presentes na palestra demonstrou interesse em participar do curso, mas alguns não puderam participar por motivos de conflito com seus horários de aula. A direção da escola nos disponibilizou uma sala com computadores, quadro e *datashow*, que poderia ser utilizada somente na parte da tarde, quando muitos dos interessados estariam em aula. Os encontros foram realizados entre os dias 09 de maio e 20 de junho de 2016, às segundas-feiras nos horários de 13h30min às 17h. Durante todos os encontros, estava presente, além do autor deste trabalho, um aluno do Programa Institucional de Bolsas de Iniciação à Docência (PIBID) do Curso de Licenciatura em Física da UNIPAMPA – Campus Bagé, apoiando os alunos na distribuição e organização do material, favorecendo um melhor aproveitamento do tempo que tínhamos para realizar todas as atividades pretendidas. O detalhamento do que foi realizado nos encontros será o que apresentaremos a seguir.

5.1.1 Primeiro Encontro (09 mai 2016)

No primeiro encontro, ocorrido três dias depois da palestra, estavam presentes dez alunos e começamos discutindo como seriam estruturadas as aulas e quais atividades iríamos desenvolver. Em seguida, tratamos de conceitos mais gerais e básicos da programação. Nesta parte, mais teórica, mostramos alguns conceitos importantes como: o que é um algoritmo; o que é um programa de computador; as diferenças entre os dois termos; o que é o sistema binário e como funciona a conversão de um número binário para decimal e vice-versa; e princípios básicos da lógica de programação. Nesta aula não desenvolvemos atividades práticas com o material dos *kits* Arduino. As atividades realizadas neste primeiro momento tiveram o objetivo de proporcionar o entendimento dos principais operadores lógicos. Para esse propósito, mostramos vários exemplos fazendo uma analogia entre portas digitais e circuitos elétricos contendo chaves, fontes de tensão e associações de lâmpadas. Também realizamos atividades utilizando planilhas eletrônicas¹³ em que os alunos, em duplas, foram desafiados a construir pequenos comandos para simular as entradas e saídas de portas digitais utilizando os comandos lógicos OU, E e NÃO (normalmente a identificação mais comum é na língua inglesa como OR, AND e NOT).

¹³ O programa de planilhas eletrônicas utilizado foi o Microsoft Excel.

Nesta aula, após explicações teóricas sobre noções conceituais da programação, os participantes desenvolveram atividades de conversão entre números binários e decimais. Os alunos não tiveram muitas dificuldades na conversão de valores decimais para binários, em que usamos apenas a divisão por dois, mas encontraram algumas nas conversões de binário para decimal, usando números com potências de base 10. Na atividade com as planilhas eletrônicas, atribuímos que a dificuldade encontrada pela maioria ao pouco conhecimento sobre o programa. Neste encontro, a orientação foi para que os alunos utilizassem as células das planilhas eletrônicas como entradas e saídas reproduzindo as portas digitais E, OU e NÃO, montando, assim, circuitos digitais simulados. A atividade das planilhas foi apresentada aos alunos através do exemplo da fig. 3.

Figura 3 – Planilha eletrônica do Microsoft Excel com as fórmulas utilizadas na simulação das portas digitais.

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5	Entradas			Porta Digital		Saídas		Fórmulas correspondentes
6								
7	0							
8	1			OU (OR)		=OU(A7;A8)		"=OU(A7;A8)"
9								
10								
11				E (AND)		FALSO		"=E(A7;A8)"
12								
13								
14				NÃO (NOT)		VERDADEIRO		"=NÃO(A7)"
15								
16								

Fonte: autoria própria.





Existem programas específicos que realizam simulações com portas digitais como o *Logisim*¹⁴, mas utilizamos as planilhas para que os participantes escrevessem os comandos necessários e começassem a entender os operadores lógicos. Esta atividade vem ao encontro da ideia de, no decorrer do curso, fazer com que o aluno entenda a principal funcionalidade da plataforma Arduino: que é a de atuar no mundo físico (saídas) a partir de uma verificação e análise de sensores conectados a suas entradas. No início não estávamos muito preocupados

¹⁴ O *Logisim* é uma ferramenta educacional para a concepção e a simulação digital de circuitos lógicos. Com uma interface simples e com ferramentas para simular circuitos à medida em que são construídos. É gratuito e o link para download bem como outras informações podem ser encontrados em: <<http://www.cburch.com/logisim/pt/>> (Acesso em: 20 mai. 2017).

em explicar a relação dessas planilhas com a proposta do curso, mas sim fazê-los começar a utilizar uma linguagem diferente e próxima do que utilizaram nas próximas atividades do curso. Organizados em duplas, todos conseguiram executar a atividade. Alguns tiveram mais dificuldade por nunca terem utilizado o recurso de planilhas. Já outros, ficaram curiosos quando souberam que essas poderiam ser usadas para programar alguns comandos. A fig. 4 mostra a relação entre as portas, fórmulas e seu símbolo correspondente.

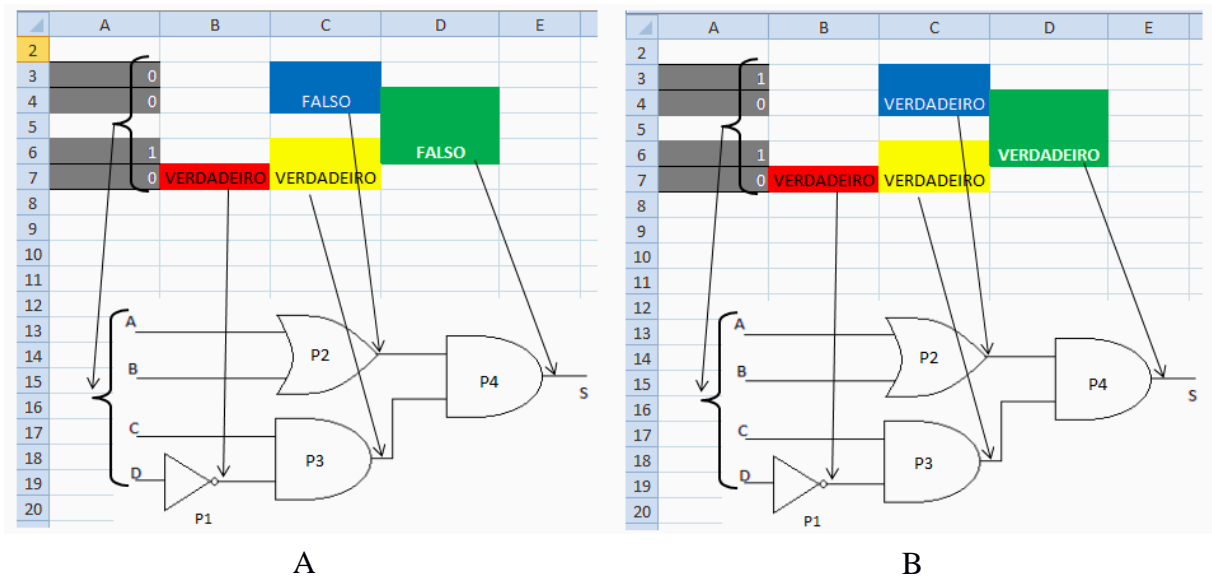
As fig. 5a e 5b mostram de que forma foi realizada a proposta com os alunos. As células de cor cinza (A3, A4, A6 e A7) são as entradas do circuito digital que poderiam variar conforme a sequências de 0 e 1 escolhida. Estas células não possuem fórmulas, pois são apenas entradas que devem ser preenchidas com 0 e 1. As células coloridas: vermelha (B7), azul (C3), amarela (C6) e verde (D4) são as “portas digitais” representadas por cada fórmula fixa contendo um tipo de operador lógico. A saída de cada célula que contém a fórmula é a expressão “VERDADEIRO” e “FALSO”, que representa o valor 1 e 0, respectivamente. A primeira fórmula está na célula vermelha (B7) e como é uma função NÃO (NOT) inverte o valor de sua entrada (célula A7). A fórmula da célula azul (C3) é uma função OU (OR) que compara os valores de sua entrada que são as células A3 e A4 dando uma saída “VERDADEIRO”, se pelo menos uma das entradas for verdadeira (1) e “FALSO”, caso as duas entradas forem falsas (0). A fórmula da célula amarela (C6) é uma função E (AND) que compara os valores de sua entrada, que são as células A6 e B7 (saída da célula vermelha) dando uma saída “VERDADEIRO” se todas as entradas forem verdadeiras (1) e “FALSO” caso uma das entradas for falsa (0). A fórmula da célula verde (D4) é outra função E, que verifica suas entradas (que são as saídas das células azul e amarela). Note que para uma resposta final de um circuito as saídas de umas se tornam entradas das seguintes.

Figura 4 – Fórmulas utilizadas para cada tipo de porta digital.

Porta digital	Fórmula Excel	Tipo de Porta	Símbolo
P1	=NÃO(A7)	NOT	
P2	=OU(A3;A4)	OR	
P3	=E(A6;B7)	AND	
P4	=E(C3;C6)	AND	

Fonte: autoria própria.

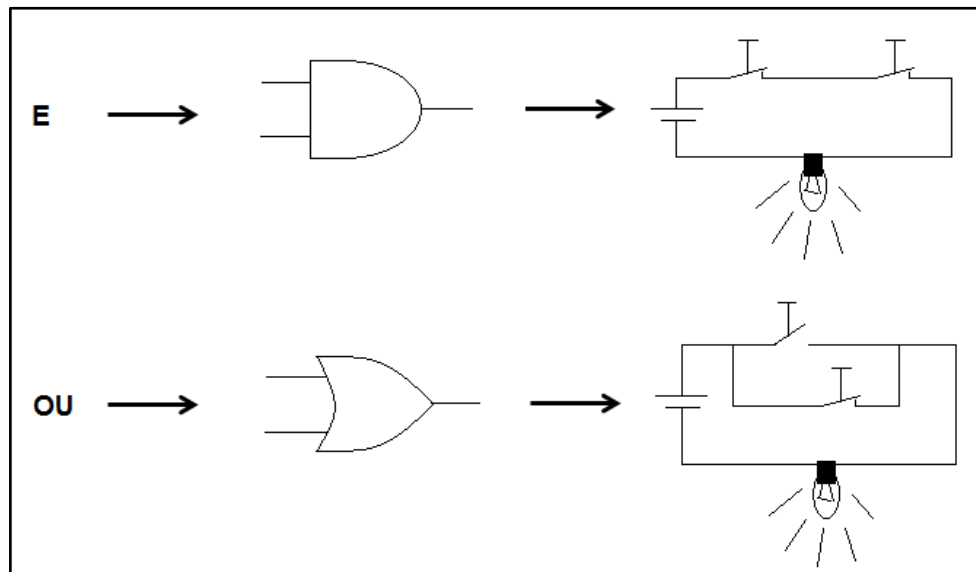
Figuras 5a e 5b – Circuito digital representado por células com fórmulas. Saídas: falsa (A) e verdadeira (B).



Fonte: autoria própria.

A analogia realizada entre os circuitos elétricos e portas digitais também têm a função de mostrar esses operadores lógicos de uma forma mais familiar. Mostramos aos alunos imagens de circuitos elétricos com lâmpadas e chaves em série e em paralelo que podem ser tratados como portas digitais. Na fig. 6, podemos ver esses circuitos. As chaves representam as entradas e quando abertas correspondem ao valor “0” ou “falso”, já as chaves fechadas correspondem ao valor “1” ou “verdadeiro”. As lâmpadas representam as saídas e quando acesas correspondem ao valor “1” ou “verdadeiro”. Quando apagadas correspondem ao valor “0” ou “falso”.

Figura 6 - Relação entre o operador lógico, porta digital e circuito elétrico correspondente.



Fonte: autoria própria.

Ao final da aula, perguntamos aos alunos se tinham entendido essas comparações e alguns relataram que com o exemplo dos circuitos com lâmpadas ficou mais fácil de compreender o funcionamento das portas e dos operadores. Naquele momento, começamos a explicar por que tratamos dessas portas digitais, que futuramente foram importantes para compreender as entradas e saídas da placa Arduino. Os alunos demonstraram interesse tanto nas atividades com as planilhas quanto da análise dos circuitos. Nos relatos dos próximos encontros serão descritas as atividades já com a manipulação dos *kits* Arduino.

5.1.2 Segundo Encontro (16 mai 2016)

Estavam presentes neste dia seis alunos. Começamos lembrando o que foi discutido no encontro anterior. Estava programado para este encontro mostrar, na prática, a programação do Arduino. Tivemos um imprevisto, pois os computadores que a escola disponibilizou não eram compatíveis e não conseguimos que rodassem o programa perfeitamente. Depois de várias tentativas sem êxito, resolvemos apresentar os primeiros passos da programação utilizando os computadores dos próprios alunos – já que alguns levavam seus *notebooks* para os encontros – formando grupos maiores. Como foram menos presentes do que no primeiro encontro, fizemos grupos de três utilizando dois computadores e mais a apresentação com *datashow*. Passamos então a apresentar a janela de programação do

IDE¹⁵ do Arduino, onde digitamos as linhas de comando para depois enviar para a placa. Com um código exemplo (fig. 7) projetado igual ao que eles tinham na apostila¹⁶ identificamos cada item e linha detalhadamente, especificando quais suas funções na programação. Essa primeira visão geral apresentada foi para que eles pudessem se familiarizar com a forma de escrita da programação e com as funcionalidades principais do programa. Foi uma parte importante da aula, quando os alunos tiveram o primeiro contato com um código contendo declarações de variáveis, entradas e saídas e funções que, embora simples, possuíam um objetivo definido de acender um LED enquanto um botão estivesse pressionado. Também realizamos o carregamento desse código para a placa e a montagem do botão e LED, para demonstrar o funcionamento do programa.

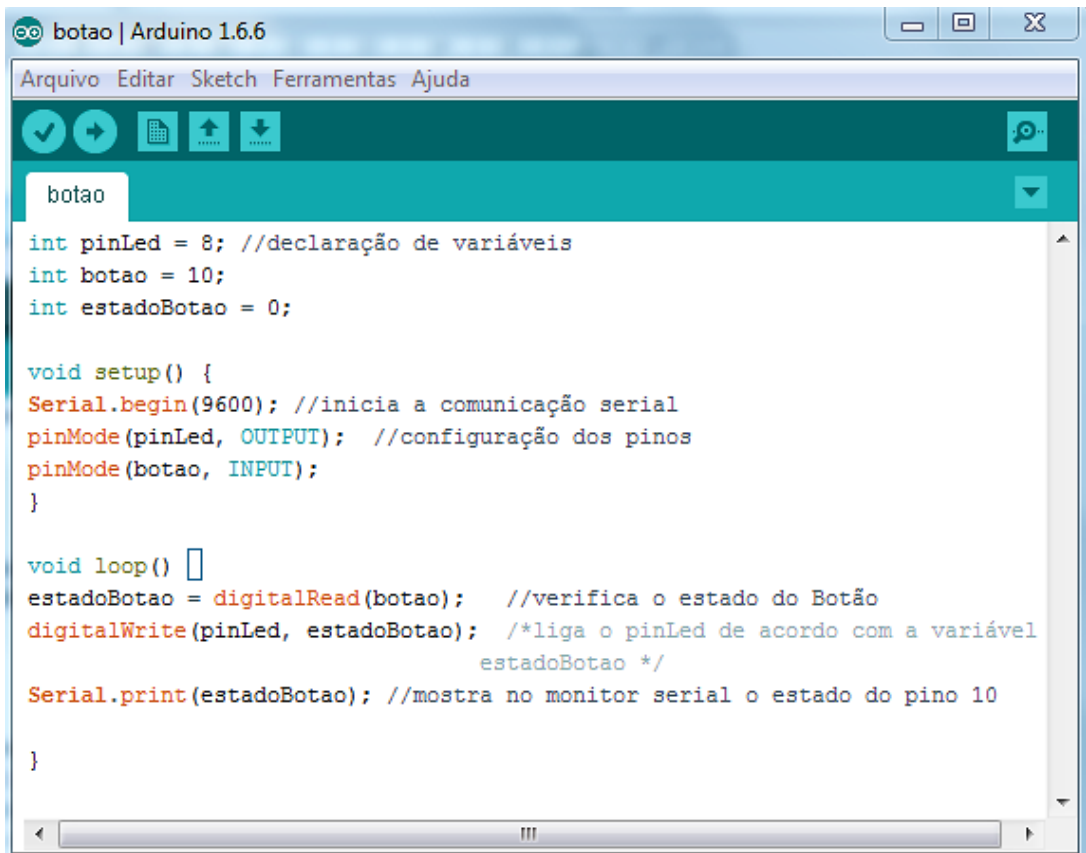
A primeira impressão dos alunos a respeito do código exemplo foi verificada pela expressão de dúvida e dificuldade de compreensão. Num primeiro momento, alguns comentaram que talvez não conseguissem trabalhar com aquela linguagem de programação, evidenciando, claramente sua ansiedade ao ver um código de programação pela primeira vez. O que parecia assustar, aos poucos foi se tornando mais compreensível para alguns à medida que os códigos iam sendo analisados linha por linha. Uma aluna perguntou se era possível utilizar a placa Arduino sem precisar entender esses códigos. Ao responder que era essencial a compreensão, ela ficou preocupada.

Na ocasião relatada, notamos que, embora os alunos estivessem motivados com o curso e com as possibilidades de aprendizado, ainda estavam presos a um imediatismo de aprender algo e tinham grande dificuldade com outras formas de ensino, que não a de simplesmente decorar um conteúdo para a prova. Ao que nos pareceu, a montagem do exemplo que deveria acender o LED com o botão, analisando passo a passo todo o processo e cada linha do programa motivou um pouco mais os alunos. Talvez porque analisando todo processo, do código até o resultado final do funcionamento, foi mostrada a finalidade de cada item estudado até então. Neste exemplo, alguns alunos já começavam a compreender as entradas e saídas da placa Arduino como algo importante, pois fizeram referências ao que foi tratado no encontro anterior, ou seja, o botão (sensor) ligado a uma entrada e o LED (atuador) ligado a uma saída.

¹⁵ IDE – *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado) é o programa onde podemos editar os códigos e enviar para a placa. Possui várias ferramentas para desenvolvimento de projetos. Disponível em: <<https://www.arduino.cc/en/Main/Software>> (Acesso em: 21 mai. 2017).

¹⁶ A apostila entregue aos alunos encontra-se no apêndice B.

Figura 7 – Janela do IDE com o código exemplo.



```

botao | Arduino 1.6.6
Arquivo Editar Sketch Ferramentas Ajuda
botao
int pinLed = 8; //declaração de variáveis
int botao = 10;
int estadoBotao = 0;

void setup() {
  Serial.begin(9600); //inicia a comunicação serial
  pinMode(pinLed, OUTPUT); //configuração dos pinos
  pinMode(botao, INPUT);
}

void loop() {
  estadoBotao = digitalRead(botao); //verifica o estado do Botão
  digitalWrite(pinLed, estadoBotao); /*liga o pinLed de acordo com a variável
                                     estadoBotao */
  Serial.print(estadoBotao); //mostra no monitor serial o estado do pino 10
}

```

Fonte: autoria própria.

5.1.3 Terceiro Encontro (23 mai 2016)

No terceiro encontro estavam presentes oito alunos. Nesse dia, pedimos para que respondessem um questionário (apêndice C) sobre circuitos elétricos que ainda não havia sido discutido até então, para servir como uma avaliação prévia do conhecimento da turma, além de outras questões relacionadas ao que tinha sido abordado nas primeiras aulas. Esses circuitos eram compostos principalmente por chaves e lâmpadas em associações mistas. Em seguida, fizemos uma longa discussão sobre essas questões, principalmente as relacionadas aos efeitos da corrente elétrica, tensão e resistência elétrica. Os alunos comentavam suas respostas mostrando por que as escolheram e nós corrigimos explicando cada detalhe da questão e dando exemplos utilizando um *software*¹⁷ de simulação de circuitos. Depois de corrigir e discutir as questões, falamos dos sensores e atuadores e do divisor de tensão e então, realizamos uma atividade prática, na qual os alunos montaram um divisor de tensão com

¹⁷ Programa de simulação de circuitos utilizado foi o *Multisim* da *National Instruments*. Possui um conjunto de ferramentas para a construção e simulação de circuitos para estudo de eletrônica. Existe uma versão para estudantes. Disponível em: <http://www.ni.com/multisim/pt/> (Acesso em: 21 mai. 2017).

resistores e mediram seu valor de saída na porta analógica do Arduino e depois compararam com o valor medido com o voltímetro. Nessa prática, foram acrescentadas algumas informações sobre cálculo do divisor de tensão, leitura das cores dos resistores e medição de tensão utilizando um multímetro digital.

Com a aplicação do questionário identificamos que os alunos possuíam muito pouco conhecimento sobre os conceitos de tensão, resistência e corrente elétrica. Não sabiam a verdadeira causa de um curto circuito e possuíam muitos equívocos sobre as questões apresentadas. Essa informação, a respeito dos conhecimentos prévios, foi retirada não apenas das respostas marcadas no questionário, mas também da sua correção, quando pedimos que cada um dos alunos comentassem suas respostas e motivos pelos quais as escolheram. Essa dinâmica de correção ajudou-nos a verificar quais as questões eram respondidas com mais convicção e quais foram respondidas apenas com um “chute”, por realmente não saberem a resposta ou por não conseguirem eliminar alternativas erradas. No final, questões sobre assuntos já tratados em encontros anteriores, como circuitos formados por portas digitais (E, OU, NÃO) e uma questão subjetiva contendo um pequeno código para que verificassem erros na sua sintaxe (escrita dos códigos).

Analisando os erros e acertos indicados na tabela 1, atentamos, principalmente para as questões 1 e 2. Na questão 1, quase que a totalidade dos alunos (6 de 7 alunos) responderam corretamente. Isso pode demonstrar que eles compreendem o fato da corrente elétrica fluir em um circuito fechado, indo de um polo a outro da bateria. Sendo assim, parecem ter compreendido que é impossível acender uma lâmpada ligada apenas a uma extremidade da fonte de tensão. Por outro lado, a questão 2 mostra a não compreensão do que é e como ocorre um curto circuito, já que todos erraram a questão. A fala do Aluno 7, ao expressar que “a chave S3 não influencia no circuito [...]” pode ser um indício dessa falta de compreensão a respeito das consequências de uma baixa resistência aplicada ao circuito, pois a maioria dos alunos acreditava que mesmo adicionando uma conexão entre as duas extremidades da bateria (fazendo um curto circuito) as lâmpadas continuariam acesas. Para a correção desta questão, fizemos uma longa discussão sobre o assunto e explicamos através de simulações no computador e até mesmo com a montagem de um circuito real com resistor e LED ligado a uma fonte de tensão.

Tabela 1 - Respostas por aluno do questionário 1.

Questionário 1

C - correta E - errada	Múltipla escolha				Subjetiva
	Questão 1	Questão 2	Questão 3	Questão 4	Questão 5 (encontrar 4 erros no código)
Aluno 1	C	E	E	C	2 referências corretas
Aluno 2	C	E	C	E	Em branco
Aluno 3	E	E	C	C	2 referências corretas
Aluno 4	C	E	C	C	1 referência correta e 1 errada
Aluno 5	C	E	C	E	Em branco
Aluno 6	C	E	E	E	2 referências corretas e 1 errada
Aluno 7	C	E	C	C	1 referência correta
Corretas	6	0	5	4	

Fonte: autoria própria.

As outras questões de múltipla escolha tiveram mais acertos do que a questão 2. Na questão 3, cinco alunos responderam corretamente, identificando, possivelmente, uma semelhança com a questão 1, a qual apenas um aluno respondeu errado. Já a questão 4, em que possui conceitos trabalhados nos primeiros encontros, inclusive com uma atividade específica utilizando planilhas eletrônicas, quatro alunos tiveram bom desempenho sobre as portas digitais, escolhendo os valores de entrada que faziam a lâmpada acender. A última questão era subjetiva, na qual os alunos deveriam fazer referências a erros encontrados no código de programação. Nenhum aluno encontrou os quatro erros contidos no código e foram feitas algumas referências incorretas, pois indicaram partes que não continham erros. Sobre esta questão, mesmo se tratando de algo que já tinham visto no primeiro encontro, ainda consideramos que era cedo para que eles fossem capazes de reconhecer tais erros, pois, ainda não estavam tão acostumados com a linguagem e a prática da programação.

5.1.4 Quarto Encontro (30 mai 2016)

No quarto encontro estavam presentes quatro alunos. Começamos revendo o que foi discutido em encontros anteriores e passamos a parte mais prática do curso, em que eles começaram a montar circuitos e programar o Arduino do início ao fim. Desde o começo, eles já começaram a realizar atividades práticas, mas naquela ocasião, deveriam montar os circuitos usando uma *protoboard*¹⁸, conectando os componentes (sensores, atuadores e resistores), verificando os possíveis erros nos códigos e nas ligações, descobrindo a forma correta de usar o material disponibilizado no *kit*. Como atividade prática utilizamos, primeiramente, os exemplos que estavam na apostila. Nestes exemplos, continham todos os passos para a montagem e traziam os códigos de programação para funcionamento correto. Antes de montar os circuitos, usamos um período da aula para falar especificamente do divisor de tensão. Procuramos destacar a importância da atividade do encontro anterior, quando montaram e calcularam o valor de tensão elétrica que entra em um divisor de tensão e o que sai, pois, para utilizar um sensor LDR é necessário saber como ocorre a divisão das tensões entre duas resistências. Depois da explicação complementar desse tema, passamos para a parte prática, em que os alunos deveriam fazer os exercícios que estavam no final do capítulo da apostila. Esses exercícios eram trazidos com aumento gradativo de dificuldades. A sequência da apostila sempre continha uma ordem de conteúdo: teoria, exemplos com “passo a passo” com mais explicações e, por fim, exercícios. Uma das atividades daquele encontro foi o controle de acendimento de um LED pela quantidade de luz incidente no sensor de luminosidade (LDR). Eles deveriam realizar todo o processo de montagem na *protoboard* e criação do código. Tudo era acompanhado por nós e, quando necessário, fazíamos intervenções ajudando os grupos.

Embora o número de alunos tivesse reduzido, foi um encontro bem produtivo em que demonstraram bastante interesse e dedicação nas atividades propostas. Depois de muitos exemplos de demonstração e de códigos prontos para eles reproduzirem, solicitamos que realizassem tarefas e propusemos o que os programas deveriam fazer, junto com a montagem física da parte eletrônica. Para essas montagens, os alunos deveriam compreender de forma clara o funcionamento de um divisor de tensão.

¹⁸ É uma placa de ensaio ou matriz de contato com furos e conexões condutoras para montagem de circuitos elétricos experimentais. A grande vantagem da placa de ensaio na montagem de circuitos eletrônicos é a facilidade de inserção de componentes, uma vez que não necessita soldagem. Disponível em: <https://pt.wikipedia.org/wiki/Placa_de_Ensaio> (Acesso em: 27 mai. 2017).

A maior dificuldade observada até então foi a de que, em todo o início de atividade, os alunos ficavam perdidos e sem saber o que fazer para iniciar ou de que ponto partir. Muitos ficavam parados na frente do computador esperando que nós chegássemos até eles para dizer o que deveria ser feito primeiro. Isso causava um atraso significativo nas atividades, pois sempre tínhamos que rever os passos de como realizar o programa, principalmente como configurar as partes do código, declarar as variáveis, quais funções utilizar, as entradas e saídas necessárias, etc. Ficou relativamente claro que eles não estavam lendo a apostila em casa e nem realizando alguma tentativa fora do período dos encontros. Embora, muitas vezes, tenha sido oferecida a possibilidade de levarem os *kits* para casa, imaginando que em algum momento seriam utilizados, isso não aconteceu; pelo menos não houve evidência disto. Nos encontros seguintes, muitos não faziam mais questão de ficar com os *kits*, talvez porque sabiam que iam ser questionados sobre ter tentado trabalhar algum exemplo ou exercício em casa.

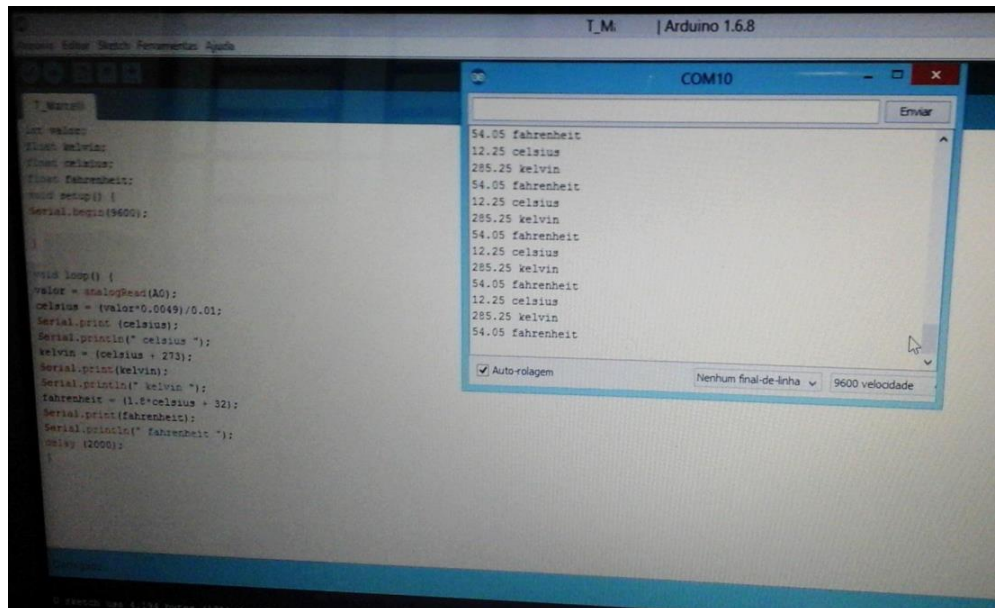
5.1.5 Quinto Encontro (06 jun 2016)

Com a presença de quatro alunos, realizamos o nosso quinto encontro discutindo a utilização de outro tipo de sensor: o LM35 (circuito integrado que fornece uma tensão elétrica proporcional à temperatura). Começamos discutindo a forma de utilizar o sensor, como reconhecer os pinos de ligação e a maneira correta de ligá-lo no Arduino. Em seguida, fizemos com que os alunos montassem no *protoboard* somente o sensor ligado a uma bateria para mostrar que é possível utilizá-lo sem uma placa Arduino, apenas verificando a sua temperatura através da medida da tensão do pino de saída, utilizando um voltímetro. A montagem com o Arduino, primeiramente, foi uma reprodução do exemplo que se encontrava na apostila, o código apenas lia o valor do sensor, transformava em graus Celsius e mostrava no monitor serial do programa. Após realizar esta atividade, propusemos o desafio de refazer um código que mostrasse no monitor serial o valor da temperatura nas três diferentes escalas termométricas: Celsius, Fahrenheit e Kelvin, utilizando a equação de conversão utilizada no estudo das escalas de temperatura. A outra atividade realizada foi adicionar mais de um sensor em combinação, fazendo com que eles utilizassem na programação do Arduino as funções condicionais e os operadores AND e OR, estudadas no primeiro encontro.

Assim como fizemos no 3º encontro, verificando com o voltímetro o divisor de tensão sem utilizar a placa Arduino, pedimos para que os alunos medissem a temperatura com o sensor LM35 utilizando apenas um voltímetro. Como este sensor tem uma saída de tensão

elétrica proporcional à temperatura em graus Celsius (10 mV/°C, por exemplo, se o voltímetro indicar 250 mV a temperatura correspondente será de 25 °C), os alunos deveriam usar o sensor de forma direta, apenas medindo sua tensão de saída. Essa atividade foi interessante porque reforçou a ideia frisada desde o início dos encontros: que o Arduino apenas consegue medir tensão elétrica nas suas entradas analógicas ou digitais. Os alunos 1 e 2, porém, lembraram que mesmo medindo tensão elétrica ainda devemos multiplicar por 0,0049 para obter o valor de tensão pelo Arduino. Aproveitando a colocação dos alunos, reforçamos o porquê da necessidade dessa multiplicação, para que não ficassem dúvidas, já que pareceu confuso o fato de que, mesmo somente lendo tensão elétrica na sua entrada, o valor lido seja impresso como um número inteiro entre 0 e 1023. Em seguida, os alunos realizaram a montagem e programaram um código de leitura da temperatura pela placa Arduino que deveria dar o valor em graus Celsius. Realizada essa atividade, pedimos que modificassem o código e fizessem o programa indicar o valor a ser impresso na tela em três diferentes escala termométricas: Celsius, Fahrenheit e Kelvin, como mostra a fig. 8.

Figura 8 - Imagem do computador do Aluno 6 com o código (à esquerda) e a janela do monitor serial (à direita) mostrando a temperatura nas três escalas.



Fonte: acervo do autor.

As montagens dos circuitos e a programação nesta fase do curso já começavam a ficar mais complexas, pois estavam utilizando mais de um sensor e atuador funcionando em conjunto, como, por exemplo, o sensor de temperatura (LM35) juntamente com o sensor de luminosidade (LDR). Neste caso, o código de programação deveria executar uma função condicionada à verificação de duas grandezas (temperatura e intensidade luminosa). Nestas

atividades, acontecia como nos encontros anteriores: os alunos não sabiam por onde começar. A falta de estudo em casa e o pouco tempo de aula que tínhamos dificultou tanto a programação quanto a montagem dos circuitos na *protoboard*.

5.1.6 Sexto Encontro (20 jun 2016)

Chegamos ao nosso último encontro, com a presença de quatro alunos novamente. Esse dia foi reservado para fazermos um apanhado geral do que foi tratado no curso e um segundo questionário de avaliação (apêndice D), acerca de conhecimentos adquiridos até então. Os alunos responderam individualmente e anotaram suas respostas. Então, pedimos para que formassem duplas e refizessem as questões, só que agora discutindo com o colega para que entrassem em acordo sobre as respostas. Continuamos a montar outros circuitos utilizando vários sensores e atuadores associados e dependentes, dificultado um pouco a construção dos códigos, mas que eram apenas formas mais avançadas de utilizar o Arduino, unindo todo o conhecimento adquirido durante os seis encontros. No fechamento, agradecemos a participação de todos e pedimos para que respondessem um formulário *online* de avaliação da nossa proposta de curso, com perguntas relacionadas à motivação, dificuldades encontradas, relevância na formação, material disponibilizado, método de ensino, sequência dos conteúdos, tempo dedicado ao estudo e a indicação da possibilidade de continuar o curso e assim, aprender mais sobre o assunto.

O questionário 2 trazia perguntas sobre tudo que foi trabalhado no decorrer do curso. Eram no total seis questões, sendo cinco de múltipla escolha e uma contendo oito itens para completar com o número da linha do código de programação a sua função correspondente. Abaixo a tabela 2, referente às respostas individuais e das duplas.

Tabela 2 - Respostas individuais e por duplas, referente ao questionário 2.
Questionário 2.

D U P L A S	C - Correta E - Errada	RI - Resposta Individual RD - Resposta em Dupla											
		Múltipla Escolha										Relacionar a linha correta (8 itens)	
		Q1 RI	Q1 RD	Q2 RI	Q2 RD	Q3 RI	Q3 RD	Q4 RI	Q4 RD	Q5 RI	Q5 RD	Q6 RI	Q6 RD
D1	Aluno 1	C		E		E		C		E		6	6
	Aluno 4	C	C	C	C	E	E	C	C	E	E	6	
D2	Aluno 3	C		C		E		C		E		7	7
	Aluno 7	C	C	C	C	E	E	C	C	E	E	7	
Total corretas		4	2	3	2	0	0	4	2	0	0		

Fonte: autoria própria.

Analisando a tabela 2 vemos que a questão 1 foi respondida corretamente por todos, tanto individualmente quanto pelas duplas. Essa questão estava relacionada à comparação entre portas digitais OU e E com circuitos elétricos. Além de ser colocada no questionário 1 e novamente no 2, esta pergunta foi bastante discutida no primeiro encontro e, pelo que parece, os alunos conseguiram assimilar tal comparação. O mesmo aconteceu com a questão 2 (sobre as portas digitais) que possuiu um erro individual, apenas. Esta questão também estava presente no primeiro questionário. Foi amplamente discutida e reforçada na atividade de simulação com as planilhas eletrônicas. A questão 3 não foi respondida por nenhum aluno ou dupla corretamente. Essa questão sobre o divisor de tensão envolve muitos conceitos físicos relacionados a eletricidade, tais como: tensão, resistência e corrente elétrica. A montagem do divisor de tensão foi muito utilizada em montagens que usavam a *protoboard*. Os alunos não apresentaram muita dificuldade na montagem, mas os conceitos que fundamentam os conhecimentos sobre o circuito não foram assimilados completamente. Talvez deveríamos trabalhar mais os conceitos sobre eletricidade, mas não teríamos tempo suficiente. A falta de conhecimentos prévios relacionados a esta questão, também, foi verificada na aplicação do primeiro questionário e durante as atividades práticas. A questão 5 também não teve acerto por nenhum aluno ou dupla. O que verificamos dessa questão foi que todos os alunos

responderam a alternativa “b”, que poderia ser a correta caso o símbolo do operador OR (||) fosse trocado pelo símbolo do AND (&&). A resposta dos alunos pode ter sido apenas uma confusão de símbolos, já que, na linguagem de programação do Arduino, são diferentes dos utilizados nas fórmulas das planilhas eletrônicas. A correspondência desses símbolos foi discutida nos encontros e constava na apostila elaborada para o curso, mesmo assim não podemos considerar um erro de conceito sobre o operador OU (OR). A última questão do questionário 2 era composta por oito itens, em que os alunos deveriam escrever qual a linha do código de programação apresentado correspondia a função descrita em cada item. O número de acertos dessa questão mostra uma evolução no entendimento das funções de cada parte do código, pois conseguiram relacioná-las corretamente em grande parte. Na tabela 2, vemos que a dupla D1 errou apenas um item e a D2 errou dois.

A fig. 9 mostra um apanhado de imagens das atividades realizadas na escola.

Figura 9 – Imagens das atividades na escola.



Fonte: acervo do autor.

5.1.7 Considerações sobre o Estudo Piloto: achados relativos à intervenção propriamente dita

Durante o Estudo Piloto alguns indícios puderam ser levantados ao longo das atividades realizadas, sendo eles: o interesse por atividades práticas por parte dos alunos; o material instrucional (apostila e *kits* experimentais) pode ser considerado potencialmente significativo e a falta de estudos periódicos fora do horário de aula que demandava a constante retomada de assuntos já tratados.

Com relação ao interesse demonstrado pelos alunos ao realizarem atividades práticas, evidenciou-se, desde os primeiros encontros, a maior aceitação nas atividades em que precisavam montar os circuitos na *proto-board*. As atividades de programação dos códigos no computador, as quais também tratamos como uma atividade prática, apesar da maior dificuldade encontrada, foram muito bem aceitas. Os alunos montavam os circuitos e revisavam as conexões para em seguida construírem os códigos. Muitas vezes, quando os circuitos não estavam montados corretamente, alguns grupos retiravam todos os componentes e refaziam as ligações. Essas ações foram quase que constantes no decorrer do curso, pois era através do êxito no funcionamento desses circuitos, em conjunto com os códigos, parecia que toda a prática fazia sentido para eles, motivando-os para montagens mais elaboradas na sequência das aulas.

O material instrucional utilizado nas aulas (*kits* e a apostila elaborada no decorrer do EP) mostrou-se indispensável na forma como foram organizados os encontros. Os *kits* foram montados somente com os componentes que seriam necessários para realizar as atividades e a apostila favoreceu o andamento do curso por sua estrutura lógica de conteúdos. Os exemplos e exercícios da apostila pareciam facilitar a compreensão, pois os alunos, sempre que tinham dúvidas, recorriam a ela, principalmente nos exemplos, pois possuíam figuras e sequências de montagens, além dos códigos de programação contendo comentários explicando as linhas de comando.

Os indícios de interesse dos alunos pelas atividades dos encontros e do potencial do material instrucional utilizados vêm ao encontro do que foi proposto na Teoria da Aprendizagem Significativa. Segundo Moreira (1999), há duas condições essenciais para que ocorra a aprendizagem significativa, que precisam acontecer simultaneamente no processo de ensino/aprendizagem: o aluno deve manifestar predisposição para relacionar o novo material potencialmente significativo e, o material a ser aprendido deve ser relacionável à estrutura

cognitiva do aluno. Se o aprendiz não possuir o conhecimento mínimo suficiente para relacionar o material a sua estrutura cognitiva, a aprendizagem se torna arbitrária.

A falta de estudos periódicos, fora dos períodos dos encontros ficou evidente pela descontinuidade da expressão dos conteúdos a cada encontro, ou seja, os alunos esqueciam-se de detalhes simples entre um encontro e outro. Como cada encontro tinha um intervalo de uma semana entre o outro, os alunos sempre precisavam de uma revisão sobre os passos iniciais, para realizar as atividades propostas. Estes detalhes “esquecidos” eram passos iniciais sempre iguais que, se fossem revistos fora dos encontros, deveriam ser feitos quase que automaticamente. Outra percepção que favoreceu a afirmação deste indício é que, mesmo sendo incentivados a levarem os *kits* para praticar a programação e montagens em casa, os alunos não faziam questão de ficar com o material. Talvez porque eram questionados no início das aulas seguintes sobre o que tinham feito com os *kits*.

Consideramos que esta falta de envolvimento dos alunos, fora do horário do curso, pode ter ocorrido pelo fato dos alunos estarem acostumados a estudar com foco apenas nas avaliações realizadas em aula, estando muito presente a questão da nota ou da recompensa. Diante disso, avaliamos a possibilidade de realizar a continuidade deste trabalho com um estudo, em condições semelhantes ao EP, focando na investigação de novas metodologias para motivar os alunos a desenvolverem atividades fora do horário do curso. Porém, optamos em investigar a possibilidade de concretizar este objetivo a partir do trabalho na formação dos professores, pois, são os grandes protagonistas na rotina escolar que podem desenvolver atividades de maneira continuada com os alunos.

Pelo fato de se tratar de um conteúdo novo, a dedicação dos alunos apenas nos horários do curso não seria suficiente, mas sim a dedicação como parte da aprendizagem do dia a dia, como ocorre com outras componentes curriculares. Nesta nova perspectiva, a proposta a ser realizada a partir do EP, não era a inserção de um novo componente curricular, mas sim criar possibilidades para professores usarem a programação e automação como uma ferramenta no ensino de Física, para proporem problemas contextualizados e inovadores, como, por exemplos, os citados nos trabalhos de Neto (2013), Rodrigues (2014), Dworakowski (2015) e Filho (2015).

Com essas considerações, o EP possibilitou uma sondagem de como deveria ser elaborado o material e a estrutura de um curso sobre programação e automação na Educação

Básica. Por isso, consideramos o EP como essencial na construção de um curso para professores, pois, através dele, entendemos que este tema deveria ser tratado inicialmente não como os alunos, mas com os professores destes alunos da Educação Básica.

O quadro 2 é um resumo dos conteúdos e atividades que foram realizadas no EP.

Quadro 2 – Resumo das Atividades do Estudo Piloto.

Número de alunos	Atividades (Carga horária total: 20 horas)	Dia
37	Palestra no auditório	06/05
10	Noções básicas de algoritmos e programação Conceito de algoritmos Conceito de programa Representação binária e decimal (conversão) Operadores lógicos (E, OU, NÃO) Portas lógicas representadas por circuitos elétricos Atividade prática: fazendo portas lógicas em planilhas eletrônicas	09/05
6	Conhecendo a plataforma Arduino Programação do Arduino (primeiros códigos) Partes principais da janela do programa Configurações iniciais Variáveis Entradas e saídas digitais Explicação de um exemplo: ligando um LED (atuador) com um botão digital (sensor)	16/05
8	Questionário sobre circuitos e sobre o que foi tratado até então Discussão sobre as questões utilizando simulações Continuação com a programação usando sensores e atuadores Divisores de tensão com resistores Atividades de montagem	23/05
4	Divisor de tensão Sensor de luminosidade (LDR) Montagens com <i>protoboard</i>	30/05
4	Atividades de montagem com o sensor de temperatura (LM35) Montagens com LDR e LM35 juntos	06/06
4	Mais atividades práticas, aplicação de questionário final e fechamento do curso	20/06

Fonte: autoria própria.

5.2 CURSO PARA PROFESSORES E PROFISSIONAIS DA EDUCAÇÃO (CPPE)

Nesta seção serão descritas as atividades realizadas nos módulos I e II do CPPE e será feita uma exposição e discussão dos resultados obtidos nessas atividades.

5.2.1 Modulo I - Formação

A seguir, faremos um relato detalhado de como aconteceram os encontros do CPPE - Módulo I. Analisaremos as características dos professores e profissionais da educação participantes, além de considerações a respeito da predisposição para aprender e sua motivação no decorrer do curso; indícios de aprendizagem de conceitos básicos da programação e utilização da Plataforma Arduino; utilização de sensores e atuadores; montagem e análise de circuitos elétricos simples e avaliação do material de ensino utilizado, de acordo com os objetivos descritos na seção de procedimentos metodológicos.

5.2.1.1 Perfil dos Participantes

No grupo de participantes do CPPE eram variadas as suas formações acadêmicas e seus locais de atuação, sendo todos na Educação Formal. Os participantes serão referenciados neste trabalho como Professor ou Profissional da Educação ou simplesmente pela sigla PPE. Embora ambos sejam profissionais da educação, denominamos como professores aqueles que ministram aulas como docentes e os profissionais da educação aqueles que auxiliam nas aulas ou fazem parte da organização ou manutenção de laboratórios, elaboram experimentos para aulas ou participam de projetos dentro de instituições de ensino, estando diretamente ligados ao ensino formal.

O público atingido pelo curso mostrado no quadro 3 traz uma ideia do quanto era variado o grupo dos 12 PPEs inscritos em relação às suas formações. Com relação às cidades que atuavam, observamos que o curso não teve como inscritos somente pessoas interessadas da cidade de Bagé, mas sim de outras cidades da região: Candiota que fica a cerca de 54 km do campus da Universidade e de Lavras do Sul, que fica em torno de 81 km do campus.

Quadro 3 – Distribuição dos PPEs em relação à formação e local de atuação.

Formação		Cidade de atuação	
Técnica	2	Bagé	8
Graduação	3	Candiota	2
Especialização	4	Lavras do Sul	2
Mestrado	3		

Fonte: autoria própria.

Com relação à formação, área de atuação e experiências profissionais dos PPEs, faremos uma descrição individual para conhecermos melhor o perfil de cada um. Os dados foram retirados do formulário *online* (apêndice E) preenchido na ocasião de inscrição do curso e das observações e conversas realizadas durante os encontros.

PPE 1

Servidor atuante em instituição de Ensino Superior como técnico de laboratório. Possui graduação em Licenciatura em Física e Mestrado em Educação.

PPE 2

Servidor atuante em instituição de Ensino Superior como técnico de laboratório. Possui graduação em Pedagogia e Técnico em Automação Industrial. Tem experiência como professor de Ensino Profissionalizante, ministrando aulas nas disciplinas de eletricidade e eletrônica.

PPE 3

Professor da rede estadual em uma escola de Ensino Fundamental e Médio. Possui graduação em Gestão da Tecnologia da Informação. Ministra aulas de informática e o conteúdo que trabalha é a programação em *Scratch* (movimentação e criação de jogos) e formatação de trabalhos científicos.

PPE 4

Professor da rede estadual em uma escola de Ensino Fundamental e Médio. Possui graduação em Bacharelado em Informática. Ministra aulas na disciplina de Técnicas da Informação para crianças de até cinco anos, trabalhando com jogos computacionais que facilitem o raciocínio lógico dos alunos.

PPE 5

Professor da rede estadual em um curso Técnico em Mecânica Industrial. Possui formação em Técnico em Mecânica. Ministra aulas na disciplina de Técnicas em Mecânica e CNC (comando numérico computadorizado), trabalhando, também, com conteúdos de termodinâmica e automação industrial.

PPE 6

Professor com Formação em Bacharelado em Informática, com especialização em Mídias na Educação. Ministra aulas de Tecnologias Educacionais na formação de professores. Trabalha com conteúdos relacionados ao uso de aplicativos de celular em sala de aula, entre outros. Tem experiência como tutor de cursos a distância.

PPE 7

Professor da rede municipal. Possui graduação em Licenciatura em Matemática e especialização em Metodologia do Ensino de Matemática e Física. Ministra aulas de Matemática e TIC para o Ensino Fundamental. Trabalha com conteúdos gerais de Matemática e com *softwares* como Geogebra e Scratch.

PPE 8

Professor da rede estadual em escola de Ensino Fundamental e Médio. Tem graduação em Biologia e mestrado em Ensino de Ciências. Tem experiências como professor da disciplina de Física e com a utilização do Arduino em sala de aula.

PPE 9

Professor da rede estadual em uma escola de Ensino Fundamental e Médio. Tem graduação em Sistemas da Informação com especialização em Educação Inclusiva. Ministra aulas na disciplina de Tecnologias Educacionais e trabalha com conteúdos de Lógica de Programação, Scratch e internet. Tem experiência com robótica e mídias educacionais e como monitor de curso a distância.

PPE 10

Professor universitário e também atua como profissional da educação em laboratório de informática de escola de Ensino Fundamental. Tem graduação em Sistemas da Informação com mestrado em Engenharia da Computação. Ministra aulas na disciplina de Inteligência Artificial e trabalha com conteúdos de Introdução à Inteligência Artificial, algoritmos, entre

outros. Possui grande experiência em várias linguagens de programação e demais áreas relacionadas às tecnologias, atuando como professor de curso de Sistemas de Informação.

PPE 11

Professor da rede municipal em escola de Ensino Fundamental. Tem graduação em Ciências Biológicas com especialização em Metodologia do Ensino de Ciências. Ministra aulas na disciplina de Ciências e trabalha com conteúdos como Física e Energias para o 9º ano. Este professor realizou inscrição, mas não participou do curso.

PPE 12

Tem formação Técnica em Informática. Atua como professor particular de informática e trabalha com manutenção de computadores e programação em *Visual Basic* e HTML. Tem experiência com desenvolvimento de sites e com suporte a redes. Embora inscrito no curso, não podemos considerá-lo como profissional da educação, pelo fato de não atuar no ensino formal. Este profissional realizou inscrição, mas não participou do curso.

Com base nos perfis individuais, concluímos que a maioria dos PPEs trazia experiências profissionais em suas áreas que não se relacionavam com o uso da Plataforma Arduino, seja como professor ou profissional ligado à área da educação. Portanto, os conhecimentos prévios sobre o Arduino eram muito superficiais para alguns e para outros eram praticamente inexistentes, sendo que o curso seria o primeiro contato com a plataforma, seja com a programação ou com a montagem de circuitos.

A seguir faremos um relato dos encontros presenciais que aconteceram nos dias 17/9, 24/9, 08/10 e 22/10 do ano de 2016, no Laboratório de Física da Universidade Federal do Pampa, Campus Bagé.

5.2.1.2 Primeiro Encontro (17 set 2016)

No primeiro encontro estavam presentes dez PPEs. Inicialmente recebemos os participantes e falamos brevemente dos objetivos da nossa pesquisa e de como seriam propostas as atividades do curso e quais os temas iriam ser tratados. Entregamos a apostila e começamos a apresentação do conteúdo utilizando o quadro branco e apresentação de *slides*. Avisamos aos PPEs que as aulas seriam gravadas em áudio para registro de dados, além do registro fotográfico. No primeiro momento da aula, tratamos de forma mais geral o *hardware* da placa Arduino e falamos sobre suas partes mais importantes, como: as entradas digitais e

analógicas; as saídas digitais; os pinos de alimentação e comunicação com o computador, entre outros. Falamos, também, dos diversos modelos de placas Arduino e mostramos, especificamente, o modelo UNO¹⁹ (que seria utilizado durante todo o curso). Pedimos para que fôssemos interrompidos a cada dúvida que eles tivessem e que poderiam se manifestar a qualquer momento para que a aula ocorresse de maneira mais dinâmica possível. No meio das explicações iniciais, começamos a interagir com os participantes, procurando obter nas suas falas conhecimentos relacionados aos conteúdos do curso. Como já sabíamos o perfil de cada PPE, tentávamos, sempre que possível, relacionar o que estava sendo ensinado a algo que eles conheciam em sua vivência profissional. Isso nos levou a um diálogo mais aberto com cada participante, pois as perguntas e intervenções de cada um eram relacionadas a esses conhecimentos que já adquiriam.

Procuramos não tornar a aula maçante, pois muitos dos PPEs tinham formação na área da informática e o primeiro assunto tratado na sequência da apostila seria o de conceitos básicos de lógica de programação. Para que isso não ocorresse, tentamos envolve-los nos exemplos e explicações e alertamos que talvez seriam tratados assuntos que alguns já conheciam, baseados em sua formação, embora alguns reconhecerem que, mesmo sendo da área da informática, não tinham (ou não lembravam de ter) um bom domínio da parte de programação.

O PPE 10, em especial, tinha um grande conhecimento em lógica de programação e muitas vezes recorremos a ele para explicar ou dar algum exemplo mais elaborado. Esse participante, embora um conhecimento muito grande em programação, não tinha nenhuma experiência em eletrônica ou automação, pois nunca tinha trabalhado com microcontrolador²⁰ ou feito uma montagem de componentes em uma *protoboard*. O PPE 6 relatou que gostaria que fosse tratado como alguém que não tinha conhecimento em programação, embora fosse

¹⁹ O modelo UNO foi disponibilizado nos *kits*, sendo um dos modelos mais utilizados em projetos educacionais por sua versatilidade de uso e custo acessível.

²⁰ O microcontrolador é uma espécie de computador constituído de um processador (CPU), memória de armazenamento de programa, memória para armazenamento de variáveis, além de alguns possuírem periféricos para comunicação, conversão analógico/digital, etc. Como um computador, é programado por meio das chamadas linguagens de programação, como a linguagem C. Disponível em: <<https://fiozero.com.br/microcontroladores-914a59cbf7de>> (Acesso em: 16 fev. 2018). O Arduino UNO possui um microcontrolador ATmega328 que pertence à família AVR da Atmel. Possui 14 pinos de entrada/saída digitais (dos quais 6 podem ser utilizados como saídas PWM), 6 entradas analógicas, um cristal de quartzo de 16 MHz, além de conexão USB e outras características importantes. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>> (Acesso em: 16 fev. 2018)

Bacharel em Informática. Começamos, então a tratar dos princípios básicos da lógica de programação.

Seguindo a sequência da apostila, começamos uma discussão sobre: a diferença entre algoritmos e programas de computador; diferenças entre a um sinal analógico e um sinal digital; o que o Arduino “entende” ou “lê” nas suas portas e as representações dos sistemas binário e decimal, além da conversão entre esses sistemas.

Neste encontro, discutimos sobre as portas lógicas ressaltando a importância do seu entendimento para que depois sejam relacionadas com as entradas e saídas digitais do Arduino. Além da importância dessas portas lógicas, mostramos circuitos com portas digitais AND, OR e NOT (E, OU e NÃO). Essas portas digitais, conforme foi realizado no EP, foram explicadas através de tabelas verdades²¹ e pela analogia com circuitos elétricos com chaves acionando lâmpadas. Alguns PPEs mostraram mais facilidade para entender os circuitos elétricos do que a lógica digital e outros tinham mais facilidade com a lógica. Isso pela diferença de formação de cada participante do curso. Os oriundos de cursos de formação relacionados à informática conheciam os operadores lógicos e as tabelas verdade, já outros tinham mais compreensão com a análise de circuitos elétricos que eram utilizados nas analogias que foram propostas. A união entre as duas formas de trabalhar o conceito de portas lógicas facilitou essa diferenciação entre os conhecimentos de todos.

Passamos para a parte da aula em que começamos a mostrar a plataforma Arduino (IDE de programação e a placa) com todas as suas principais partes e características. Nesse ponto o CPPE começava a se tornar mais prático do que teórico, pois eles começariam a interagir com o material dos *kits*. Começamos a falar sobre a programação do Arduino e a importância de entender as diferenças entre as portas analógicas e digitais da placa. Após uma explicação de como deve ser feito os códigos para que o programa entenda como também as formas que se equivalem na escrita dos códigos. Os PPEs começavam a interagir e alguns começavam a relembrar conceitos de programação ou de outros que já tinham trabalhado em alguma ocasião. As formas de expressar nos códigos os valores de entrada de cada tipo de portas, que muitas vezes é difícil de entender no primeiro momento, foram ficando mais claras e também as diferenças entre estas, como pode ser visto na fala do PPE 5 – que não possuía um conhecimento prévio relacionado ao assunto – durante explicação: “[...] só para entender, a porta digital trabalha com 0 e 5 (ligado ou desligado) e a analógica também

²¹ Tabelas utilizadas para verificar todos os casos possíveis de um argumento (MONTARI, 2001).

trabalha de 0 a 5 volts, só que com todos os valores de 0 a 5.” A afirmação está correta em parte, porque a porta analógica trabalha de 0 a 5 volts, mas não com todos os valores e sim com a subdivisão de 1023 partes deste 5 volts (ou seja, aproximadamente 0,0049V). Como podemos ver, não é possível ter todos os valores do intervalo citado, mas apenas valores múltiplos de 0,0049V. Esse entendimento é importante quando se quer utilizar sensores analógicos, pois, as portas analógicas têm como base a leitura de uma tensão elétrica que é expresso apenas por valores inteiros de 0 a 1023. A conversão da tensão elétrica em um valor que deve ser utilizado nos códigos (que o Arduino “entende”) se dá pela eq. 1. O resultado deve ser arredondado para um valor inteiro.

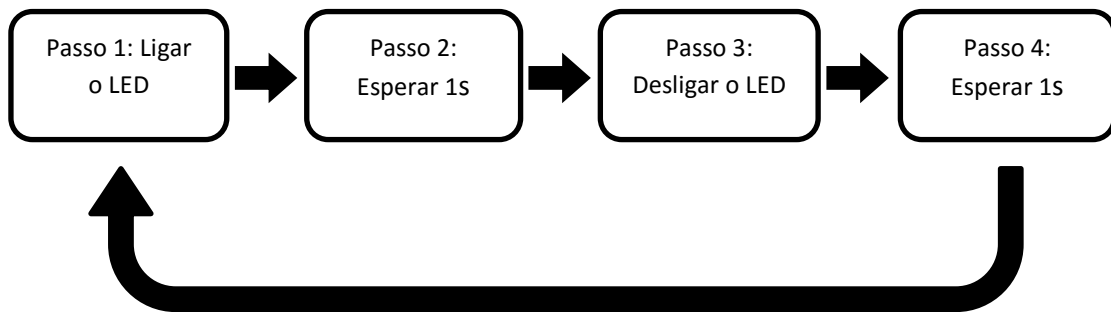
$$Valor_{(0-1023)} = \frac{V_{(volts)}}{0,0049} \quad \text{Eq. 1}$$

Na sequência da aula voltamos a falar sobre as portas digitais, depois de termos explicado as diferenças entre portas digitais e analógicas. As digitais foram tratadas através de analogias com circuitos elétricos, uso de tabelas verdades e o estudo de circuitos digitais com mais de um tipo de porta. Esta estratégia foi utilizada no EP com os alunos e teve bom retorno nos questionários respondidos por eles. Então, pesamos ser uma boa estratégia para facilitar o entendimento dos PPEs. Como já foi dito acima, por terem formação em diversas áreas, cada forma de tratar o assunto contribuiu de forma diferente para cada participante, uns mais com a parte de analogias com os circuitos elétricos e outros com as tabelas verdades e lógica das portas. Durante todo o tempo sempre procuramos deixar claro o objetivo de cada exemplo e no que isso ajudaria quando começássemos programar o Arduino e montar os circuitos.

Ainda na primeira parte do encontro, passamos a mostrar a janela do programa (IDE) do Arduino. Explicando os principais comandos que seriam utilizados e as funções de configuração e reconhecimento da placa pelo computador. A apostila também trazia essa informação, mas especificamos tudo e fizemos um apanhado de todo o processo desde abrir, digitar o código, fazer a compilação e carregar o código na placa. Depois de garantir que todos os computadores estavam corretamente conectados à placa Arduino, pedimos para que eles abrissem um exemplo chamado de “*blink*” que é nada mais do que um pequeno código de exemplo que o programa já possui e faz com que um LED conectado ao pino 13 do Arduino ligue por um segundo e depois desligue por um segundo repetido esse ciclo indefinidamente. Por ser um código muito simples, serviu para explicar cada parte separadamente e mostrar que para executar uma tarefa (pisca o LED) todos os passos devem ser predeterminados, de outra

forma pode não funcionar como queremos. O diagrama da fig. 10 mostra os passos que esse o programa deve conter para executar a simples tarefa do exemplo mostrado.

Figura 10 – Sequência de passos do exemplo “blink”.



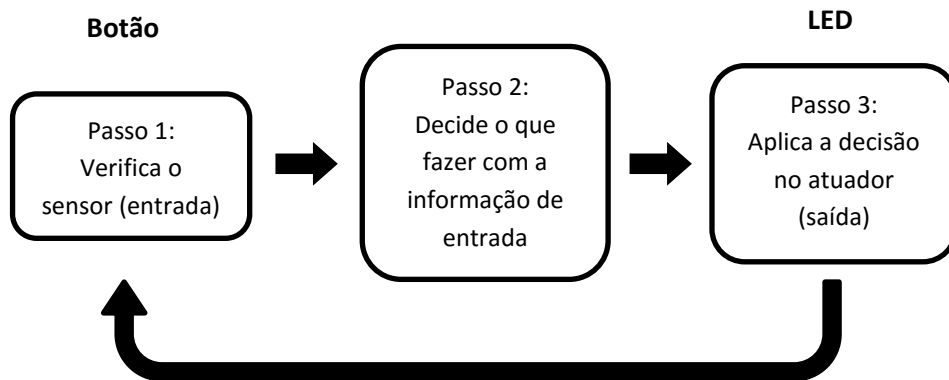
Fonte: autoria própria.

A partir desse exemplo, alguns PPEs começaram a trocar parâmetros de tempo e configurar outros pinos como saídas e nelas conectando LEDs. Isso foi feito sem que fosse pedido, o que mostra que eles já se encontravam mais seguros com o programa, pois ao trocar valores no código eles já estavam, de alguma forma, construindo seus próprios programas. Essa iniciativa, às vezes bem-sucedida às vezes não, fortaleceu o diálogo entre as duplas, mostrando compreensão básica do que estava sendo transmitido no primeiro momento. Notou-se que os conhecimentos prévios trazidos se relacionavam diretamente com a atitude de realizar testes em cima do exemplo: os oriundos de formação ligada à informática se sentiam mais aptos a mexer no código e outros mais a montar circuitos com LED e resistor na *protoboard*. Naquele momento, surgiram muitas dúvidas com relação à programação e montagem dos circuitos, o que foi muito proveitoso porque oportunizaram esclarecimentos no decorrer do encontro. Com certeza, essas dúvidas surgiriam novamente durante os próximos encontros, pois eram suas primeiras tentativas de prática com o Arduino e, obviamente, seriam sanadas com mais detalhamento na evolução das atividades.

No final do encontro, mostramos a diferença entre um atuador e um sensor, além das principais funções utilizadas nos códigos para usá-los. A proposta inicial de estudar um pouco como funciona as portas digitais e definir o conceito de entradas e saídas teve como principal objetivo o uso desses sensores e atuadores. Os atuadores, como o próprio nome já diz, atuam no meio externo e relacionamos com as saídas da placa Arduino, já os sensores relacionamos com as entradas da placa. Isso foi nossa tentativa de dar um sentido à forma como estruturamos nosso material instrucional para o curso. Com a ideia que este material tenha

uma forma organizada e que faça sentido durante o processo de aprendizagem dos PPEs. O diagrama da fig. 11 (exemplo de um LED sendo acionado por um botão) mostra que novos elementos iam sendo incluídos, o que era apenas um simples exemplo de programação se torna uma integração entre a programação e objetos físicos (sensores e atuadores).

Figura 11 – Sequência de passos para ligar o LED (atuador) com botão (sensor). Agora não depende apenas da programação feita, mas também do estado do sensor (botão externo).



Fonte: autoria própria.

5.2.1.3 Segundo Encontro (24 set 2016)

Neste encontro estavam presentes sete PPEs. Dando continuidade ao curso, tratamos sobre a programação do Arduino, principalmente o uso de variáveis, tipos de dados e funções utilizadas para construção dos códigos. As funções principais como as de “ler” estados de sensores e acionar atuadores. Também as funções de tempo, estruturas de controle e operadores de comparação, aritméticos e lógicos. Para realizar as atividades que foram propostas nesse encontro os PPEs começavam a integrar a programação com a montagem de circuitos elétricos. Para a maioria dos participantes, com exceção ao PPE 10, começar a construir um código, mesmo com auxílio de exemplos que se encontravam na apostila, era uma tarefa desafiadora e não muito simples. Como nós acompanhávamos todos durante o processo, observamos que muitos preferiam modificar os exemplos ao invés de digitar um código novo. A primeira tarefa desse dia foi a de montarem o circuito representado na fig. 11 em que um botão quando pressionado acendesse um LED. O circuito foi desenhado no quadro e eles tinham apenas a atribuição montá-lo na *portoboard*. Essa montagem teve que ser precedida de uma explicação rápida de como utilizar a *protoboard*. Para os que não tinham muita ou nenhuma experiência com esse tipo de montagem demorou mais para compreenderem a forma correta. Outros já precisaram de menos tempo, pois já eram familiarizados. Aqueles que dominavam as montagens não estavam seguros com a criação dos

códigos. Exemplo dos PPEs 2, 3, 4, e 7 que não tiveram quase nenhuma dificuldade na montagem. Os PPEs 1, 5 e 10 encontraram mais dificuldades na montagem. Com relação à construção dos códigos, praticamente todos os participantes apresentaram uma grande dificuldade. Apenas o PPE 10 não apresentou nenhuma dificuldade na programação por ser o mais experiente. Suas dificuldades estavam apenas no entendimento relacionado às montagens dos componentes e as noções de eletrônica e eletricidade (como o em conceitos de corrente, resistência e tensão elétrica).

Depois de montarem alguns exemplos simples e testarem outros códigos, modificando os que já tinham sido criados, os PPEs foram convidados a responder um questionário (apêndice C). O objetivo principal deste questionário era de levantarmos informações a respeito dos conhecimentos de circuitos elétricos de todos os participantes presentes. Durante o primeiro encontro, foi tratado o assunto de circuitos de forma muito superficial, mas seria interessante entendermos o nível de conhecimento de cada PPEs sobre esse tema. O questionário também trazia questões relativas ao que já tinha sido visto nos dois primeiros encontros: sobre portas lógicas (circuito lógico contendo portas AND e OR) e verificação de um pequeno código de programação, em que deveriam identificar se havia erros e quais eram.

Analisando os erros e acertos na tabela 3, vemos que todos acertaram as questões 1 e 3, que eram questões sobre circuitos muito simples contendo lâmpadas e chaves. Para responder era preciso entender como a corrente elétrica se comporta no circuito (fig. 12 e 13). Na questão 1, a alternativa correta é a que indica a não existência de corrente elétrica no circuito com a chave S1 aberta. Na questão 3, a alternativa correta indica que se a lâmpada A queimar, as lâmpadas B e C não acenderão.

Tabela 3 – Respostas por PPE do questionário 1.

Questionário 1

C - correta E - errada	Múltipla escolha				Subjetiva
	Questão 1	Questão 2	Questão 3	Questão 4	Questão 5 (encontrar 4 erros no código)
PPE 1	C	E	C	E	3 referências corretas
PPE 2	C	C	C	C	1 referência correta
PPE 3	C	C	C	C	3 referências corretas
PPE 4	C	C	C	C	3 referências corretas
PPE 5	C	C	C	C	2 referências corretas
PPE 7	C	C	C	E	3 referências corretas
PPE 10	C	E	C	C	4 referências corretas
Corretas	7	5	7	5	

Fonte: autoria própria.

Figura 12 – Circuito da questão 1.

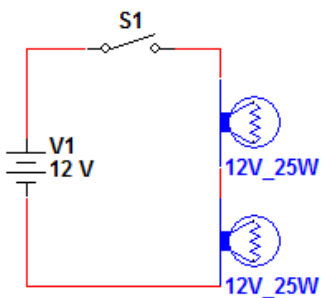
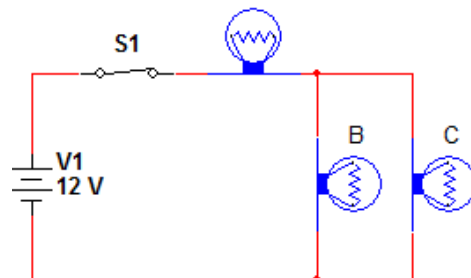


Figura 13 – Circuito da questão 3.

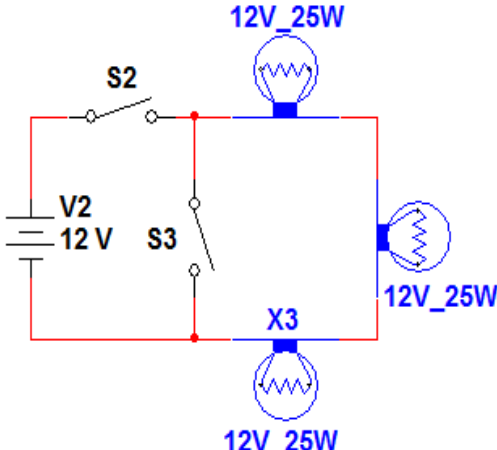


Fonte: autoria própria.

A questão 2 (fig. 14) possuía uma chave que, quando fechada, causaria um curto-circuito e dois PPEs erraram. O PPE 1 respondeu que se fosse fechada a chave as lâmpadas acenderiam com o brilho mais fraco. O PPE 10 respondeu que ao fechar a chave as lâmpadas acenderiam com seu brilho normal. Sabemos que o curto-circuito está relacionado com o conceito de resistência e corrente elétrica, pois a baixa resistência elétrica causa o aumento brusco e elevado da corrente elétrica de um circuito. No momento que as chaves S2 e S3

forem fechadas haverá um curto-circuito por S3, o que não foi considerado pelos dois participantes que erraram a questão.

Figura 14 – Questão 2 do questionário 1.



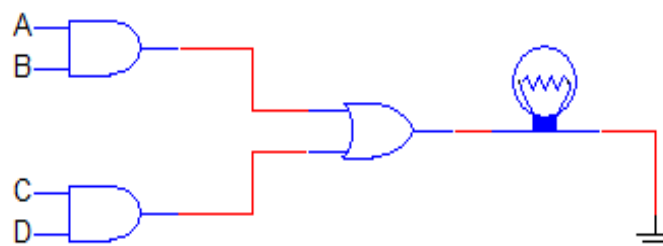
A figura ao lado é um circuito formado por uma bateria, duas chaves liga/desliga e três lâmpadas. Se fecharmos a chave S2 e S3 o que aconteceria?

- As três lâmpadas ficarão acessas com seu brilho normal.
- Haverá um curto-circuito.
- As lâmpadas ficarão acessas, mas com um brilho muito fraco.
- Somente uma lâmpada ficará acessa.

Fonte: autoria própria.

A questão 4 era sobre as portas lógicas AND e OR (E e OU) com duas entradas formando o circuito lógico da fig. 15, em que a saída era representada por uma lâmpada ligada (1) ou desligada (0). A resposta esperada era a combinação de valores nas entradas A, B, C, D para se obter na saída o valor 1 (lâmpada acessa). Apesar de ter sido bastante discutida durante o primeiro encontro, nem todos acertaram. Podemos dizer que é uma questão de um conteúdo novo para alguns, principalmente para aqueles que não tiveram contato com programação ou eletrônica em suas formações.

Figura 15 – Circuito lógico da questão 4 contendo duas portas AND e uma porta OR.



Fonte: autoria própria.

Na questão 5, os participante deveriam identificar os erros no código mostrado (fig. 16). Era um código bem simples e continha quatro erros (1º erro: a variável “Led”, linhas 8 e 10, aparece escrita diferente das linhas 1 e 4; 2º erro: na linha 4, a palavra “INPUT” deve ser trocada por “OUTPUT” por se tratar de uma saída digital; 3º erro: na linha 11, falta um “;” no final da linha de comando; 4º erro: este foi considerado um erro porque o código estava impresso através de uma imagem capturada da tela que fez com que desaparecesse²² o “{” da linha 7). A maioria dos participantes conseguiu fazer mais de duas referências aos erros encontrados. O PPE 2 encontrou apenas um erro no código, talvez porque esse tipo de questão, somente visualizando um código de programação, não seja algo trivial para quem não tem o costume de programar ou não é da área da informática. Somente pela prática é possível identificar esse tipo de erro, já que são detalhes que podem fazer o programa não funcionar. O PPE 10, por exemplo, que usa a programação na sua atuação como professor conseguiu identificar todos os erros que continham no código. O erro que consideramos mais importante (2º erro), porque se relaciona com a configuração do pino para ligar o LED, deve ser configurado como saída, mas foi trocado por entrada. Esse erro foi identificado por quatro PPEs, entre eles o 1, 2 e 5 os quais não possuíam quase nenhuma experiência em programação.

Figura 16 – Código da questão 5 contendo quatro erros.

```

Arquivo  Editar  Sketch  Ferramentas
sketch_may22a $
int Led = 13;

void setup() {
pinMode(Led, INPUT);
}

void loop() {
digitalWrite(led,1);
delay(1000);
digitalWrite(led,0);
delay(1000)
}

```

Fonte: autoria própria.

²² É uma característica do IDE quando o cursor esta localizado em uma chave ele mostra a outra chave que marca o bloco de comando para avisar onde ele começa e onde termina. Ao colocar o cursor em outro lugar do código a chave “{” da linha 7 voltaria a aparecer.

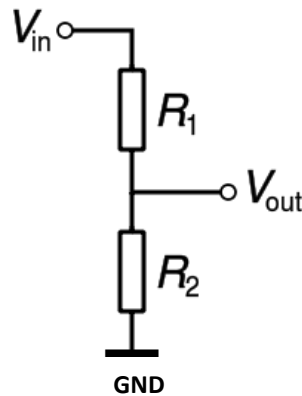
5.2.1.4 Terceiro Encontro (08 out 2016)

Neste encontro estavam presentes nove PPEs. Continuamos a sequência do curso, mas com uma abordagem bem mais prática começando, então, a montar pequenos projetos com a plataforma Arduino. Esses pequenos projetos traziam praticamente todos os conceitos principais que os PPEs deveriam entender até o final do curso. Nesta fase, em que eles deveriam colocar em prática o que tinham sido visto até então, precisamos lembrar algumas coisas importantes, pois este encontro teve um intervalo grande em relação ao último. Embora realizados aos sábados, não conseguimos ter os encontros em todas as semanas na sequência. Por isso, precisamos usar algum tempo a mais lembrando partes importantes que já tinha sido vistas nos dois primeiros encontros. Alguns PPEs ficaram com os *kits* e tentaram refazer alguns exemplos que foram feitos durante o encontro. Mas a nossa expectativa com relação a praticarem em casa não foi totalmente atingida, pois os PPEs, nesse caso, se comportaram de forma semelhante aos alunos do Estudo Piloto, ou seja, mesmo levando o material os participantes não demonstraram que tivessem utilizado, pois quando perguntado sobre o que tinham praticado normalmente relataram falta de tempo para fazer. Neste terceiro encontro realizamos uma revisão discutindo os principais passos para executar a montagem e programação de um pequeno projeto que são basicamente:

- i. Saber o que se pretende que o sistema faça;
- ii. Identificar que tipos de sensores e atuadores serão utilizados (discretos ou contínuos);
- iii. Entender como esses sensores e atuadores funcionam (se dependem de circuitos);
- iv. Decidir que tipos de portas serão utilizadas (digitais ou analógicas);
- v. Saber quais as funções, estruturas de controle e operadores poderão ser utilizados.

Discutimos sobre o circuito divisor de tensão, pois o seu entendimento é de suma importância quando utilizamos, principalmente, alguns tipos de sensores analógicos. A fig. 17 representa o circuito divisor de tensão.

Figura 17 – Circuito divisor de tensão.



Fonte: autoria própria.

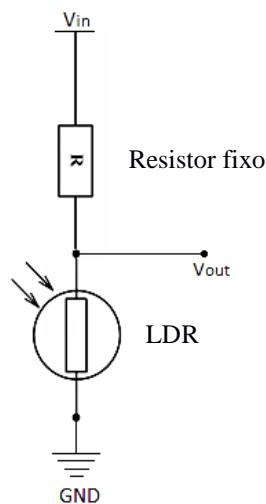
$$V_{out} = \frac{R_2}{(R_1 + R_2)} V_{in} \rightarrow (R_1 \neq R_2) \quad \text{Eq. 2}$$

$$V_{out} = \frac{1}{2} V_{in} \rightarrow (R_1 = R_2) \quad \text{Eq. 3}$$

O divisor tem uma tensão de saída (V_{out}) proporcional aos valores da tensão de entrada (V_{in}) e R_1 e R_2 (eq. 2). Quando R_1 é igual a R_2 utilizamos a Eq. 3. Um ponto importante que procuramos ressaltar é que essa tensão de saída é medida em relação ao “terra” do circuito (*ground* ou GND). As portas do Arduino medem sempre a diferença de potencial entre o centro do divisor e o “terra”, ou seja, entre V_{out} e GND. Os participantes tiveram certa dificuldade na interpretação desse circuito, pois a maioria não era de áreas relacionadas a conceitos de eletricidade como diferença de potencial, resistência e corrente elétrica. Mesmo o PPE com graduação em Física também teve dificuldades. O que mais mostrou entendimento sobre o assunto foi o PPE 2 que já ministrou aulas de eletrônica. Como entendemos que havia dificuldades diversas, pedimos para que os participantes realizassem uma tarefa de montar um divisor de tensão com dois resistores escolhidos aleatoriamente. Em seguida descobrir qual a tensão de saída por duas formas: a primeira calculando o valor utilizando a Eq. 2 e logo medindo a saída do divisor utilizando um voltímetro. Depois de verificar o valor, eles deveriam ligá-lo à porta analógica e mostrar na tela do computador o valor da tensão medida pelo Arduino. Todos conseguiram montar e encontrar o valor próximo ao que foi calculado e medido. O que foi importante nessa atividade é que os PPEs conseguiram identificar no circuito divisor de tensão em cima de qual resistor a tensão deveria ser medida e, conseqüentemente, como esse circuito deveria ser ligado a porta do Arduino para obter a leitura.

A próxima atividade realizada envolveu a substituição de um dos resistores por um sensor de luminosidade (LDR), um resistor variável que tem sua resistência elétrica inversamente proporcional à intensidade de luz que incide sobre ele. Os PPEs deveriam utilizar um LDR para ligar um LED indicativo e uma lâmpada através de um relé quando as luzes da sala fossem desligadas. Essa atividade envolvia um circuito divisor de tensão com sensor (fig. 18) e um código que verificava a intensidade de luz da sala e que ligava o LED e o relé e, por conseguinte, a lâmpada. A atividade envolvia uma porta analógica e duas digitais, além da montagem dos circuitos na *protoboard*. Não era um código muito complicado, mas exigia uma montagem com componentes mais elaborada.

Figura 18 – Divisor de tensão com LDR. A variação da intensidade de luz varia a tensão V_{out} .



Fonte: autoria própria.

Todas as duplas conseguiram montar e executar a atividade com êxito. Alguns PPEs notaram o seguinte efeito durante os primeiros testes: quando as luzes da sala eram apagadas a lâmpada do circuito ligava normalmente e em seguida desligava sem que as luzes da sala fossem acessas. Muitas vezes a lâmpada ficava “piscando”. Ao notar esse efeito e discutindo o que poderia estar ocorrendo, alguns conseguiram explicar o comportamento do circuito. Acontece que quando a lâmpada era acionada no momento em que as luzes da sala eram apagadas, a própria lâmpada iluminava novamente o sensor LDR, causando instabilidade no sistema. O correto era o sensor estar longe da lâmpada. Essa explicação dada por alguns participantes era correta e demonstrava que já havia um entendimento bem razoável sobre a programação e o funcionamento do circuito. Se os PPEs não estivessem entendendo o processo que estavam executando, não conseguiriam explicar de maneira tão clara o efeito

observado. Ao colocar a lâmpada longe do sensor o problema foi resolvido. “[...] é por isso que nos postes o sensor fica em cima!” (PPE 5).

No final do deste encontro, propusemos aos PPEs trazerem para o próximo, ideias que gostariam de aplicar em suas escolas, considerando que o módulo II do curso tratou da aplicação de atividades nos locais onde atuavam. Solicitamos para que trouxessem as propostas, mas, caso não tivessem, poderíamos ajudar com uma lista de exemplos que selecionamos para serem aplicados em aulas de Física. Manifestamos que os participantes poderiam propor ideias de qualquer área, não especificamente envolvendo a Física. No primeiro momento, todos os PPEs que atuavam em escolas se mostraram interessados em aplicar algo. Essas atividades seriam acompanhadas por nós, como parte de nossa pesquisa.

5.2.1.5 Grupo Focal 1

Ainda no encontro do dia 08/10, realizamos o primeiro de dois grupos focais com os PPEs presentes. Essa atividade visava obter mais informações sobre a aprendizagem dos conteúdos do curso, além de uma autoavaliação sobre o grau de motivação dos participantes e sobre as suas percepções a respeito de seus desempenhos nas atividades. Algumas questões foram elaboradas para as atividades dos grupos focais:

- i. Sobre experiências de alguma atividade diferenciada seja como professor ou como aluno.
- ii. Qual foi a motivação para fazer o curso?
- iii. Quais as expectativas quanto a levar esses conhecimentos aos seus alunos?
- iv. Sobre a importância do tema do curso tanto nas escolas quanto no dia a dia.
- v. Quais suas expectativas de aprendizagem sobre o tema proposto ao final do curso?
- vi. Quais as maiores dificuldades que você está encontrando durante o curso?
- vii. O que você considera ter menos dificuldade no curso, baseado nos seus conhecimentos já existentes?

As questões acima não foram discutidas na mesma ordem, nem colocadas da mesma forma listada, mas apenas serviram de base para direcionar o grupo focal.

Perguntando ao grupo qual a motivação para fazer o curso e se tinham a intenção de levar os conhecimentos para suas escolas o PPE 9 respondeu: “Eu acho que a ideia é levar para a sala de aula. Levar os conceitos de programação e do próprio equipamento.” Sua ideia

era adaptar a programação, que já era trabalhada com seus alunos através do programa *Scratch*, para o uso do Arduino. O PPE 2 disse que não tinha nenhum conhecimento sobre Arduino, mas que os alunos com quem trabalha sempre perguntavam se ele conhecia e quando soube do curso viu oportunidade começar a entender sobre a plataforma Arduino. Nas palavras dele: “[...] pegar o mais básico para depois a gente corre atrás.” Também comentou sobre a importância da integração Escola-Universidade e sobre sua participação em projetos de extensão da Universidade que visam esse objetivo. O PPE 5 falou que no primeiro contato com Arduino teve o interesse de aprender por *hobby*, mas viu alguns trabalhos de iniciativa própria de seus alunos e percebeu a possibilidade de levar para a escola como um incentivo a mais para as aulas e, também, para utilização em projetos mais completos na apresentação em feiras de ciências que participa com os alunos. O PPE 4 disse ter conhecido o Arduino no nosso curso e tinha a intenção de levar os conhecimentos para a escola em que atuava. O PPE 3 comentou sobre sua atuação ensinando o *Scratch* e sobre a tentativa sem muito sucesso da inclusão da robótica em sua escola através do Projeto Butiá²³. O PPE 8 relatou sobre as experiências utilizando o Arduino durante sua aplicação no mestrado, mas que gostaria de se aprofundar mais na programação. Entende que precisa aumentar seus conhecimentos para poder usar em aula. “[...] para trabalhar com o aluno hoje, tu tens que tentar mostrar para ele como é que funciona aquilo ali. Não é só ligar e fazer um LED piscar.” Mesmo utilizando o Arduino em outras oportunidades, se achava pouco seguro na parte da programação. Até conseguia entender uma sequência de códigos prontos, mas não escrever um do início. O PPE 7 se identificou com a fala de cada um dos colegas e atribui sua motivação para realizar o curso por estar utilizando tecnologias na sua aplicação de mestrado. O PPE 10 relatou sobre a importância de conhecer a parte da montagem e automação com Arduino. Na sua atividade profissional, estava presentes somente a lógica de programação e sentia falta da parte da eletrônica e viu no curso a possibilidade de utilizar o Arduino aliado à experiência com a programação nas suas aulas. Suas prioridades eram relacionadas aos conhecimentos de circuitos eletrônicos e aplicações práticas na utilização do Arduino.

5.2.1.6 Quarto Encontro (22 out 2016)

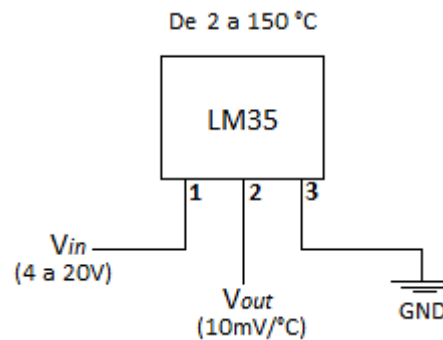
Este foi o último encontro do nosso curso e estavam presentes oito PPEs. Os conteúdos do curso estavam sendo finalizados neste encontro, além da avaliação final

²³ O Projeto Butiá, criado e desenvolvido pelo grupo de pesquisa em Inteligência Artificial e Gestão de Redes (MINA) do Instituto de Computação da Faculdade de Engenharia da Universidade da República do Uruguai (Fing-Udelar), com base na robótica livre, tem o objetivo de incorporar a Robótica nas escolas públicas.

realizada neste dia, ainda fizemos mais um grupo focal e continuamos a montar exemplos presentes na apostila, reforçando os pontos importantes vistos nos encontros anteriores.

O último sensor a ser estudado, conforme a programação do curso, foi o sensor de temperatura LM35. Esse sensor tem uma boa versatilidade para se utilizar em circuitos, principalmente, utilizando microcontroladores como o da placa Arduino. O LM35 é um circuito integrado com saída analógica de tensão proporcional à temperatura e pode medir, em uma configuração básica (fig. 19), dentro de uma faixa de 2 a 150 °C. O sensor quando alimentado com a tensão de 4 a 20V tem uma saída de 10mV/°C. Por todas essas características de facilidade na montagem, além de ser um sensor de baixo custo, se torna muito útil para realizar montagens para aulas de Física ou Química ou outras ciências.

Figura 19 – Circuito típico para medir temperatura com o LM35.



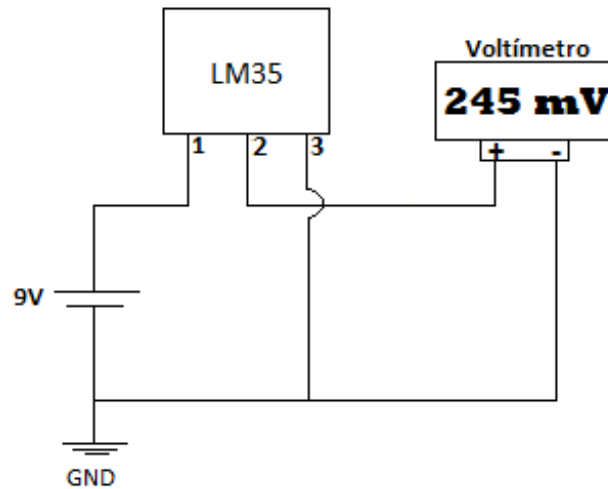
Fonte: autoria própria.

Na primeira atividade, pedimos para que montassem um circuito para medir a temperatura ambiente, mas não utilizando a porta analógica do Arduino e sim usando um voltímetro. Queríamos mostrar como funciona a sensor, pois não é preciso da placa microcontrolada podendo apenas montar o circuito da fig. 20. No exemplo da figura temos a tensão medida no voltímetro de 245mV. Usando a Eq. 4 temos o valor em graus Celsius.

$$Temperatura(^{\circ}C) = \frac{Tens\tilde{a}o\ pino\ 2(V)}{10(mV/^{\circ}C)} \quad Eq. 4$$

$$Temperatura(^{\circ}C) = \frac{245mV}{10\ mV/^{\circ}C} = 24,5^{\circ}C$$

Figura 20 – Circuito para medir temperatura utilizando uma bateria e um voltímetro.



Fonte: autoria própria.

Em seguida pedimos para que medissem a temperatura com o Arduino, escrevendo na tela do computador o valor em graus Celsius. Para isso teriam que escrever o programa, ligar o sensor à porta correta, etc. Todos os passos aprendidos teriam que ser seguidos. Demos um tempo para todos fazerem e ficamos auxiliando em alguma dúvida que fosse aparecendo. Nessa atividade os PPEs não tiveram muita dificuldade, pois era muito parecida com as outras já realizadas. O que causou mais dúvida foi em relação à forma de mostrar a temperatura de forma correta. A saída do valor da temperatura não é direta, ou seja, se mandamos escrever na tela a variável do pino analógico de entrada teremos apenas um número de 0 a 1023 que corresponde à temperatura. Todos os cálculos deveriam estar presentes dentro do código para, primeiro, passar o valor inteiro para um valor de tensão e depois transformar essa tensão para o valor em grau Celsius. Alguns optaram por colocar toda a equação em uma variável apenas (Eq. 6), outros usaram uma variável com o cálculo da tensão (Eq. 5) e outra variável com o cálculo da temperatura (Eq. 4).

$$V_{(volts)} = Valor_{(0-1023)} \cdot 0,0049(V) \quad \text{Eq. 5}$$

$$Temperatura = \frac{Valor_{(0-1023)} \cdot 0,0049(V)}{10(mV/^{\circ}C)} (^{\circ}C) \quad \text{Eq. 6}$$

A cada nova atividade os PPEs tinham dificuldade em escrever os códigos, mas mostravam entender como fazer, que portas usar, tipo de variáveis, entre outros. Com exceção do PPE 10, todos demoravam a descobrir os erros de sintaxe nos códigos. Essa dificuldade era

mais a falta de prática em programar do que o não entendimento de como utilizar os sensores. Pelas falas e perguntas dos participantes, observamos que o conceito do programa, ou seja, a base do entendimento lógico era, de certo modo, bem entendida por eles. O que ainda não era bem dominado por todos era a forma correta de escrever como o programa “entende”, ou seja, compilar os códigos sem erros.

5.2.1.7 Grupo Focal 2

O segundo grupo focal foi realizado para obter mais informações a respeito do que aprenderam e se o curso foi válido para ampliação de seus conhecimentos e nas suas atividades como professor ou profissional da área da educação. Perguntamos aos PPEs que vieram da cidade de Lavras do Sul se acharam proveitoso o curso, já que tiveram custos em vir até a Universidade. O PPE 4 respondeu que “com certeza! O meu problema foi sempre a programação [...]”. Para ele, fazer a programação, montar, conseguir testar e visualizar atraiu muito mais do que simplesmente programar e ver apenas o resultado na tela. Se referindo às suas experiências como aluno do curso de informática. Também o motivou continuar a aprender mais e montar projetos na escola e também em casa, como forma de se aprofundar um pouco mais. O PPE 9 disse que aprendeu bastante sobre eletrônica e utilização do Arduino, mas ainda achava complicado ensinar esse tipo de programação para os alunos. Entende que seria mais fácil utilizar a programação em blocos para ensinar. O PPE 10 falou sobre a sua aprendizagem na parte da eletrônica, pois só em conhecer os componentes já foi importante para ele que nunca tinha visto nada além de programação. Gostaria de inserir uma parte sobre Arduino no plano de ensino da componente curricular da qual ministra aulas no curso de graduação em informática. O PPE 1 disse que o curso serviu muito na sua atividade profissional, principalmente na montagem de circuitos e o uso da *protoboard*. O PPE 8 acredita que alguns exemplos feitos nos encontros são bem viáveis para ser feitos com seus alunos. Considera que seu aprendizado foi muito favorável com relação à programação e a parte de circuitos serviu para lembrar muito conceitos que havia esquecido.

Como última avaliação presencial realizadas com os PPEs, propomos que eles respondessem o questionário 2 (apêndice D). Ainda faríamos um questionário final *online*, como última atividade à distância. O questionário 2 trazia questões parecidas com as do questionário 1, mas com o nível de dificuldade bem maior. Isso porque gostaríamos de testar o conhecimento adquiridos pelos participantes através de questões mais completas que envolviam tanto a lógica como os conceitos de circuitos elétricos.

Analisando os erros e acertos da tabela 4, vemos as questões 1 e 2 com apenas dois erros em cada uma. Tratavam de reconhecer e entender o funcionamento de portas lógicas. Essas questões, que não são consideradas difíceis, tiveram um bom índice de acertos.

Tabela 4 – Respostas por PPE do questionário 2.

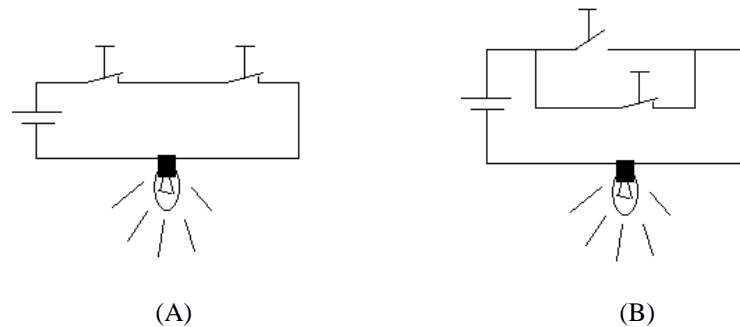
Questionário 2

C – Correta E – Errada	Múltipla Escolha					Relacionar as linhas do código (8 itens)
	Questão 1	Questão 2	Questão 3	Questão 4	Questão 5	Questão 6
PPE 1	C	C	C	E	E	5
PPE 3	E	C	E	C	E	1
PPE 4	C	E	E	C	C	7
PPE 5	E	E	E	C	E	5
PPE 8	C	C	C	C	C	8
PPE 9	C	C	E	C	C	7
PPE 10	C	C	E	C	C	8
Corretas	5	5	2	6	4	

Fonte: autoria própria.

Na questão 1, em que apenas dois erraram (PPEs 3 e 5), notamos que eles compreenderam o funcionamento dos circuitos (A) e (B) da fig. 21, mas trocaram o tipo de porta lógica que eles representam. Os dois confundiram o circuito (A) afirmando ser a representação da porta OR (OU). O PPE 5 falou que no seu entendimento havia duas respostas corretas. Ele entendeu que na figura (B) apenas uma das chaves garantia o acendimento da lâmpada, mas reconheceu que confundiu o nome da porta, pensando que fosse a representação de uma AND (E).

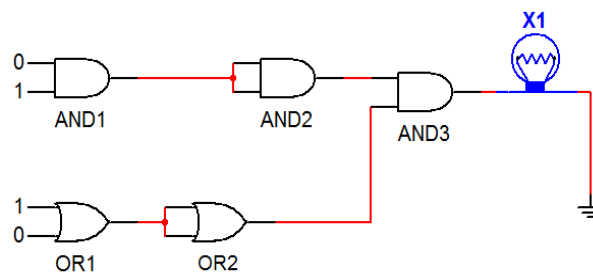
Figura 21 – Circuitos da questão 1.



Fonte: autoria própria.

Na questão 2, também com apenas dois erros (PPEs 4 e 5), era parecida com a questão do questionário 1, mas com um nível de dificuldade maior e mesmo assim teve bom índice de acertos. A questão trazia um circuito composto por portas digitais (fig. 22) em que a alternativa correta era a que fazia a lâmpada acender.

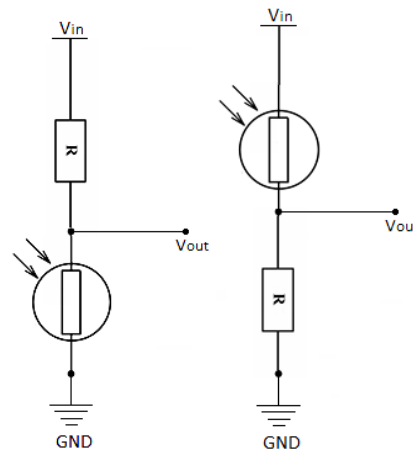
Figura 22 – Circuito digital da questão 2.



Fonte: autoria própria.

A questão 3 teve um índice de acerto muito baixo, apenas dois PPE acertaram. Tratava do entendimento sobre o divisor de tensão e consideramos uma das questões mais complexas, que para ser respondida corretamente deveria ser bem compreendida. O divisor de tensão com um resistor fixo e um LDR foi mostrado de duas formas, de acordo com a fig. 23. Para responder deveriam compreender como se comporta o circuito nos dois casos e verificar cada alternativa. Uma questão que envolvia certos conceitos físicos como resistência, corrente e tensão elétrica, além da compreensão correta da forma de como se comporta o sensor. Os dois PPEs que acertaram eram de áreas afins com os conceitos citados. O PPE 1 formado em Física e o PPE 8 professor de Física e formação técnica relacionado a área da eletrônica.

Figura 23 – Divisor de tensão com LDR e resistor fixo em duas configurações possíveis.

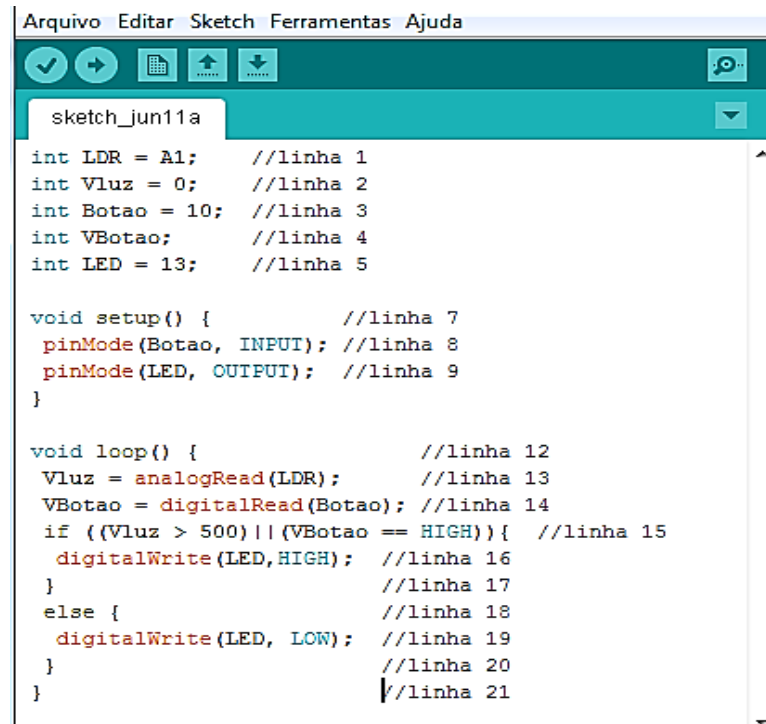


Fonte: autoria própria.

A questão 3 afirmava que o LDR diminui sua resistência conforme aumenta a incidência de luz sobre ele. Mesmo assim houve muita confusão com a questão, pois a alternativa correta deveria afirmar que, no circuito da direita, se aumentássemos a incidência de luz no LDR a tensão de saída (V_{out}) iria aumentar. Para entendermos melhor essa questão devemos compreender o funcionamento do sensor e como as quedas de tensão nos dois componentes se comportam. Se a resistência do LDR diminuir (consequência do aumento de luminosidade) a tensão V_{in} ficará dividida proporcionalmente, ou seja, maior resistência, maior tensão elétrica. No circuito da esquerda temos o contrário desse raciocínio, em que o aumento da luminosidade causará uma diminuição da tensão de saída V_{out} . O que podemos observar nessa questão é que dos cinco PPEs que erraram, quatro escolheram a alternativa que afirmava que: *mesmo variando a resistência do LDR a tensão de saída (V_{out}) será sempre a metade da tensão de entrada (V_{in}), pois é um divisor de tensão*. Essa alternativa é incorreta, pois está totalmente contra o conceito do divisor de tensão. Podemos avaliar que eles não compreenderam corretamente os circuitos, pois foi a questão que tivemos menos acertos.

A questão 4 se baseava em um código exemplo (fig. 24) que perguntava a função de duas linhas: `pinMode(Botao, INPUT)` e `pinMode(LED, OUTPUT)`. Apenas um PPE errou, todos os outros responderam corretamente que tinham a função de configurar os pinos de entrada e saída digital.

Figura 24 – Código para responder as questões 4, 5 e 6.



```

Arquivo Editar Sketch Ferramentas Ajuda
sketch_jun11a
int LDR = A1; //linha 1
int V luz = 0; //linha 2
int Botao = 10; //linha 3
int V Botao; //linha 4
int LED = 13; //linha 5

void setup() { //linha 7
  pinMode(Botao, INPUT); //linha 8
  pinMode(LED, OUTPUT); //linha 9
}

void loop() { //linha 12
  V luz = analogRead(LDR); //linha 13
  V Botao = digitalRead(Botao); //linha 14
  if ((V luz > 500) || (V Botao == HIGH)) { //linha 15
    digitalWrite(LED, HIGH); //linha 16
  } //linha 17
  else { //linha 18
    digitalWrite(LED, LOW); //linha 19
  } //linha 20
} //linha 21

```

Fonte: autoria própria.

A questão 5 era um pouco mais complexa e teve quatro acertos, tratou-se do uso do operador lógico OR (OU) dentro de uma estrutura de controle *if* (SE). A ideia geral do código apresentado era de ligar um LED se um botão fosse pressionado ou se a intensidade de luz no LDR fosse acima de 500, as principais linhas que respondiam a questão eram:

```

if ((V luz > 500) || (V Botao == HIGH)) {
  digitalWrite(LED, HIGH); }

```

Pela lógica do *if* essas linhas significam que o LED será ligado caso a luz no LDR seja maior que 500 **ou** o botão seja pressionado (a função OR é representada pelo símbolo “||” no código). Qualquer uma das opções ligará o LED independente da outra. A maioria que errou a questão não entendeu a independência que o operador OR dá ao código. A dependência dos dois só seria correta se fosse colocado o símbolo && (AND). A alternativa correta era a que diziam: *se V botao estiver em estado alto, o LED ligará, independente do valor de V luz*. O PPE 9 chamou atenção dizendo: “Olhem o exercício 1!” Isso porque na fig. 21 mostra o circuito com as chaves em paralelo, o que, em nossa avaliação, a analogia feita entre operadores lógicos e circuitos foi compreendida pelo PPE 9, tanto que a utilizou para exemplificar a questão que estava sendo discutida.

Na questão 6 (fig. 25) os PPEs deveriam relacionar a função com a linha do código correspondente. A média de acertos foi alta com exceção ao PPE 3, que nos surpreendeu com apenas um acerto. Não sabemos ao certo se ele não entendeu a questão ou realmente não conseguiu responder. Pois durante o curso ele não demonstrou que não estava acompanhando as atividades.

Figura 25 – Questão 6 do questionário 2.

6) Complete com o número da linha correspondente:

- I) Liga o LED (linha ____);
- II) Declara uma variável de um pino analógico (linha ____);
- III) Configura um pino como entrada digital (linha ____);
- IV) Coloca em uma variável o valor lido no pino analógico (linha ____);
- V) Coloca em uma variável o valor lido no pino digital (linha ____);
- VI) Usa um operador lógico **OU** (linha ____);
- VII) Configura um pino como saída digital (linha ____);
- VIII) Declara a variável que receberá o estado do botão (linha ____);

Fonte: autoria própria.

Alguns PPEs não realizaram o questionário 2, pois não estavam presentes na hora da aplicação. Os que realizaram os dois questionários eram os mais assíduos no curso, pois estavam presentes nos dois turnos (manhã e tarde).

5.2.1.8 Atividade a Distância: Questionário 3 (Online)

Finalizando o módulo I do CPPE, enviamos para os participantes um questionário *online* na forma de formulário digital (apêndice F), em que as respostas eram identificadas individualmente. Continha questões que envolviam todos os conteúdos tratados nos encontros e que se assemelhavam com os dos questionários 1 e 2. Essa última atividade tinha o objetivo de concluir a avaliação tentando descobrir ainda mais sobre as principais facilidades e dificuldades encontradas pelos PPEs.

As respostas dos PPEs das questões de 1 a 6 estão na tabela 5. O questionário 3 trazia questões com nível de dificuldade maior que os questionários anteriores.

Tabela 5 – Respostas por PPE do questionário 3 (questões de 1 a 6).

Questionário 3

C – Correta E – Errada	Q1	Q2	Q3	Q4	Q5	Q6
PPE 1	E	E	C	E	E	E
PPE 2	C	C	C	C	C	C
PPE 3	E	E	C	C	E	E
PPE 7	E	E	C	C	C	C
PPE 8	E	E	C	C	C	C
PPE 9	C	C	C	C	E	E
PPE 10	E	C	E	C	E	E
Corretas	2	3	6	6	3	3

Fonte: autoria própria.

As respostas dos PPEs das questões 7 e 8 estão descritas na íntegra no quadro 4. Foram classificadas da seguinte forma:

- **Totalmente Correta (TC)** – Quando a resposta traz todos os conceitos de forma correta e clara e não omite conceitos importantes.
- **Parcialmente Correta (PC)** – Quando a resposta traz mais conceitos corretos do que incorretos, mas omite conceitos importantes.
- **Parcialmente Incorreta (PI)** – Quando a resposta, mesmo trazendo conceitos corretos, traz mais conceitos incorretos e omite outros importantes.
- **Totalmente Incorreta (TI)** – Quando a resposta traz conceitos incorretos e nenhum conceito correto.

A questão 7 perguntava: *Qual a principal diferença entre a entrada analógica e a entrada digital do Arduino. Quando devo usar uma ou outra?*

A fig. 26 mostra a questão 8.

Figura 26 – Questão 8 do questionário 3 (*online*).

Faça um breve comentário de cada linha de comando do código abaixo e explique o que o programa faz. *

```
int pinLed = 8;
int botao = 10;
int estadoBotao = 0;

void setup() {
  Serial.begin(9600);
  pinMode(pinLed, OUTPUT);
  pinMode(botao, INPUT);
}
void loop() {
  estadoBotao = digitalRead(botao);
  digitalWrite(pinLed, estadoBotao);
  Serial.print(estadoBotao);
}
|
```

Fonte: autoria própria.

Quadro 4 – Respostas por PPE da questão 7 e 8.

	Questão 7	Avaliação	Questão 8	Avaliação
PPE 1	Quando precisar obter valor de uma variável, cálculo ou medida, usa-se a entrada analógica.	TI	Estabelece que o Led está no pino 8, o botão no pino 10, e o estado do botão no pino 0; iniciar o programa, ler em 9600; quando modo do pinLed estiver ligado, o botão está desligado. repetir comando; ler botão em modo digital, escrever o comando do Led conforme o estado do botão, obter valor do estado botão;	PI
PPE 2	Na entrada digital assume apenas dois valores: alto ou baixo, 1 ou 0 enquanto que na analógica os valores podem variar de 0 a 1023.	PC	O led é uma saída a entrada botão esta no pino 10 o estado do botão sempre parte da posição desligado. Após leitura do botão é inscrito na saída Led, se zero led desligado, se 1 liga led	PC
PPE 3	Entrada digital - serve para analisar se o pino especificado está liga/desligado, pois lê 0 ou 5V. Entrada analógica - serve para fazer analogias de valores; pois o Arduino lê, através de um circuito de divisor de tensão, valores em bits que multiplicados por 0,0049V se chegará ao valor real de Volts entre 0 e 5 que a placa do mesmo emite.	TC	variáveis: pinLed = 8; // armazena pino 8 botao = 10; // armazena pino 10 estadoBotao = //armazena pino 0 void setup Serial.begin (9600); //inicial a comunicação serial pinMode (pinLed, OUTPUT); // pinLed saída, atribui as finalidades das variáveis. pinMode (botao, INPUT); // botao entrada Void loop estadoBotao = digitalRead (botao); // analisa o estado da variável botão, e armazena os dados que o mesmo possui. digitalWrite (pinLed, estadoBotão); // carrega a variável pinLed, com a carga que está na variável estadoBotão. Serial.print (estadoBotao); // Imprime no monitor, o valor do estadoBotao pino 10.	PC

PPE 7	Na entrada analógica os valores variam de 0 a 1023, ou seja poderíamos ter um valor mais preciso. Na entrada digital existem somente dois estados, alto ou baixo 0 ou 5 volts.	PC	<pre>//define LED como pino 8 //define BOTÃO como pino 10 //define estado BOTÃO como baixo //inicia a serial do arduino //configura o pino 8 como saída //configura o pino 10 como entrada // lê o valor do botão alto ou baixo // liga ou desliga o led de acordo com o estado do botão //imprime o estado do botão</pre>	TC
PPE 8	Entradas digitais só leem valores alto (HIGH)=5v, ou baixo (LOW)=0. Entradas analógicas leem valores intermediários entre 0v e 5v, mas não mostra diretamente o valor em volts e sim um número inteiro de 0 a 1023.	PC	<p>Duas primeiras linhas: declara as variáveis e diz em quais pinos estão ligadas;</p> <p>linha 3: zera o valor da variável estadobotão</p> <p>void setup (): inicia a configuração principal;</p> <p>Serial.begin: inicia e define a velocidade da comunicação serial;</p> <p>linhas 6 e 7: estabelece a configuração dos pinos;</p> <p>void loop(): roda o que estiver dentro deste bloco repetidamente;</p> <p>linha 9: verifica o estado da variável botão;</p> <p>linha 10: liga o pinLed, de acordo com a variável estadobotão;</p> <p>linha 11: mostra no monitor serial o estado do pino 10.</p>	TC
PPE 9	Porta digital apresenta dois estágios, ligado (1) ou desligado (0) sendo usado para representar no mundo físico uma ação de fechar/abrir, acender/apagar. Já, quando tratamos de ações contínuas do mundo físico, o melhor é a utilização de portas lógicas que estabelecem um contínuo registro das modificações do mundo físico, exemplo variação da temperatura em um dia.	PC	<p>declara pino 8</p> <p>declara valor de botão = 10</p> <p>declara valor inicial da variável "estadoBotao"</p> <p>início da comunicação serial</p> <p>configuração do "pinled" para saída</p> <p>configuração de botão para entrada</p> <p>começo do laço:</p> <p>Lê estado do botão e armazena na variável "estadoBotao"</p> <p>Liga pinLed de acordo com a variável estadoBotao</p> <p>Apresenta no monitor serial o estado do botão</p> <p>Ligará o led caso o botão for pressionado.</p>	PC
PPE 10	As digitais só assumem dois valores: HIGH e LOW. As analógicas podem assumir 1024 valores.	PC	Lê o estado do pino 10, atribui o valor ao pino 8 e imprime na tela o estado do pino 10.	PI

	Dependendo da necessidade da aplicação, se usa uma ou outra.			
Avaliação: TC – Totalmente Correta PI – Parcialmente Incorreta PC – Parcialmente Correta TI – Totalmente Incorreta				

Fonte: autoria própria.

5.2.2 Modulo II – Aplicação nas Escolas

O modulo II é a parte de aplicação do curso em que os PPEs colocariam em prática os conhecimentos adquiridos e utilizando o material disponível para o curso como os *kits* Arduino e outros componentes que fossem necessários. Dos oito PPEs que participaram do modulo I (formação) apenas dois realizaram atividades nas suas escolas: o PPE 8 e PPE 9. Embora outros participantes se mostraram motivados a realizar essa etapa no primeiro momento, somente esses PPEs realizaram de fato. Essas atividades feitas com seus alunos foram acompanhadas, concluindo a parte final de nossa pesquisa. Faremos uma descrição e uma análise da forma como ocorreram essas aplicações.

5.2.2.1 Aplicação do PPE 9

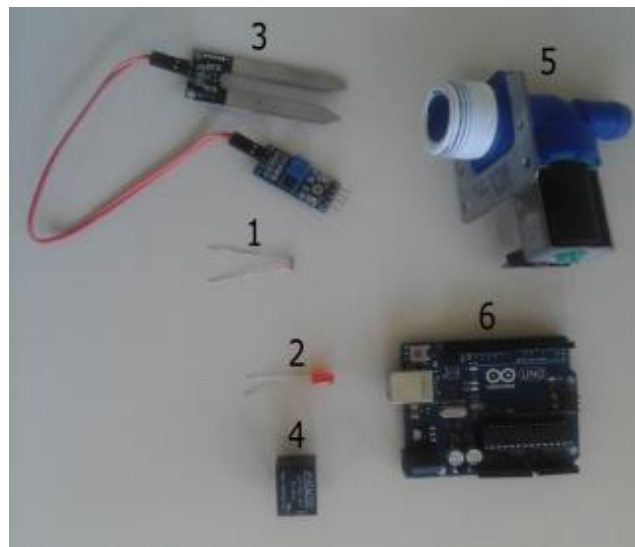
A aplicação ocorreu em uma Escola Estadual de Ensino Médio da cidade de Bagé/RS. Os participantes eram alunos do 8º ano do Ensino Fundamental da componente curricular de Tecnologias Educacionais ministrada pelo PPE 9. A aplicação ocorreu em alguns encontros na segunda quinzena do mês de novembro de 2016. Como foi realizada no final do ano, era muito difícil contar com todos os alunos da turma, mas as aulas que foram acompanhadas tinham presentes entre quatro e seis alunos. Envolvidos no projeto final permaneceram quatro alunos.

Os alunos tinham como conteúdo de aula a utilização do programa *Scratch* em que a programação é feita a partir de blocos lógicos. Os alunos tinham uma noção de lógica utilizando esse programa, mas nunca tinham utilizado a plataforma Arduino. A ideia do projeto que seria implementado pelo PPE 9 era de construir um sistema de irrigação de uma horta de forma automatizada. A horta deveria ser irrigada automaticamente respeitando algumas condições de umidade e intensidade luminosa para liberar a água. Essas condições deveriam ser programadas pelos alunos e deveriam ser utilizados sensores e atuadores para executar as funções. Como os alunos já estavam familiarizados com o *software Scratch*, o PPE 9 propôs utilizar o Arduino com o *software S4A*²⁴ (*Scratch for Arduino*) que é um *Scratch* modificado que interage com a placa Arduino.

²⁴ S4A é uma modificação do *Scratch* que permite programação simples da plataforma Arduino. Ela provê novos blocos para gerenciar sensores e atuadores conectados ao Arduino. O foco principal deste projeto é atrair pessoas para o mundo da programação. Disponível em: http://s4a.cat/index_pt.html (Acesso em: 05 nov. 2017).

Os sensores utilizados no projeto foram o LDR e um sensor de umidade de solo. Ambos utilizam as portas analógicas do Arduino para leitura. Como atuadores foram utilizados um LED indicador e um relé responsável para ligar a válvula solenoide para água, utilizada comumente em máquinas de lavar. Na fig. 27 vemos todos os principais componentes utilizados no projeto e na fig. 28 os primeiros testes do sistema de irrigação.

Figura 27 – LDR (1); LED (2); sensor de umidade do solo (3); relé (4); válvula solenoide de máquina de lavar (5) e placa Arduino (6).



Fonte: acervo do autor.

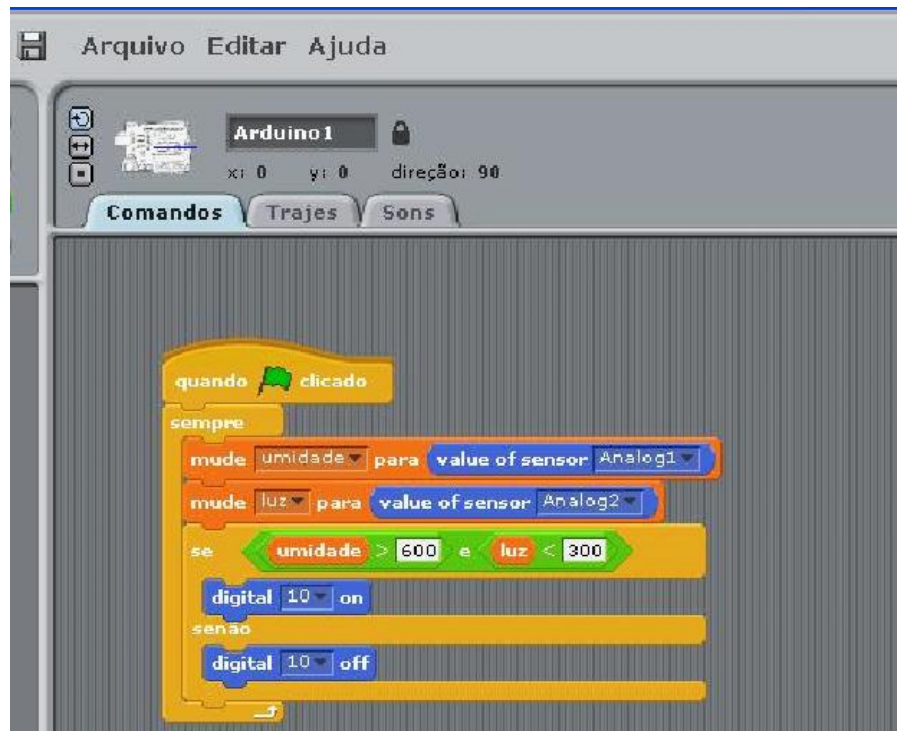
Figura 28 – À esquerda, os primeiros testes com os sensores e à direita a maquete da horta construída pelos alunos.



Fonte: acervo do autor.

No dia 25 de novembro os alunos participaram de uma mostra que aconteceu na escola apresentando o projeto da horta. O pôster feito para apresentação se encontra no anexo A. Na fig. 29, temos o programa feito no S4A para comandar o sistema de irrigação.

Figura 29 – Janela de programação do S4A com o código em blocos do sistema de irrigação feito pelos alunos.



Fonte: construção dos alunos com o PPE 9.

Durante a mostra, os alunos colocaram em garrafas PET um pouco de terra para simular as condições da horta. Uma com terra bem seca, outra com terra um pouco úmida e a terceira com terra bem úmida (fig. 30). A luz do solar era simulada utilizando a lanterna do celular para demonstração.

Figura 30 – Garrafas contendo terra para demonstração utilizando sensor de umidade ligado ao Arduino.



Fonte: acervo do autor.

Os alunos apresentaram a todos da escola seu projeto, ficando responsáveis pela explicação de como funcionava o projeto e por sua demonstração (fig. 31). O PPE 9 conseguiu motivar os alunos durante todo tempo e até mesmo aqueles que eram mais envergonhados e não estavam participando muito se mostraram muito empolgados com o projeto. Acompanhando as explicações que eles davam sobre a programação, sobre o funcionamento do sistema, sobre os sensores e demais partes do projeto mostrou que, mesmo com pouca experiência, é viável incentivá-los a aprender mais sobre programação e montagem de projetos com o Arduino. Mesmo que eles já tivessem conteúdos de programação no currículo escolar, pareceu relevante o fato de reconhecerem na prática o que haviam estudado só na teoria.

Figura 31 – Alunos apresentando o trabalho à comunidade escolar.

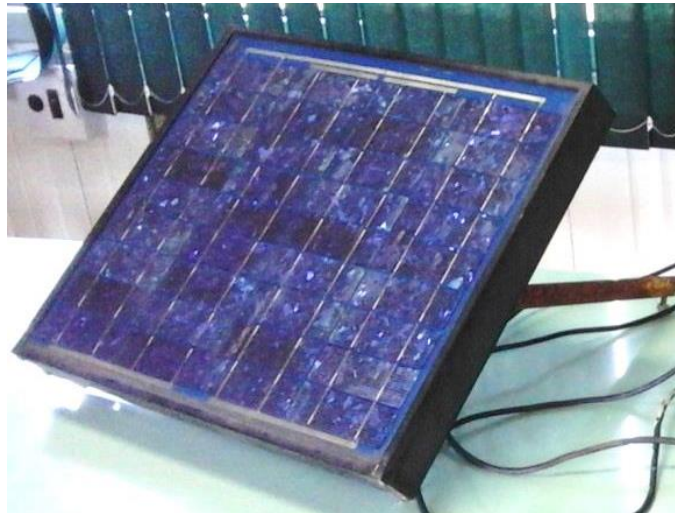


Fonte: acervo do autor.

5.2.2.2 Aplicação do PPE 8

A aplicação ocorreu em uma Escola Estadual de Ensino Médio da cidade de Candiota/RS onde o PPE 8 ministra aulas de Física. Os participantes eram alunos do 3º ano do Ensino Médio. A aplicação com os alunos ocorreu em dois encontros no mês de dezembro de 2016 e nesta data havia poucos alunos na escola. Foram convidados a participar da aplicação três alunos do turno da manhã. O PPE 8 propôs uma atividade que envolvia conteúdos que já tinham sido vistos por eles como: potência, tensão, resistência e corrente elétrica. Na aplicação, foi realizada a verificação de potência elétrica de um painel solar fotovoltaico (fig. 32) que a escola havia recebido de doação por uma empresa local. Como não sabiam quais as especificações do painel, o PPE 8 sugeriu fazer um sistema utilizando o Arduino que fosse capaz de descobrir a potência elétrica de geração desse painel solar.

Figura 32 – Painel solar fotovoltaico.

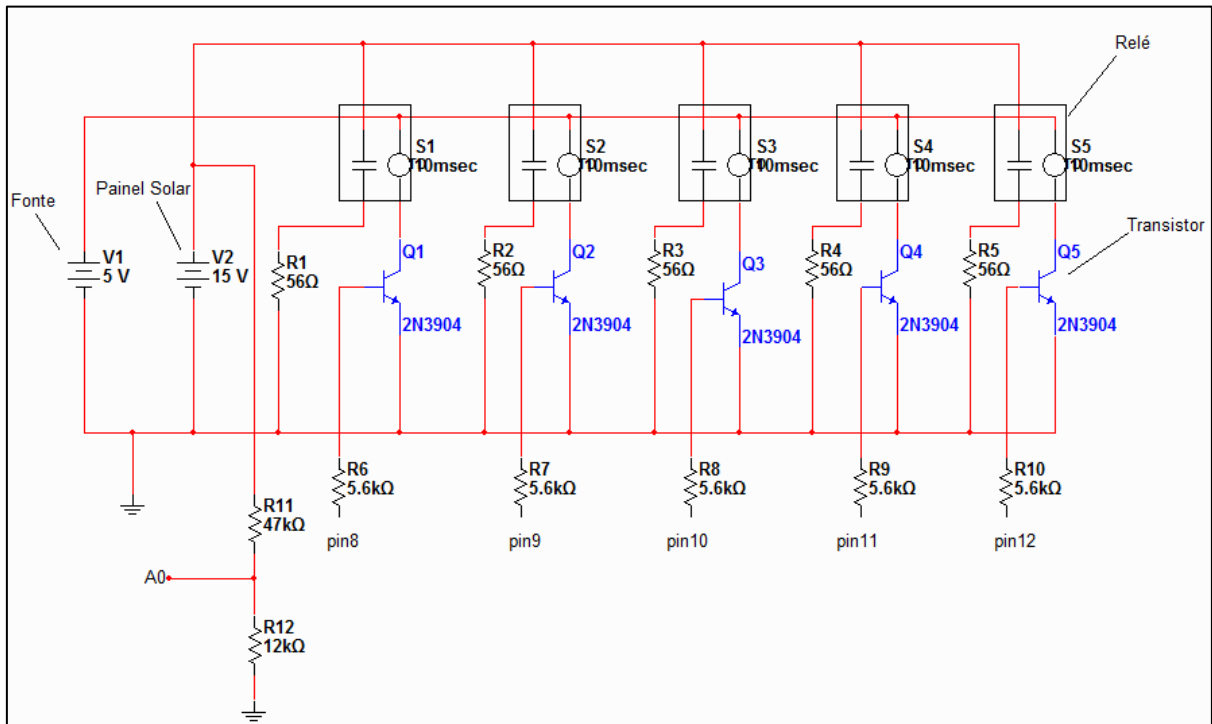


Fonte: acervo do autor.

Foram realizados alguns encontros antes de apresentar aos alunos porque existiam alguns detalhes no projeto que deveriam ser bem organizados e planejados, com antecedência, já que possuíamos pouco tempo para a aplicação. O sistema para medir o painel era composto de: um circuito eletrônico (fig. 33 e 34) montado na *protoboard* e um código de programação para a placa Arduino (apêndice G). O circuito continha resistores, relés e transistores. O código de programação tinha a função de acionar os relés (pelos pinos de saída do Arduino) em um tempo preestabelecido para que fosse adicionada mais carga ao painel solar. Pelas portas de entrada analógica eram feitas as medidas de tensão elétrica do painel solar, calculando também, através programa, a potência elétrica dissipada na carga (resistores

ligados sequencialmente em paralelo). As informações de tensão e potência elétrica eram mostradas em tempo real na tela do computador, via monitor serial. Todos esses dados foram tabelados para depois serem analisados.

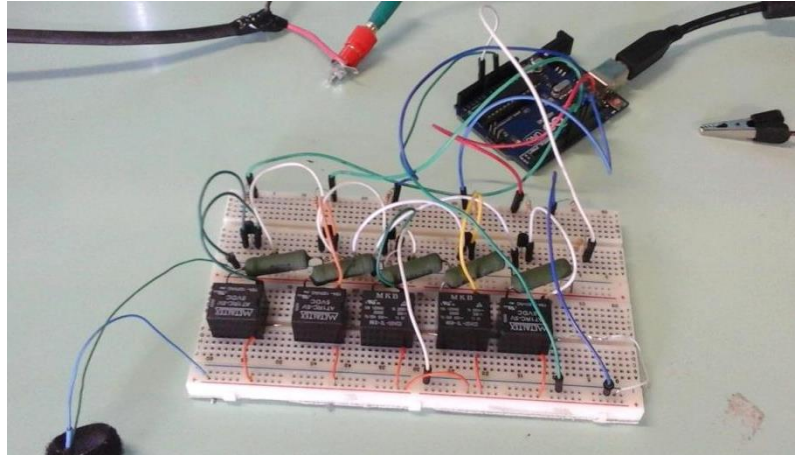
Figura 33 – Esquema eletrônico do circuito utilizado para medir tensão e potência elétrica do painel solar. A tensão elétrica era medida pelo divisor de tensão ligado à porta A0, e as portas de 8 a 12 acionavam os relés via transistores Q1 a Q5.



Fonte: autoria própria.

O primeiro encontro foi para montar o circuito eletrônico e fazer os primeiros testes no painel solar. O dia estava limpo, sem nuvens o que ajudou para fazer as medias. O painel solar foi direcionado o mais diretamente para o Sol e no horário perto do meio-dia. O circuito da fig. 34 era para que as medidas fossem automatizadas. Os alunos apenas iriam se preocupar em mover manualmente o painel durante o passar do tempo, para que ficasse sempre direcionado diretamente para o Sol, garantindo a máxima concentração de raios solares sobre o painel.

Figura 34 – Circuito montado com relés para adicionar carga sequencialmente.



Fonte: acervo do autor.

No segundo encontro foram feitas as coletas de dados utilizando o sistema projetado (fig. 35). Nas primeiras medidas o tempo de acionamento das cargas era de 1 min, ou seja, a cada 1 min mais um resistor (de 56Ω) era adicionado em paralelo ao painel solar, fazendo com que a carga total aumentasse. Os cinco resistores de carga do circuito eram ligados sequencialmente pelo Arduino através dos relés. Os valores de tensão elétrica e potência elétrica sobre a carga eram medidos a cada segundo e registrados pelo computador. A segunda medida foi realizada com tempo de 20 min e os valores eram igualmente registrados a cada segundo.

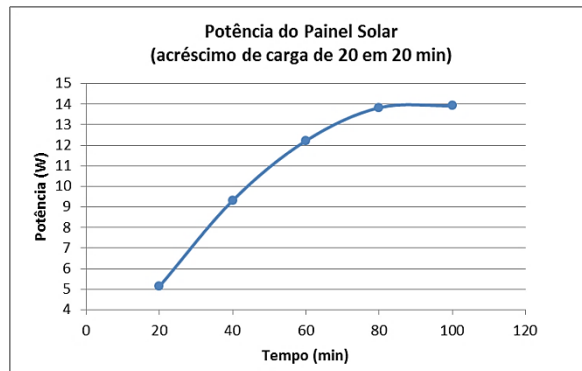
Figura 35 – Painel solar no pátio da escola direcionado para o Sol.



Fonte: acervo do autor.

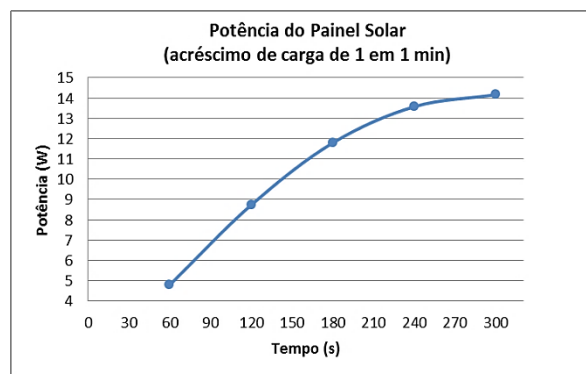
A potência encontrada para o painel, baseado na média de leituras em 1 min para cada acréscimo de carga, foi verificada no gráfico da fig. 36. A potência tende a ser no máximo entre 14 e 15 W. O mesmo acontece no gráfico da fig. 37, que tem acréscimo de carga a cada 20 min.

Figura 36 – Gráfico da medida de potência elétrica do painel em 5 min.



Fonte: autoria própria

Figura 37 – Gráfico da medida de potência elétrica do painel em 100 min.



Fonte: autoria própria.

As atividades de aplicação tanto do PPE 8 quanto do PPE 9 foram prejudicadas por alguns motivos. A escola do PPE 9 estava com a maioria dos professores paralisados por causa do não pagamento dos salários em dia pelo governo do Estado. A escola do PPE 8 praticamente estava com a maioria dos alunos aprovados e não estavam mais indo a aula. Por estas circunstâncias, as atividades foram mais simplificadas, pelo pouco tempo para desenvolver uma aplicação com mais conteúdo, e que realmente pudesse desenvolver os temas abordados no curso.

5.2.3 Considerações sobre o CPPE: achados relativos aos efeitos da intervenção sobre seus participantes

Na avaliação do Curso para Professores e Profissionais da Educação – CPPE procurou-se observar três principais eixos relativos aos objetivos de nossa intervenção pedagógica, são eles:

- i) Material instrucional;
- ii) Motivação e predisposição para aprender;
- iii) Aprendizagem.

Consideramos como **material instrucional** tudo que foi utilizado no curso para possibilitar as atividades de ensino. Envolve os *kits* Arduino, apostila e materiais auxiliares como programas de computador e slides informativos. Segundo Moreira (1999) uma das condições para que ocorra a aprendizagem significativa é que o material utilizado seja “potencialmente significativo”. Portanto, o material deve ser construído de forma que obedeça uma estrutura lógica e que esteja ao encontro da estrutura cognitiva do aprendiz. Isso está diretamente relacionado com aquilo que o aprendiz já sabe. As atividades desenvolvidas se modificavam ao longo do curso. A apostila contém exemplos baseados em conhecimentos básicos de programação e automação, visto que o perfil dos participantes era distinto e a maioria não dominava os dois campos. Por esse motivo a apostila trazia elementos que sempre tentavam vincular-se com algum conceito mais familiar a eles. Conforme esses exemplos iam sendo assimilados de forma correta, outros exemplos mais complexos iam sendo adicionados. Durante as atividades observamos que a apostila era realmente um material de auxílio que era consultado, sempre que necessário, pelos PPEs. Ao longo do desenvolvimento das atividades identificamos que os PPEs eram capazes de interpretar o que estava sendo solicitado e, até mesmo, quando tinham dúvidas eram capazes de formularem perguntas com sentido sobre o que estavam desenvolvendo. Assim, temos indícios de que o conjunto de materiais (apostila²⁵, *kit*²⁶, programas e *slides*) constitui-se em um material potencialmente significativo, na acepção de Ausubel (apud MOREIRA, 1999).

A **motivação e predisposição para aprender** dos PPEs durante o curso foi avaliado de acordo com algumas ações observadas. No início das atividades os participantes se encontravam bem motivados. Nos últimos encontros, os que continuaram assíduos, seguiram

²⁵ Cf. apêndice B.

²⁶ Cf. *kit* Arduino na p. 23 – 24.

apresentado a mesma motivação inicial. Conforme os exemplos iam sendo montados e aumentando o entendimento dos conteúdos causava ainda mais vontade de aprender. O indício de motivação mais evidente foi a participação de cada PPE, em que dos dez presentes no primeiro encontro apenas um não retornou mais. Todos não mediram esforços para montar os projetos propostos e não desistiam de uma atividade, mesmo que não estivessem tão seguros para fazer. A participação de todos por meio de perguntas, considerações e muitas discussões durante os grupos focais também demonstrou o interesse de cada PPE. A predisposição para aprender, conforme Moreira (op. cit.) é a segunda condição importante para haver a aprendizagem significativa. O aprendiz deve querer relacionar novos conhecimentos ao que já possui (conhecimentos prévios). Enquanto mais os PPEs conseguiam realizar as tarefas, que eram dadas como desafios durante as aulas, mais dispostos se encontravam a adquirir novos conhecimentos. Oposto a isso “no caso da aprendizagem mecânica, ocorre o inverso: quanto mais o aprendiz tem que memorizar conteúdos mecanicamente, mais ele ou ela se predispõe contra esses conteúdos, ou disciplinas” (MOREIRA, 2012, p. 21).

Aprendizagem é o que podemos dizer que dá sentido ao nosso processo de intervenção pedagógica realizada, pois nada poderia ser avaliado sem que fosse observado, de fato, o que os participantes aprenderam e como eles aprenderam. A partir do conceito da aprendizagem significativa (AUSUBEL, 2003) e da premissa de que “só há ensino quando há aprendizagem e esta deve ser significativa.” (MOREIRA, 2011, p. 44) a aprendizagem é fator principal. Buscamos, então, verificar nos PPEs, por meio de uma análise de nossos objetivos de aprendizagem, evidências que nos mostrassem o quanto cada participante mostrou ter atingido o objetivo da aprendizagem proposto, sempre levando em consideração as relações não arbitrárias entre os conhecimentos prévios e conceitos novos abordados a partir das atividades propostas. No quadro 5, temos a classificação de todos e com algumas observações relevantes a respeito do desempenho que cada um conseguiu atingir ou evoluir no decorrer do curso. As respostas dos questionários, grupos focais realizados e observações de ações de cada PPE foram utilizados nessa análise qualitativa. Nota-se que os conhecimentos prévios foram fator determinante na aprendizagem, pois aqueles que os possuíam, dentro de sua determinada área, tiveram mais facilidade nas atividades, seja atingindo alguns objetivos como evoluindo em outros de forma mais satisfatória. Muitas vezes alguns PPEs conseguiram respostas para problemas que surgiam durante as práticas em que a solução nem sempre se encontrava na apostila ou em outro lugar, mesmo assim eram solucionadas por formas que só

poderiam ser feitas a partir de uma aprendizagem não mecânica, que segundo Ausubel (apud Moreira, 1999, p. 52) “a melhor maneira de evitar a ‘simulação da aprendizagem significativa’ é formular questões e problemas de maneira nova e não familiar que requeira máxima transformação do conhecimento adquirido”.

Podemos observar também que os conhecimentos prévios apareceram intrínsecos dentro da estrutura cognitiva de alguns participantes. Durante as falas alguns relataram não saber programar, ou que nunca conseguiram entender programação, mesmo tendo estudado. Embora tenham tido alguma dificuldade, não era algo novo e nunca visto, mas algo que era semelhante e já visto de outra forma. A questão da eletrônica houve mais dificuldades para a maioria porque era um conteúdo totalmente novo. Procuramos tratar de uma forma que estivesse próximo, sempre que possível, ao que já tinham estudado, evitando assim uma aprendizagem mecânica e visando uma aprendizagem significativa. Exemplo dessa tentativa de atingir aprendizagem a partir de conhecimentos prévios dos PPEs são as analogias que fizemos para explicar as portas lógicas utilizando circuitos elétricos. Logo depois, utilizando as entradas e saídas das portas lógicas como um suporte para que compreendessem as entradas e saídas das portas digitais do microcontrolador do Arduino.

Quadro 5 – Avaliação dos PPEs dentro dos objetivos de aprendizagem

	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4	Objetivo 5	Objetivo 6	Objetivo 7	Observações
Descrição dos Objetivos	Compreender os conceitos básicos da programação (funções, operadores lógicos e entradas e saídas)	Compreender conceitos de automação utilizando sensores e atuadores	Identificar erros em códigos de programação	Identificar e relacionar as portas digitais com os circuitos elétricos correspondentes	Montar circuitos elétricos utilizando a <i>protoboard</i>	Construir códigos e montar circuitos para obtenção de medidas elétricas pelas portas analógicas	Compreender os conceitos de diferença de potencial, resistência e corrente elétrica através da análise de circuitos divisores de tensão utilizados nas montagens com sensores	
PPE 1	AP	AP	AP	AP	AP	AM	AL	Os conhecimentos gerais de programação tiveram um pequeno ganho. Conceitos sobre circuitos elétricos eram mais bem compreendidos e foram melhorados no decorrer do curso.
PPE 2	AP	AL	AL	AC	AC	AL	AL	Pela experiência em automação e eletricidade não encontrou dificuldades com montagem de circuitos. Teve um bom ganho na programação, embora não possuir nenhuma experiência prévia.
PPE 3	AP	AP	AM	AP	AC	AP	AP	Demonstrou alguma compreensão nas atividades, mas nas avaliações teve um desempenho muito baixo, principalmente, nas questões sobre análise de códigos de programação.
PPE 4	AP	AP	AP	AL	AL	AP	AP	Demonstrou bom desempenho nos questionários, mas teve algumas dificuldades na construção de códigos e na análise de circuitos divisores de tensão.

	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4	Objetivo 5	Objetivo 6	Objetivo 7	Observações
PPE 5	AP	AL	AP	AL	AL	AP	AP	Demostrou bastante dificuldade inicial na programação. Na montagem dos circuitos teve um bom ganho de desempenho, mas com alguma dificuldade de utilizar as portas analógicas corretamente.
PPE 7	AL	AP	AL	AL	AC	AP	AP	Teve um bom desempenho nos conceitos de programação e nos circuitos. Alguma dificuldade em encontrar os erros em códigos e na análise dos divisores de tensão.
PPE 8	AL	AL	AP	AC	AC	AL	AC	Tinha experiência com eletrônica e uso do Arduino, mas não na construção de programas. Teve ótimo desempenho na análise de circuitos. Evoluiu bastante nas práticas de montagem e programação.
PPE 9	AL	AL	AL	AP	AC	AP	AP	Teve bom desempenho em todas as atividades realizadas. Obteve um ganho no entendimento de lógica, construção de programas e montagem de circuitos, mas alguma dificuldade na análise de circuitos.
PPE 10	AC	AL	AC	AL	AL	AL	AP	Ótimo desempenho em lógica e programação (já tinha muita experiência). Bom ganho de desempenho na montagem de circuitos. Na análise de circuito evoluiu um pouco, mas não teve bom desempenho.

Fonte: autoria própria.

Legenda:

AC – Atingiu Completamente

AL – Atingiu Largamente

AP – Atingiu Parcialmente

AM – Atingiu Minimamente

Quantificando as classificações de aprendizagem (AC, AL, AP e AM) pode se ter uma ideia da pontuação de cada PPE (tabela 6) e de cada objetivo de aprendizagem (tabela 7). Para se chegar a esses escores, as classificações de aprendizagem tem um valor quantitativo equivalente, da seguinte forma:

- 1 AC → 4 pontos
- 1 AL → 3 pontos
- 1 AP → 2 pontos
- 1 AM → 1 ponto

Com base no quadro 5 que traz a classificação de cada PPE para cada objetivo de aprendizagem estipulado, temos a pontuação total dos PPEs.

Tabela 6 – Pontuação dos PPEs com relação às classificações de aprendizagem (7 menor pontuação e 28 a maior pontuação possível).

	AC (4 pontos)	AL (3 pontos)	AP (2 pontos)	AM (1 pontos)	Pontuação Total
PPE 1	0	1	5	1	14
PPE 2	2	4	1	0	22
PPE 3	1	0	5	1	15
PPE 4	0	2	5	0	16
PPE 5	0	3	4	0	17
PPE 7	1	3	3	0	19
PPE 8	3	3	1	0	23
PPE 9	1	3	3	0	19
PPE 10	2	4	1	0	22
Total	10	23	28	2	167

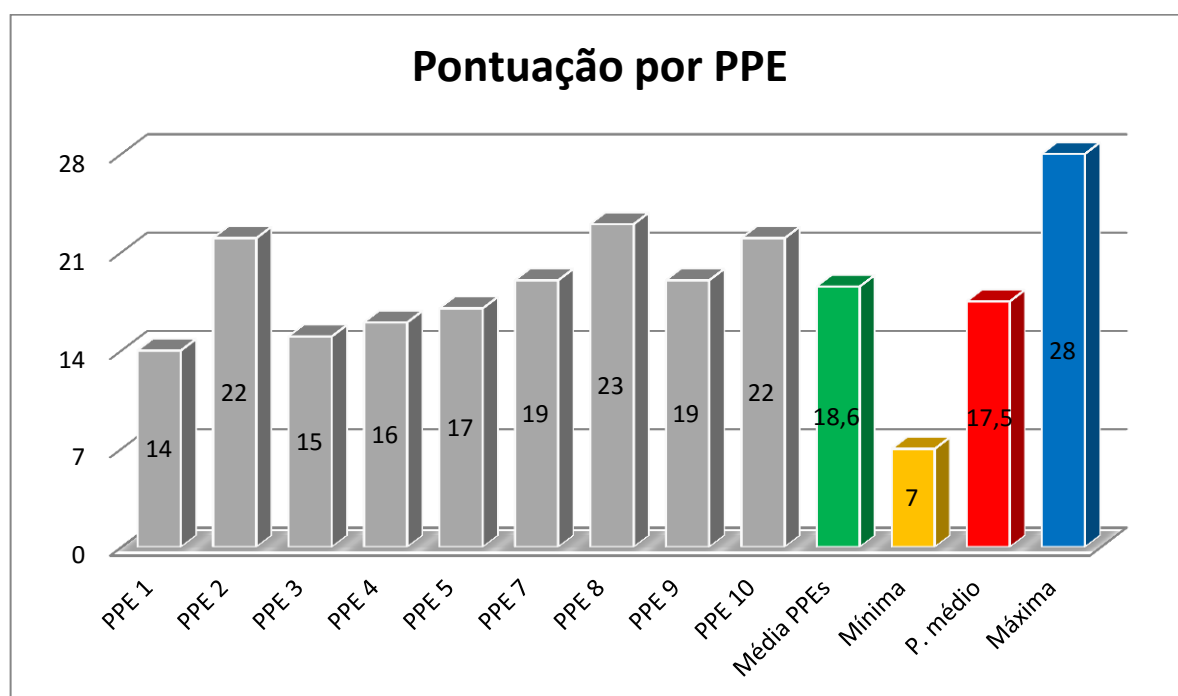
Fonte: autoria própria

Na tabela acima, observa-se a pontuação dos PPEs baseado no quanto atingiram de classificação nos parâmetros de avaliação criados. O PPE 8 foi o que mais pontuou com 23. Esse participante obteve ótimo desempenho em quase todos os objetivos alcançando 3 ACs (atingiu completamente). A sua avaliação mais baixa foi um AP (atingiu parcialmente) na identificação de erros nos códigos. Em seguida o PPE 2 com um ótimo desempenho com 22

pontos, principalmente em áreas em que não trazia uma bagagem de conhecimentos como em programação. O PPE 1 e 3 tiveram os menores desempenhos com 14 e 15 pontos respectivamente.

No gráfico da fig. 38, observa-se que a maioria dos PPEs (cinco de nove) ficou acima do ponto médio (17,5). O pior desempenho (14) ficou com o dobro da pontuação mínima possível de 7 (caso atingisse AM em todos os objetivos). Porém, nenhum se aproximou do valor máximo, mas consideramos satisfatórios esses dados, pois a pontuação média dos PPEs foi 18,6 (valor acima do ponto médio das pontuações).

Figura 38 – Gráfico das pontuações dos PPEs comparando com os valores médios, máximos e mínimos.



Fonte: autoria própria

Em relação aos objetivos, também foi construída a tabela 7, que sintetiza as pontuações de cada objetivo.

Na fig. 39 temos a representação gráfica das pontuações dos objetivos.

Tabela 7 – Pontuação dos objetivos de aprendizagem com relação às classificações de aprendizagem (9 menor pontuação e 36 a maior pontuação possível).

Objetivos	AC (4 pontos)	AL (3 pontos)	AP (2 pontos)	AM (1 pontos)	Pontuação Total
Obj. 1	1	3	5	0	23
Obj. 2	0	5	4	0	23
Obj. 3	1	3	4	1	22
Obj. 4	2	4	3	0	26
Obj. 5	5	3	1	0	31
Obj. 6	0	3	5	1	20
Obj. 7	1	2	6	0	22

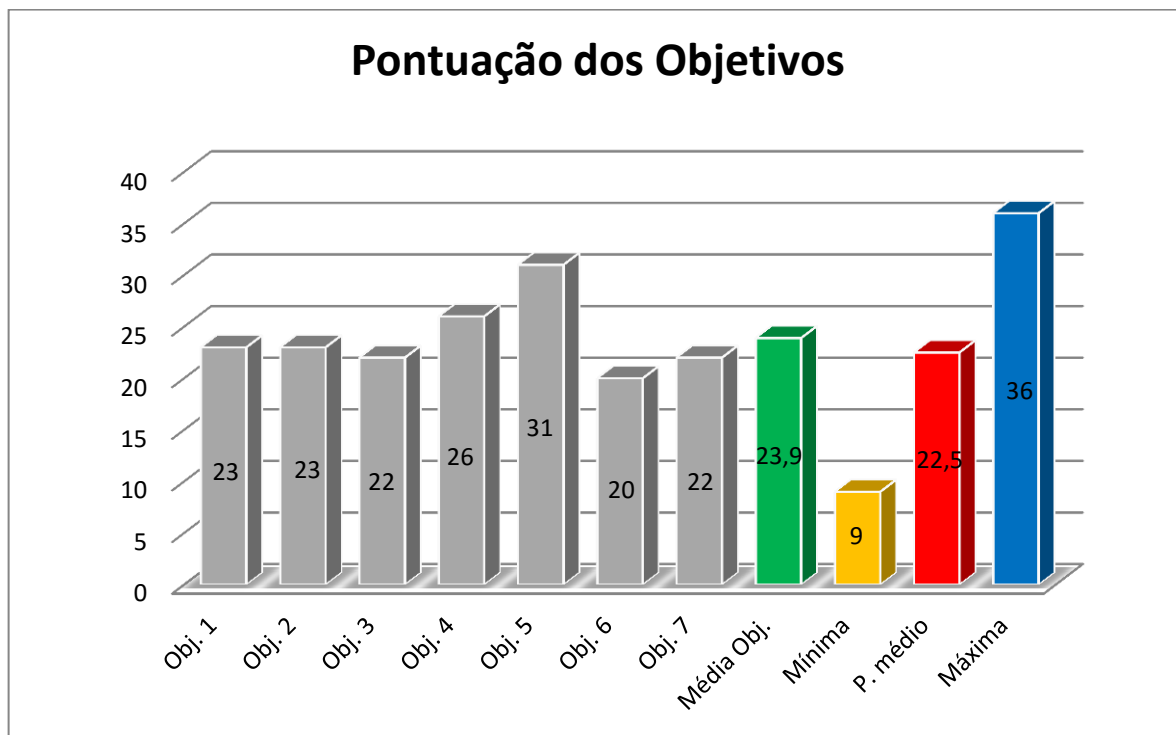
Fonte: autoria própria

Na tabela 7, observa-se a pontuação por objetivo de aprendizagem. Observa-se que o objetivo de aprendizagem que mais pontuou (31 pontos) foi o relacionado à montagem de circuitos na *proto-board*. Esse objetivo é mais prático e foi avaliado observando o grau de capacidade que os PPEs tinham para desmontar um circuito e montar outro em seguida. O que obteve menos desempenho foi o objetivo 6, relacionado à montagem de circuitos e obtenção de dados pela porta analógica do Arduino (20 pontos). Esse objetivo era considerando o mais complexo, pois envolvia mais conceitos como entendimento dos divisores de tensão, cálculos dentro da programação e conversão de valores.

Os objetivos sobre identificação de erros dos códigos (objetivo 3) e sobre conceitos de resistência, tensão e corrente elétrica e os divisores de tensão (objetivo 7) obtiveram uma pontuação baixa com apenas 22 pontos. O objetivo 1 relacionado a lógica de programação teve uma pontuação bem satisfatória, assim como o objetivo 4 sobre a relação de circuitos elétricos com portas lógicas que obteve uma pontuação ainda maior (26 pontos), sendo a segunda maior pontuação de objetivos. A média dos objetivos (23,9) ficou acima do ponto médio das pontuações possíveis (22,5).

Na fig. 39 temos a representação gráfica das pontuações dos objetivos.

Figura 39 – Gráfico da pontuação de cada objetivo.



Fonte: autoria própria

5.2.4 Considerações sobre o CPPE: achados relativos à intervenção propriamente dita

Analisando os resultados obtidos podemos fazer algumas considerações a respeito do CPPE, levando em conta as informações relevantes a respeito da intervenção e dos objetivos específicos definidos na subseção 4.1.2.1 (Módulo I: Formação) e ao que foi delimitado na seção 4.2 (Método de Avaliação da Intervenção), em que se avaliam os “achados relativos à intervenção propriamente dita” (DAMIANI et al., 2013, p. 63).

No sentido da intervenção pedagógica realizada, podemos dizer que muito do que foi constatado no decorrer do EP também se refletiu no CPPE, como: pouco tempo de estudo fora de horários das aulas; as dificuldades de recomeçar as atividades entre um encontro e outro - mas que eram superadas quando se realizava uma breve retomada dos conteúdos trabalhados, antes atividades. Também, a falta de conhecimentos prévios no CPPE foi minimizada em virtude do perfil dos participantes que já possuíam algum conhecimento sobre os vários assuntos tratados. Isso ajudou no andamento das atividades. Com relação aos dias e horários dos encontros que aconteciam aos sábados, consideramos que isto foi bem aceito pelos participantes, pois tivemos poucas faltas durante os encontros e, apenas um (PPE 6) desistiu durante o curso. Nunca tivemos pedidos para terminarmos antes do horário definido, por

exemplo. Alguns não conseguiam vir os dois turnos, por questões de trabalho ou questões pessoais, o que não atrapalhou o andamento das atividades, pois todos responderam, no mínimo, dois questionários. Os grupos focais constituíram-se em uma interessante ferramenta de discussão que ajudou a entender as dificuldades de cada PPE, suas dúvidas e perspectivas com o curso.

As atividades, exceto os questionários, foram desenvolvidas, na maioria das vezes, em duplas. Isso foi importante, porque pudemos atender cada um quando solicitado e, quando as perguntas eram comuns, aproveitávamos para responder a todos ao mesmo tempo.

Com relação à elaboração dos materiais disponibilizados, exemplos, organização do conteúdo, comunicação com os participantes e elucidação de dúvidas relativas ao conteúdo, entendemos que foram bem sucedidas. Isso pode ser verificado pelas respostas positivas de dois PPEs que preencheram uma folha de anotações (apêndice H) que serviu de avaliação em relação ao curso, por parte dos participantes, em relação ao curso. Em suas próprias palavras destacamos:

As noções básicas de lógica de programação foram essenciais no meu ponto de vista, para melhor aprendizado sobre como programar o Arduino. A disponibilidade dos materiais (apostila, *notebooks*, componentes eletrônicos e placas Arduino) foram indispensáveis para o desenvolvimento das atividades. [...] considero que o curso contribuiu muito para o meu aprendizado e cumpriu com as expectativas. Sugiro que ocorram outros cursos como esse [...] (PPE 1) .

O PPE 9 relatou que o curso foi muito proveitoso em todos os pontos, mas entendeu que, talvez, as práticas de eletrônica estejam um pouco fora da realidade que vivencia na escola onde trabalha. Esse relato é compreensível, pois quando nos propusemos a realizar essas atividades com esses profissionais, estamos tentando inserir esse tipo de conhecimento em aulas da Educação Básica, mas sabemos que isso não é algo fácil ou que vai acontecer rapidamente. De acordo com ele:

O cronograma, bem formatado, fez com que as atividades a distância fossem bem sucedidas dando possibilidades de um trabalho sequencial e de longo prazo - tratando-se de robótica educativa - e que mesmo estando em 2016, acreditamos que seja um trabalho para o futuro [...]. Concluo a observação destacando a possibilidade de ações adaptadas a realidade de cada ambiente escolar [...] (PPE 9).

Os outros participantes não entregaram essa avaliação escrita, mas manifestaram-se verbalmente, de forma positiva, sobre os mesmos itens da avaliação. Nas palavras do PPE 10 “o curso foi de grande valia, principalmente na parte de eletrônica que eu entrei aqui sabendo ‘zero’. [...] se tiverem outros cursos gostaria de ser informado [...]”.

As atividades práticas eram feitas em forma de desafios para que os PPEs resolvessem durante os encontros. Mesmo estando toda a informação no material que tinham acesso, muitas vezes partíamos de um exemplo padrão que estava na apostila e desafiávamos para que colocassem outras funções nos projetos. Mesmo dando dicas, às vezes eles não conseguiam realizar o que era solicitado e, em outros momentos, realizavam até mais do que pedíamos. A prática era sempre prioridade nos encontros (embora não “fugíssemos” das explicações teóricas), pois como é uma atividade que depende do “fazer” e repetir muitas vezes até encontrar o problema – como acontece em programação e na montagem de circuitos – sempre incentivávamos que os participantes descobrissem por si mesmos o que estavam fazendo de errado.

O tempo de curso, que foi em torno de 24 horas nos encontros presenciais, foi suficiente para desempenhar todas as tarefas que tínhamos planejado. De maneira geral o tempo foi muito bem aproveitado, pois os participantes, até mesmo nos intervalos pediam dicas e discutiam ideias de projetos. A participação a distância não foi muito efetiva por parte dos PPEs. A ideia inicial de realizar um fórum de discussão e atividades *online* não atingiu nossa expectativa. A atividade a distância que teve maior participação foi o questionário 3 (apêndice F) que foi respondido pela maioria dos PPEs somente depois de pedirmos várias vezes, inclusive condicionando a entrega deste à conclusão das atividades a distância para cumprir a carga horária do curso. Talvez, neste caso, deveríamos ter incentivado mais a utilizarem esse meio para discutir sobre assuntos do curso. Nos *e-mails* trocados não registramos nenhuma dúvida sobre exemplos ou exercícios, nem mesmo daqueles que estavam com os *kits* em casa (embora tenham relatado tentativas de utilizar).

A respeito do módulo II de aplicação nas escolas, acreditamos que este ficou prejudicado por fatores que não dependiam de nossa intervenção e sim de outros que estavam ligados à estrutura das escolas, situação atual das escolas (constantes paralisações devido aos parcelamentos de salários dos servidores) e por se tratar do período de fim de ano. Mesmo assim, entendemos que desempenhamos ações motivadoras junto aos alunos e professores. O

que foi realizado nesta etapa gerou trabalhos que, talvez, não seriam possíveis se não fossem incentivados pela proposta de curso.

Avaliando a intervenção de uma forma geral, fizemos uma seleção das principais observações a respeito das facilidades e dificuldades enfrentadas. Essas avaliações são confrontadas no quadro 6, não se tratando de uma exposição de prós e contras da aplicação do curso, mas de alguns achados relativos a intervenção pedagógica do CPPE (módulos I e II) que sintetizam os pontos fortes e os fracos identificados.

Quadro 6 – Facilidades e dificuldades encontradas no CPPE.

Facilidades	Dificuldades
<ul style="list-style-type: none"> • Alguns PPEs possuíam conhecimentos prévios sobre programação que ajudaram na compreensão dos conteúdos do curso. Dos nove PPEs, cinco eram de áreas da informática. 	<ul style="list-style-type: none"> • Os PPEs tinham mais conhecimentos prévios sobre lógica de programação do que em circuitos elétricos, a maioria teve dificuldades nesses conceitos.
<ul style="list-style-type: none"> • No trabalho em grupo, os PPEs mais capazes procuravam auxiliar os colegas e/ou duplas que apresentavam dificuldades 	<ul style="list-style-type: none"> • Às vezes o PPE mais capaz acabava realizando toda a atividade, assumindo as tarefas de forma individual.
<ul style="list-style-type: none"> • Todas as atividades de ensino propostas foram realizadas completamente por todas as duplas. 	<ul style="list-style-type: none"> • Algumas duplas demoravam mais para concluir a atividade e as duplas que terminavam primeiro ficavam dispersas e tínhamos que os desafiar com outras tarefas.
<ul style="list-style-type: none"> • As atividades em duplas facilitaram o atendimento para sanar as dúvidas que surgiam durante as aulas. Não foi necessário nenhum auxílio extra, muitas vezes os próprios PPEs ajudavam. 	<ul style="list-style-type: none"> • Muitas vezes as duplas se dividiam entre a montagem e a programação, sendo que o ideal seria todos programarem e montarem os circuitos de forma alternada. Cada PPE se incumbia na tarefa que lhe era mais familiar e não na que tinha mais dificuldades.
<ul style="list-style-type: none"> • Os participantes responderam no mínimo dois questionários que ajudaram a verificar suas aprendizagens. 	<ul style="list-style-type: none"> • As atividades propostas de forma <i>online</i> não tiveram bom retorno. Foram feitas com muito atraso e com pouco empenho em relação às presenciais.
<ul style="list-style-type: none"> • Os PPEs se empenhavam na montagem e programação, não desistindo quando algo dava errado. 	<ul style="list-style-type: none"> • Às vezes tínhamos que apontar o erro sem que eles pudessem descobrir sozinhos, para agilizar o andamento das atividades.
<ul style="list-style-type: none"> • Com o <i>kit</i>, mesmo com poucos componentes, se possibilitou a realização de várias atividades de montagem e programação. 	<ul style="list-style-type: none"> • Os sensores dos <i>kits</i> eram de funcionamento simples e estavam limitados a alguns exemplos e ao longo das atividades alguns apresentaram instabilidade durante as montagens .
<ul style="list-style-type: none"> • Os PPEs depois de executarem com êxito uma atividade tentavam modificar, variando formas de funcionamento dos programas. Isso facilitou na compreensão de alguns conceitos de programação e eletrônica. 	<ul style="list-style-type: none"> • Às vezes os PPEs tentavam realizar montagens que poderiam danificar os componentes por falta de conhecimento de alguns conceitos que ainda não tinham sido bem compreendidos.
<ul style="list-style-type: none"> • Os PPEs tentavam não copiar o código tentando fazer passo a passo desde o início. Recorriam à apostila quando precisavam, utilizando os exemplos dela. 	<ul style="list-style-type: none"> • No início da programação eles tinham muitas dificuldades, principalmente em começar as primeiras linhas dos códigos.
<ul style="list-style-type: none"> • O tempo dos encontros foi bem aceito pelos PPEs. Poucas vezes pediram para que terminássemos antes do horário previsto. 	<ul style="list-style-type: none"> • Embora entendermos que o tempo de curso poderia ser maior, pela quantidade de conteúdos, não seria viável aumentar a carga horária.
<ul style="list-style-type: none"> • Na aplicação das escolas, os alunos selecionados se empenharam juntos com os professores para realizarem as tarefas propostas. 	<ul style="list-style-type: none"> • Alguns alunos do PPE 9 desistiram no decorrer das atividades.
<ul style="list-style-type: none"> • Alunos que já estavam aprovados participaram da aplicação nas escolas, demonstrando motivação pelos projetos que foram realizados. 	<ul style="list-style-type: none"> • Nas escolas estaduais estavam ocorrendo frequentemente paralisações da categoria dos professores do magistério estadual e no fim do ano poucos alunos ainda estavam indo a aula.
<ul style="list-style-type: none"> • Os projetos feitos nas escolas tinham vários conceitos de programação e utilização de sensores e atuadores que foram tratados no curso, o que possibilitou colocar em prática o que foi aprendido pelos PPEs, 	<ul style="list-style-type: none"> • Foram feitos em muito pouco tempo. Não podemos discutir a fundo sobre os projetos com os alunos das escolas. Ficaram muitas dúvidas sobre programação e eletrônica, mesmo que os projetos tenham funcionado.

Baseado nas facilidades e dificuldades elencadas no quadro 6, e sobre o que foi discutido até então sobre o CPPE, podemos fazer algumas recomendações gerais para aqueles que pretendem usar a Plataforma Arduino, seja em cursos de formação de professores, atividades práticas em escolas da Educação Básica, ou em qualquer aplicação que envolva a introdução da lógica de programação e eletrônica. Acreditamos que essas recomendações também serão relevantes para estudos futuros, principalmente aos que tenham como cerne de pesquisa os meios de inserir conceitos introdutórios de programação e eletrônica em currículos da Educação Básica.

São elas:

- i. Levantar conhecimentos prévios utilizando pré-testes ou questionários investigativos;
- ii. Realizar atividades preliminares sobre lógica e eletrônica quando os conhecimentos prévios não forem muito consistentes;
- iii. Garantir, quando possível, que em cada grupo tenha indivíduos com conhecimentos prévios complementares entre lógica de programação e eletrônica;
- iv. Procurar que todos integrantes do grupo participem ativamente de todas as atividades, de modo a partilhar diferentes conhecimentos e experiências. Por exemplo, em duplas, os participantes podem se dividir entre as atividades de programação e montagens de circuitos, trocando papéis, realizando tarefas que são mais familiares e aprendendo sobre aquelas que não possuem experiências;
- v. Propor instrumentos avaliativos ao longo da realização das atividades, para verificar o andamento das aprendizagens dos participantes e ao mesmo tempo propiciar um *feedback* sobre as dificuldades a serem superadas;
- vi. Quando um grupo ou indivíduo realizar uma atividade com êxito, as tarefas podem ser modificadas em alguns pontos com a intenção de intensificar o pensamento lógico, fazendo pensar de formas diferentes sobre mesmos conceitos;
- vii. O ideal é que os códigos sejam construídos desde o início da cada atividade de programação, melhorando através da prática as dificuldades iniciais enfrentadas nas retomadas entre uma tarefa e outra;
- viii. Os circuitos montados na *protoboard* devem ser compreendidos no mesmo nível dos

códigos de programação para os erros de projetos sejam corrigidos. Como o correto funcionamento depende dessas duas etapas, não é suficiente ter um entendimento maior em apenas uma delas. Identificar as deficiências em cada etapa é fator essencial;

- ix. As atividades práticas devem predominar na maior parte, para que os participantes não se desmotivem pelo excesso de informações novas;
- x. É importante que as atividades propostas sejam terminadas por todos para os indivíduos com mais dificuldade não se desmotivarem;
- xi. Levando em conta o perfil dos participantes e a disponibilidade de materiais, é importante propiciar a execução de projetos a partir de temas livres, visando estimular a criatividade na solução de problemas enfrentados pelos próprios participantes.

6 REFLEXÕES FINAIS

No trabalho apresentado discutimos sobre uma intervenção pedagógica que abordou os conhecimentos introdutórios de lógica de programação, eletrônica e automação, visando estabelecer melhores condições para o uso da Plataforma Arduino no Ensino de Ciências, principalmente na Educação Básica. No primeiro momento da pesquisa foi planejada a realização do EP, com o objetivo de levantar questões preliminares sobre o material instrucional. Ao concluir a fase do EP observamos que os alunos do Ensino Médio, embora motivados a aprender, demonstram poucos conhecimentos prévios sobre lógica e conceitos físicos envolvidos no estudo de eletricidade. Outro fator que foi observado é com relação aos professores titulares desses alunos, que da mesma forma, não possuem conhecimentos necessários para iniciar atividades diferenciadas, no ambiente escolar, como: projetos e oficinas de programação e eletrônica. Neste sentido, resolvemos criar um curso de formação (CPPE) de professores ou qualquer profissional que estivesse ligado a área da educação, com base nas dificuldades encontradas no EP. Continuávamos ainda com o mesmo intuito de inserir conhecimentos de programação e eletrônica utilizando a Plataforma Arduino no Ensino Básico, mas de outra forma: através daqueles que já estavam inseridos no contexto escolar e que poderiam futuramente seguir desenvolvendo as atividades.

Embora o objetivo fosse atingir um público-alvo de docentes de Física ou Ciências, obtivemos inscrições de profissionais de várias áreas como Informática, Mecânica, Automação, Biologia, Matemática e Física e que exerciam atividades educacionais em vários níveis como Fundamental, Médio, Técnico e Superior. O que também pode ser observado é que a grande maioria já trabalhava com alguns conceitos de lógica de programação nas suas escolas, portanto já estavam inseridos em um contexto mais próximo do propósito do curso. Quase todos nunca tinham trabalhado com montagem de circuitos e isso era uma novidade que se distanciava mais da realidade das escolas.

Mesmo havendo vários estudos que buscam utilizar a Plataforma Arduino no Ensino de Ciências, nossa proposta se mostrou diferenciada no sentido de dar ênfase na lógica de programação e na eletrônica. Realizamos várias atividades que tiveram a intenção de explorar formas para facilitar a aprendizagem nesses dois temas. Como exemplos dessa proposta, citamos a atividade utilizando planilhas eletrônicas; a utilização de circuitos elétricos como analogias a portas lógicas; análise de circuitos digitais com portas AND e OR; verificação de erros em códigos; utilização da *protoboard*; análise e montagem de circuitos divisores de

tensão; utilização correta de sensores e atuadores, entre outras. Esses temas abordados tinham o intuito de incentivar a não reprodução de trabalhos que, muitas vezes, fazem com que os alunos utilizem este recurso apenas como um equipamento de medição (como um voltímetro, por exemplo) não adquirindo habilidades reais que são o objetivo original da Plataforma Arduino. Baseado no fato de que consideramos que o maior empecilho enfrentado pelos professores do Ensino Básico é a falta de experiência no ensino de programação e na montagem de circuitos, ao longo do curso foram tratadas e demonstradas a importância de tais experiências para o desenvolvimento de atividades que não visem somente a reprodução de experimentos e sim a compreensão e adaptação para projetos que levem em conta os anseios dos professores e alunos. Não queremos aqui discutir a validade de outros trabalhos relacionados a este, mas refletir sobre quais os conceitos importantes que devem ser explorados de forma mais pragmática. Embora elaborar projetos para ensinarmos conceitos físicos ou projetos de robótica que incentivem alunos nas aulas sejam interessantes, precisamos atentar para o conhecimento mais profundo de como as coisas funcionam. Entender conceitos de programação e eletrônica muitas vezes passam despercebidos em projetos e devemos, sempre que possível, evitar que os alunos apenas copiem códigos e reproduzam a montagem de circuitos sem o devido entendimento.

As ações realizadas ao longo do CPPE foram avaliadas conforme os achados relativos aos participantes e a intervenção pedagógica propriamente dita como preconiza Damiani (2013). Em relação aos participantes encontramos indícios de que o foco inicial no ensino de programação e em circuitos elétricos propiciou conhecimentos prévios para as atividades posteriores, contribuiu para os participantes se sentirem aptos ao desenvolverem as atividades e, assim, manifestaram predisposição para realizar os desafios propostos sendo capazes de relacionar, de forma não arbitrária, os conteúdos contidos nos materiais instrucionais com seus conhecimentos. Diante disso, consideramos que foram atingidas as duas condições para ocorrência da Aprendizagem Significativa, sendo elas a predisposição para aprender e a utilização de material potencialmente significativo. Sobre os achados relativos à intervenção pedagógica realizada foram estabelecidas onze recomendações, descritas na seção anterior, que poderão embasar estudos futuros, principalmente aos que tenham como cerne de pesquisa os meios de inserir conceitos introdutórios de programação e eletrônica em currículos da Educação Básica. A falta de estudo fora do horário de aula prejudicou a aprendizagem nas etapas do EP e CPPE. Como os encontros nas duas etapas eram realizados em um dia apenas na semana, este intervalo não era utilizado por eles para rever os conteúdos. Tentamos fazer

com que levassem os *kits* de Arduino para casa, mas não tivemos muito resultado porque os participantes pareciam sentir quase como uma “obrigação” de apresentar o que foi feito com o mesmo. A ideia de emprestar o material era apenas de fazer com que eles se familiarizassem com o *kit* podendo utilizá-lo quando quisessem. Se tivessem praticado algumas horas dentro da semana anterior aos encontros, algumas dúvidas que se repetiam sempre no início das tarefas poderiam ser superadas com mais facilidade.

A respeito da aplicação dos conhecimentos no ambiente escolar dos PPEs, atingimos esse objetivo com dois participantes, o que consideramos um bom resultado, considerando vários fatores que não dependiam de nossa vontade. Entre eles estão: a falta de tempo para realizar as tarefas; a conjuntura atual do ensino público, principalmente, dos professores estaduais que estavam com o seus salários atrasados; e outros fatores relacionados à falta de motivação dos alunos. O resultado da aplicação com os alunos do EP gerou outros trabalhos que foram desempenhados por iniciativa própria dos alunos que nos procuram depois de alguns meses para ajudar a realizar projetos que seriam apresentados em feiras e mostras científicas. Isso nos mostra que o trabalho não foi pontual, pois gerou resultados posteriores e esperamos que gerem ainda mais.

As experiências profissionais do autor deste estudo, que foram adquiridas durante o seu envolvimento com professores e alunos de escolas, foi de grande valia. Não só pelo acréscimo de conhecimento que a pesquisa proporcionou, mas também pela a ampliação de uma importante interação Universidade-Escola que se manteve presente na pesquisa. Essa interação deveria se manter, pois entendemos que a pesquisa em educação nunca pode se distanciar da realidade do Ensino Básico. Servidores de instituições de Ensino Superior não podem apenas preocupa-se com os futuros profissionais que formarão, mas também com aqueles que serão os seus futuros alunos. Por isso, os cursos, oficinas, *workshops*, feiras de tecnologias e etc, devem sempre ter espaço dentro das Universidades para o público de professores e servidores de escolas que desempenham trabalhos inovadores e incentivadores com seus alunos. A falta de equipamentos e estruturas nas escolas são fatores importantes que devem ser tratados em outras discussões, mas não pode impedir a tentativa de produzir conhecimento e reflexão. Os *kits* que foram utilizados são matérias que não possuem um custo muito elevado²⁷ e, muitas vezes, há possibilidade de verbas para tal por meio de projetos dos

²⁷ Atualmente (dez de 2017) encontramos *kits* Arduino para iniciantes por preços que variam entre R\$ 120,00 e R\$ 180,00. Esses *kits* possuem praticamente todos os componentes dos *kits* utilizados neste trabalho. Um computador, por exemplo, pode custar em média quase dez vezes esse valor.

governos estaduais e municipais ou até mesmo por parcerias com a iniciativa privada. Para isso, os professores tem que estar à frente da elaboração desses projetos e precisam ter conhecimento e argumento necessário para convencer da necessidade de adquirir esses materiais diferenciados para serem utilizados como ferramenta no ensino.

No que diz respeito a estudos futuros esperamos que sejam considerados os fatores levantados neste trabalho, seja para cursos de formação de professores ou para alunos em disciplinas de tecnologias educacionais ou similares. Para fornecer mais subsídio aos que necessitam de informações, este trabalho ainda fornece uma produção educacional contendo exemplos de atividades, exercícios e informações sobre a Plataforma Arduino, conceitos básicos de circuito divisores de tensão e utilização de sensores e atuadores e serve de referência ao que foi discutido.

REFERÊNCIAS

ARAÚJO, D. C et al. **O Ensino de Computação na Educação Básica apoiado por Problemas: Práticas de Licenciandos em Computação**. In: XXIII do WEI-Workshop sobre Educação em Computação. 2015.

ARAÚJO, I. S. ; VEIT, E. A. **Uma revisão da literatura sobre estudos relativos a tecnologias computacionais no ensino de Física**. Investigações em Ensino de Ciências (UFRGS), São Paulo, v. 4, n.3, p. 5-18, 2004.

AUSUBEL, D. P. **Aquisição e Retenção de Conhecimentos: uma perspectiva cognitiva**. Lisboa: Plátano, 2003. 226 p.

CAVALCANTE, M. A.; TAVOLARO, C. R. C.; MOLISANI, E. **Física com Arduino para iniciantes**. Revista Brasileira de Ensino de Física, v. 33, n. 4, p. 4503-1 a 4503-9, 2011.

DAMIANI, M. F. et al. **Discutindo pesquisas do tipo intervenção pedagógica**. Cadernos de Educação, n. 45, p. 57- 67, 2013.

DWORAKOWSKI, L. A. Q. **Construção e Interpretação de Gráficos da Cinemática: uma proposta para o Ensino Médio Politécnico**. 2015. 114 f. Dissertação de Mestrado. Curso de Pós-graduação em Ensino de Ciências, Universidade Federal do Pampa, Bagé, 2015.

FERNANDES, M. B. **Eletricidade: uma sequência didática para o ensino médio integrado**. 2015. 164 f. Dissertação (Mestrado) - Curso de Pós-graduação em Ensino de Ciências, Unipampa, Bagé, 2015.

FILHO, G. F. **Experimentos de baixo custo para o ensino de Física em Nível Médio usando a placa Arduino-UNO**. 207 f. Dissertação de mestrado. Curso de Pós-graduação em Ensino de Física, Ufrgs, Porto Alegre, 2015.

GONDIM, S. M. G. **Grupos focais como técnica de investigação qualitativa: desafios metodológicos**. Paidéia, Ribeirão Preto , v. 12, n. 24, p. 149-161, 2002 .

KENSKI, V. M. **Educação e tecnologias: o novo ritmo da informação**. Campinas: Editora Papirus, 2012. 141p.

LUCIANO, A. P. G. **A utilização da robótica educacional com a Plataforma Arduino: uma contribuição para o Ensino de Física**. 151 f. Dissertação de Mestrado. Curso de Pós-Graduação em Educação para a Ciência e a Matemática. UEM, Maringá, 2014.

MOREIRA, M. A. *¿ Al final, qué es aprendizaje significativo?* Currículum: revista de teoría, investigación y práctica educativa. La Laguna, Espanha, n. 25, p. 29-56, 2012.

_____. **Teorias de Aprendizagem**. 2. ed. São Paulo: EPU, 2011. 242 p.

_____. **Unidades de Enseñaza Potencialmente Significativas – UEPS**. Aprendizagem Significativa em Revista / Meaningful Learning Review, Porto Alegre, v.1, n. 2, p. 43 – 63, 2011.

_____. **Aprendizagem Significativa**. Brasília: Ed. Universitária de Brasília, 1999. 129 p.

MOREIRA, M. A.; OSTERMANN, F. **Teorias construtivistas**. (Texto de Apoio ao Professor de Física; nº10), Porto Alegre, UFRGS, 1999.

MORGAN, D. L. **Focus group as qualitative research. Qualitative Research Methods Series**. London: Sage Publications, 2 ed., 1997.

MORTARI, C. A. **Introdução à lógica**. São Paulo: Ed. UNESP, 2001. 397 p.

MUGNAINI, R.; STREHL, L. **Recuperação e impacto da produção científica na era Google: uma análise comparativa entre o Google Acadêmico e a Web of Science**. Encontros Bibli: Revista eletrônica de biblioteconomia e ciência da informação, n. Esp, 2008. Disponível em: <<http://www.redalyc.org/articulo.oa?id=14709808>> Acesso em: 27 nov. 2017.

NETO, B. D. **Aprendizagem de Conceitos Físicos Relacionados com Circuitos Elétricos em Regime de Corrente Alternada com Uso da Placa Arduino**. 2013. 171 f. Dissertação (Mestrado) - Curso de Pós-graduação em Ensino de Física, Ufrgs, Porto Alegre, 2013.

RODRIGUES, R. F. **Arduino como uma ferramenta mediadora no ensino de Física**, 90 f. Dissertação de mestrado. Curso de Pós-graduação em Ensino de Física, Ufrgs, Porto Alegre, 2014.

ROSADO, L.; CARMONA, A. G. **Razones didácticas y epistemológicas de la introducción de nociones de física de semiconductores en Educación Secundaria**. Revista Electrónica de Enseñanza de las Ciencias, v. 4, n. 3, 2005.

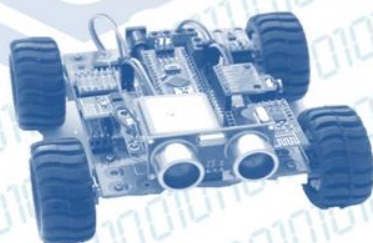
SILVEIRA, S.; GIRARDI, M. **Desenvolvimento de um kit experimental com Arduino para o ensino de Física Moderna no Ensino Médio**. Revista Brasileira de Ensino de Física, São Paulo, v. 39, n. 4, 2017 .

SOUZA, A. R. et al. **A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC**. Revista Brasileira de Ensino de Física, São Paulo, v. 33, n. 1, p. 1702, 2011.

VIANNA, H. M. **Pesquisa em educação: a observação**. Brasília: ed. Liber Livros, 2007.

APÊNDICE A – Cartaz de divulgação do curso

CURSO DE INICIAÇÃO AO ARDUINO PARA PROFESSORES



• ➔ Conceitos básicos de lógica de programação

• ➔ Conhecendo o Arduino

• ➔ Uso do Arduino em sala de aula

Inscrições de 5 a 12 de setembro em:

www.sites.google.com/site/arduinounipampa

Local: UNIPAMPA – Campus Bagé

Encontros presenciais: 17/9, 24/9, 08/10 e 22/10

Público-alvo: Professores de Física da Educação Básica

Carga horária: 60 horas

Total de Vagas: 10

Mais informações no site



Programa de Pós-Graduação em
Ensino de Ciências (PPGEC)



APÊNDICE B – Apostila utilizada no curso

Curso de Arduino para Professores

Noções Básicas de Arduino e Programação



Januário Dias Ribeiro



2016



Sumário

Capítulo I: Noções Básicas de Lógica de Programação	119
1.1 Algoritmos e Programas de Computador.....	119
1.2 Mas o Que o Computador Realmente Entende?	119
1.3 Bits e Bytes	119
1.4 Representação Binária e Decimal.....	119
1.5 Operadores Lógicos.....	120
1.6 Portas Lógicas e Tabelas Verdade	120
1.6.1 Tipos de portas lógicas e suas simbologias.....	121
1.6.2 Tabelas verdade dos operadores lógicos.....	122
Capítulo II: Conhecendo o Arduino	125
2.1 Arduino Uno	126
2.2 Características do Arduino Uno	127
Capítulo III: Instalando e Configurando o Arduino	128
Capítulo IV: Programando o Arduino	130
4.1 Variáveis	130
4.2 Tipos de Dados	131
4.3 void setup() e void loop()	131
4.4 Funções.....	131
4.4.1 Entradas e saídas digitais	131
4.4.2 Entradas e saídas analógicas.....	131
4.5 Funções de Tempo	132
4.6 Estruturas de Controle e Comparadores.....	132

Capítulo V: Praticando a Programação do Arduino	134
5.1 Comentários	135
5.2 Monitor Serial e o Comando Serial.print()	136
Capítulo VI: Montando Projetos com o Arduino	139
6.1 Sensores	139
6.2 Atuadores	140
6.3 Divisor de Tensão	140
6.4 LDR e Resistor	143
6.5 Sensor de Temperatura LM35.....	145
6.5.1 Aplicação do LM35.....	145
6.5.2 Utilizando o LM35 com o Arduino	146
Referências	147

Capítulo I: Noções Básicas de Lógica de Programação

1.1 Algoritmos e Programas de Computador

Um algoritmo nada mais é do que uma sequência de passos que devem ser seguidos para realizar uma tarefa. No nosso dia a dia utilizamos vários algoritmos: em casa, na escola, no trabalho, etc. Um programa de computador é um algoritmo escrito em uma linguagem que o computador entende. Para desenvolver um programa, assim como na execução de nossas tarefas, precisamos de uma sucessão de passos (algoritmos) e o sucesso final depende de quão bom foi a nossa lógica de passos. Claro que poderá haver vários modos de resolver um problema ou tarefa, mas alguns podem ser mais ou menos eficientes.

1.2 Mas o Que o Computador Realmente Entende?

O computador não entende a linguagem que usamos para nos comunicar, mas sim a linguagem de máquina que é uma sucessão de códigos binários que são números 0 e 1. Na verdade essa é uma representação de um estado, ou seja, o computador entende sinais elétricos que percorrem seus componentes. Podemos dizer que o zero representa o estado desligado (ou baixo) e o 1 representa o estado ligado (ou alto). Então, o que ele entende é a presença ou não de uma tensão elétrica que é medida e a partir daí vários processos acontecem para o seu funcionamento.

1.3 Bits e Bytes

Os códigos binários podem representar os números ou letras que usamos. Um bit é formado por um par 0 e 1 e um byte é formado por 8 bits.

1.4 Representação Binária e Decimal

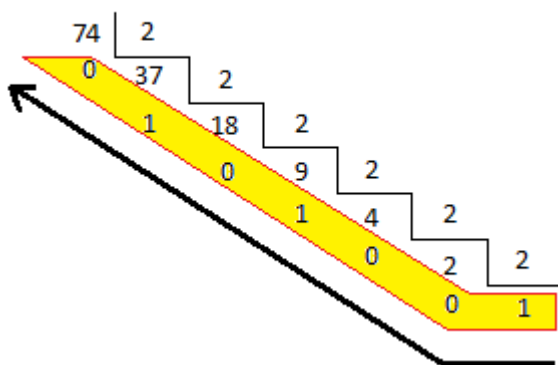
O sistema comumente usado por nós é o decimal que pode ser representado pelo sistema binário.

Ex.: o número 74 pode ser representado na forma binária

$$74 = 1001010$$

A conversão é feita dividindo o número por dois e encontrando os restos da divisão até chegar ao correspondente binário, como na figura 1.

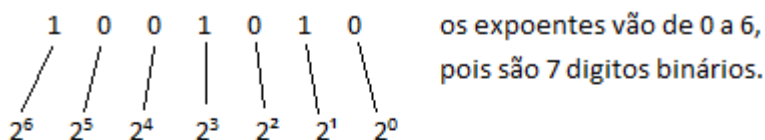
Fonte: autoria própria



Conta-se de baixo para cima e se tem o número binário correspondente ao decimal 74, que é o 1001010.

Fig. 1 - Conversão de decimal para binário.

Para converter novamente para decimal utilizamos uma sequência de soma de potências de base 2 com o expoente começando de zero até o número de dígitos binários menos um. Escrevemos as potências da direita para a esquerda.



Agora somamos todos que correspondem aos dígitos 1, os que correspondem aos dígitos 0 não são considerados.

$$2^5 + 2^3 + 2^1 =$$

$$64 + 8 + 2 = 74$$

1.5 Operadores Lógicos

Segundo a Álgebra Booleana introduzida por George Boole em 1854, existem três operadores principais: **AND** (E em português) que é definida como uma multiplicação lógica; **OR** (OU em português) que é definida como uma adição lógica e **NOT** (NÃO em português) definido como uma complementação (inversão ou negação).

1.6 Portas Lógicas e Tabelas Verdade

Esses operadores também são utilizados na eletrônica digital representados por portas lógicas. Nestas portas só há dois tipos de valores que podem assumir e podem ser denotados de várias formas como:

ligado - desligado (L - D)

alto - baixo ou High - Low (H - L)

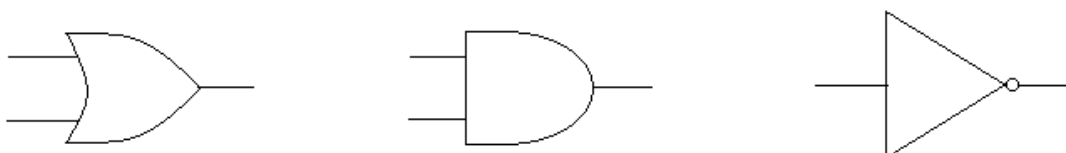
1 - 0

Verdadeiro - Falso (V - F)

Para facilitar nossa compreensão, utilizamos uma representação gráfica das portas lógicas e também as tabelas verdade.

1.6.1 Tipos de portas lógicas e suas simbologias

Porta OR (OU, adição) Porta AND (E, multiplicação) Porta NOT (NÃO, inversora)



- A porta **OR (adição)** possui dois ou mais valores de entrada e resulta em um valor verdadeiro (1) se pelo menos um dos valores de entrada seja verdadeiro, caso contrário será falso (0);
- A porta **AND (multiplicação)** possui dois ou mais valores de entrada e resulta em um valor verdadeiro (1) somente se todos os valores de entrada forem verdadeiros, caso contrário será falso (0);
- A porta **NOT (inversora)** possui um só valor de entrada e tem a função de inverter o valor que entra. Se for verdadeiro (1) na entrada a sua saída passa a ser falsa (0). Também conhecida como **porta inversora**.

Podemos fazer uma representação dessas portas digitais usando circuitos elétricos em série e em paralelo, onde as entradas são os interruptores e a saída é uma lâmpada ligada (1) ou desligada (0):

Fonte: autoria própria

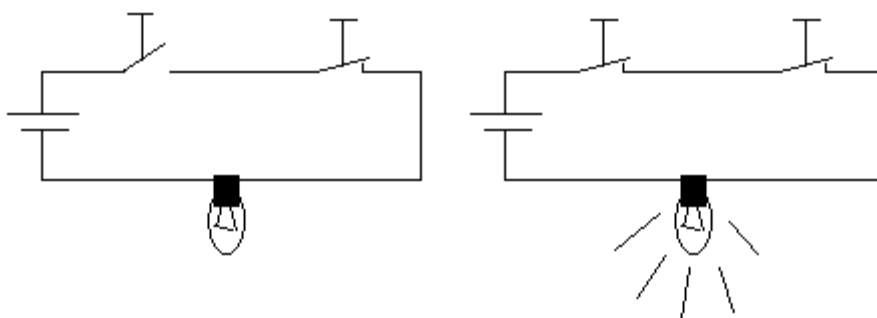


Fig. 2 - Porta **AND** representadas por circuitos elétricos em série.

Fonte: autoria própria

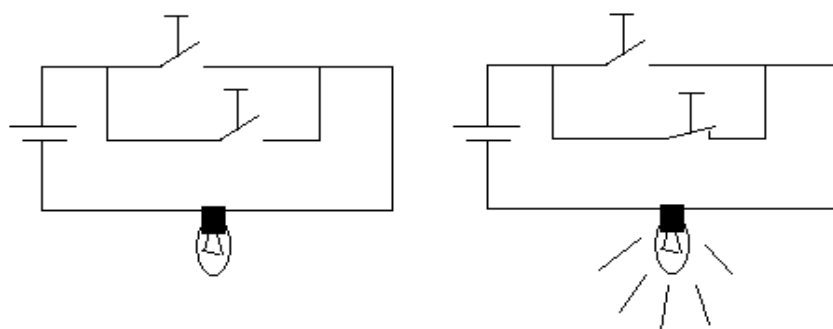


Fig. 3 - Porta **OR** representadas por circuitos elétricos em paralelo.

1.6.2 Tabelas verdade dos operadores lógicos

Os operadores lógicos podem ser colocados em **tabelas verdade** com todas as possíveis entradas e suas respectivas saída de acordo com a operação a ser realizada.

Tabela verdade do OR

Entradas		Saída
A	B	A+B
1	0	1
0	1	1
1	1	1
0	0	0



Tabela verdade do AND

Entradas		Saída
A	B	A.B
1	0	0
0	1	0
1	1	1
0	0	0

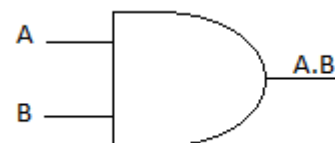
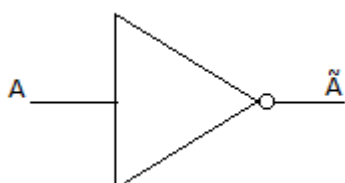


Tabela verdade do NOT

Entrada	Saída
A	\bar{A}
1	0
0	1



Existem outras portas lógicas chamadas NAND (NÃO-E) e NOR (NÃO-OU) que são formadas por portas AND e OR com sua saídas invertidas por uma porta NOT. Representadas como na figura 4a. Existem ainda outras portas chamadas XOR (OU exclusivo) e XNOR. A figura 4b é um CI (circuito integrado) real com várias portas lógicas e seu circuito interno.

Fonte: Wikipédia

Fonte: autoria própria

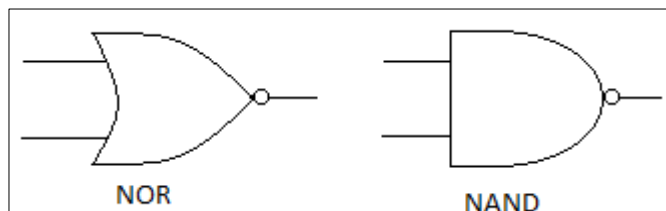


Fig. 4a - Portas NOR e NAND.

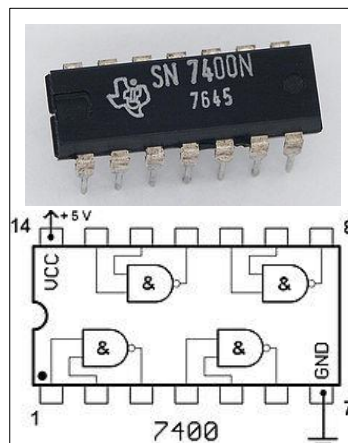


Fig. 4b - Circuito integrado 7400N contendo quatro portas NAND.

Podemos fazer circuitos unindo várias portas lógicas diferentes para termos uma saída.

Exemplo:

Fonte: autoria própria

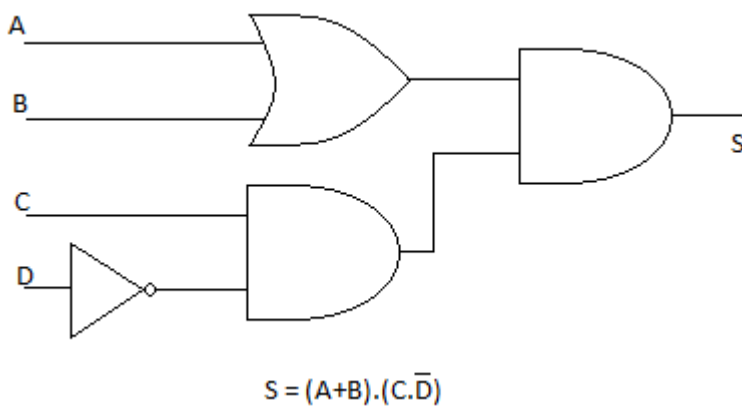


Fig. 5 - Circuito com portas lógicas e sua equação correspondente.

Atividades:

- 1) Faça a tabela verdade das portas NAND e NOR.
- 2) Faça a tabela verdade do circuito da figura 5.
- 3) Faça uma planilha eletrônica para testar diferentes entradas e obter saídas do exemplo da figura 5.
- 4) Crie portas NAND e NOR em uma planilha eletrônica.

Capítulo II: Conhecendo o Arduino

O Arduino é uma plataforma de prototipagem eletrônica *open-source* que se baseia em *hardware* e *software* flexíveis e fáceis de usar. É destinado a artistas, designers, hobbistas e qualquer pessoa interessada em criar objetos ou ambientes interativos. Pode receber entradas de uma variedade de sensores e pode atuar no ambiente ao seu redor por controladores de iluminação, motores e uma infinidade de atuadores.

A plataforma Arduino é composta por *hardware* e *software*. O *hardware* é formado por uma placa onde estão o microcontrolador – que é o “cérebro” do Arduino – e outros componentes necessários para o seu funcionamento como: resistores, cristal oscilador, capacitores, CIs, etc. Possui uma comunicação serial via USB para conexão no computador para que sejam carregadas as instruções que deverão ser executadas pelo microcontrolador. Na parte de *software* possui uma IDE para que sejam digitados e carregados os programas.

O Arduino é diferente das outras plataformas presentes no mercado devido a vários fatores, entre eles:

1. É um ambiente multiplataforma, podendo ser executado em Windows, Macintosh e Linux;
2. Pode ser programado utilizando um cabo de comunicação USB onde geralmente não é necessária uma fonte de alimentação;
3. Possui *hardware* e *software open-source*, facilitando a montagem do seu próprio hardware sem precisar pagar nada aos criadores originais;
4. Hardware de baixo custo;
5. Grande comunidade ativa de usuários;
6. Ambiente educacional, ideal para iniciantes que desejam resultados rápidos.

Devido a todos os benefícios e facilidades o Arduino se tornou muito presente no meio educacional, tanto em nível básico como superior. Hoje muitas escolas no mundo todo utilizam a plataforma para projetos educacionais de robótica e automação.

Há vários modelos de placas Arduino, na figura 6 podemos ver alguns. Durante nosso curso utilizaremos sempre o Arduino Uno como referência para os exemplos de montagens e códigos. Esse modelo é um dos mais populares entre as pessoas que estão começando em seus primeiros projetos.

Fonte: <<https://arduinoaprendizes.wordpress.com/2015/04/22/historiaarduino/>>

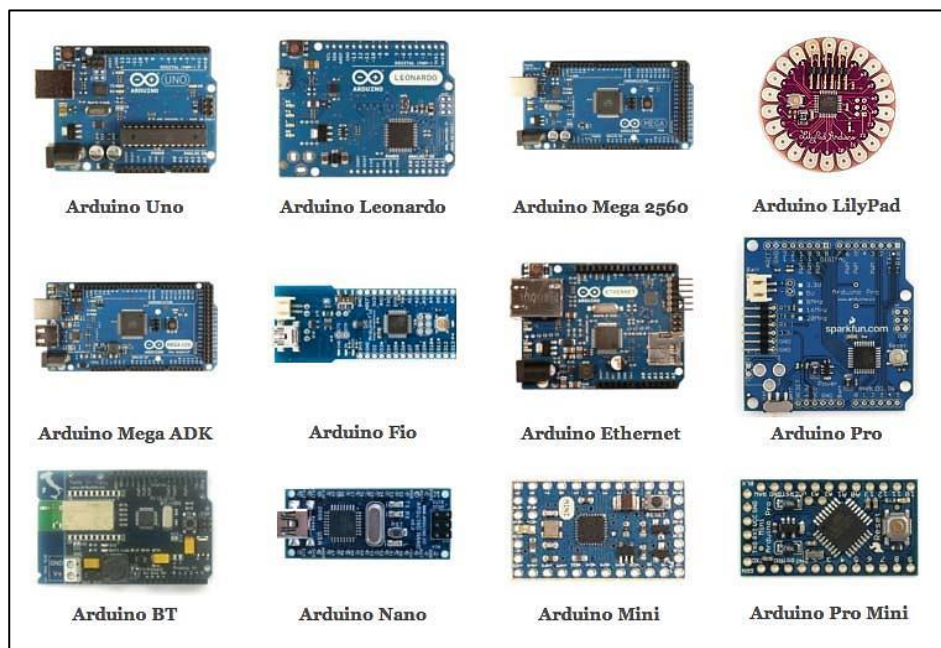


Fig. 6 – Modelos de placas Arduino.

2.1 Arduino Uno

Na figura 7, vemos as partes mais importantes que devemos conhecer sobre o Arduino Uno.

Fonte: adaptado de <<http://blog.hackerearth.com/2016/10/a-tour-of-the-arduino-uno-board.html>>

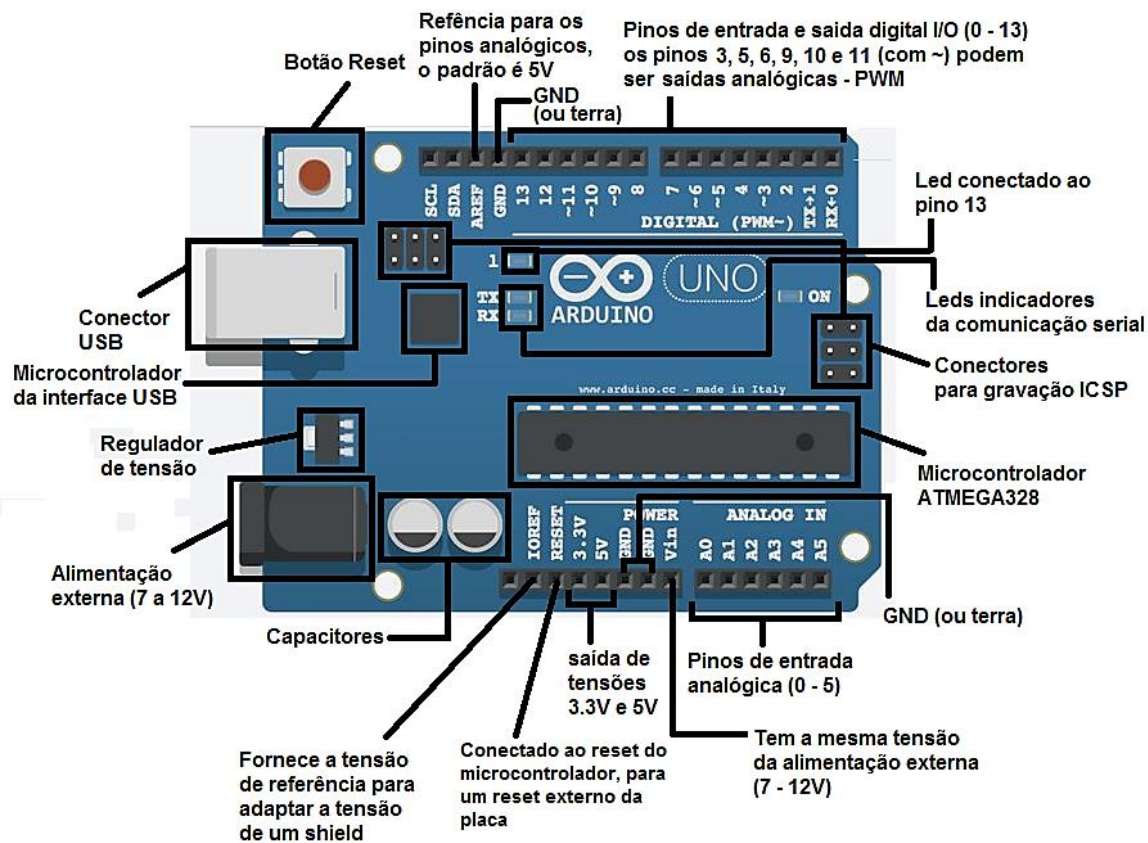


Fig. 7 – Partes principais da placa Arduino Uno.

2.2 Características do Arduino Uno

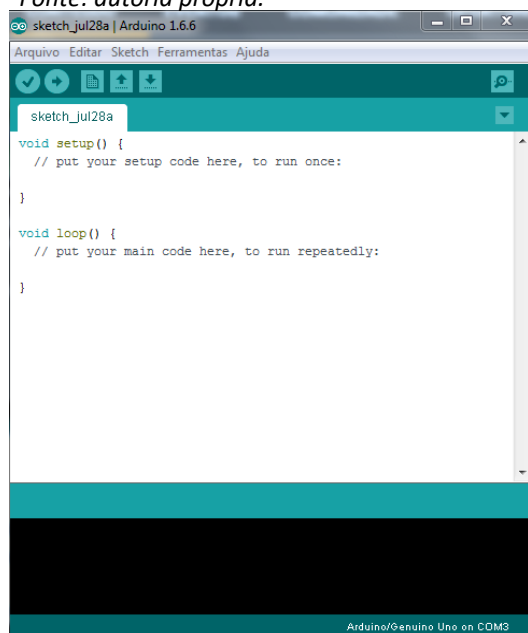
Microcontrolador	ATMEGA328P
Voltagem de operação	5V
Voltagem de entrada (recomendada)	7 – 12V
Voltagem de entrada (limites)	6 – 20V
Pinos de E/S digitais	14 (dos quais 6 podem ser saídas PWM)
Pinos de entrada analógica	6
Corrente CC por pino E/S	40mA
Corrente CC no pino de 3,3V	50mA
Memória flash	32KB dos quais 0,5KB é utilizado pelo bootloader
SRAM	2KB
EEPROM	1KB
Velocidade de clock	16MHz
Tamanho	68,6 x 53,4mm
Peso	25g

Fonte: adaptado de <<https://arduinoaprendizes.wordpress.com/2015/04/22/historiaarduino/>>

O *software* para a programação do Arduino é uma IDE (*Ambiente de desenvolvimento integrado*) que permite a criação de códigos (*sketch*), compilação, depuração de erros e outras funções importantes. Pode ser baixado gratuitamente no site do fabricante em: <https://www.arduino.cc/en/Main/Software>.

Ao abrir o IDE do Arduino temos uma janela como na figura 8. Ela é composta por três partes principais: menu na parte de cima; área de digitação do código (*sketch*) no centro; e uma janela de mensagem na área escura abaixo. Sob da barra de menu há uma barra com botões de acesso rápido com as seguintes funções:

Fonte: autoria própria.



Verificar: verifica se há erros no código digitado.



Carregar: compila o código digitado e envia para a placa se estiver conectada ao computador.



Novo: abre um novo sketch em branco.



Abrir: abre um sketch presente no sketchbook.



Salvar: salva o sketch que esta na área de digitação.



Monitor Serial: abre uma janela para monitorar a placa via comunicação USB.

Capítulo III: Instalando e Configurando o Arduino

Depois de fazer o *download* do programa do Arduino a instalação é padrão como qualquer programa da plataforma Windows. Para instalação em Linux ver em: <http://www.bosontreinamentos.com.br/electronica/arduino/instalando-o-software-arduino-ide-no-ubuntu-linux/>.

Ao utilizar o programa pela primeira vez depois de instalá-lo no computador, devemos configurar o modelo de placa que estamos utilizando e a porta COM que está conectada. Em *menu>ferramentas>placa>* e escolha o modelo de seu Arduino. Veja figura 9. Para configurar a porta COM vá no mesmo menu só que agora em *ferramentas>porta>* e selecione a porta onde está conectado a placa, caso já não esteja na porta correta, como na figura 10.

Fonte: autoria própria.

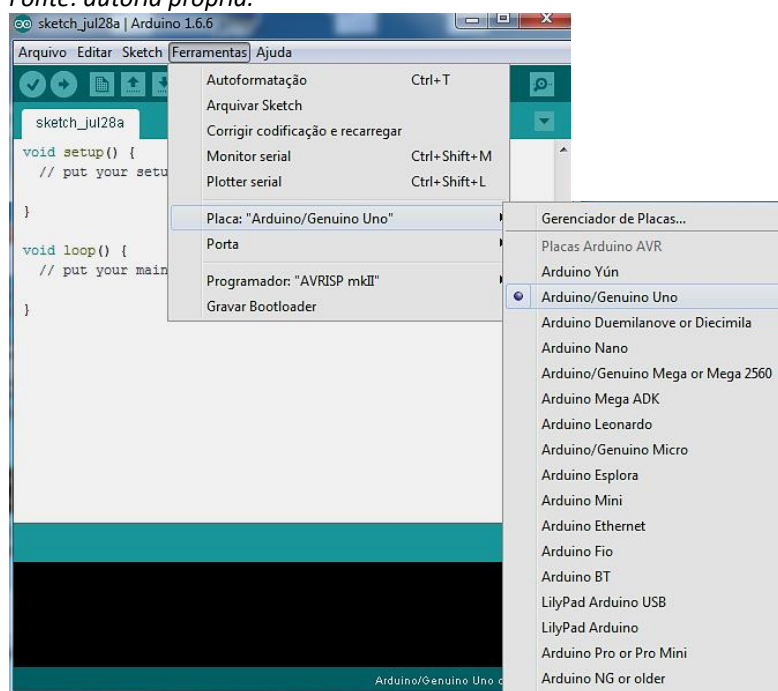


Fig. 9 – Configuração do modelo de placa.

Fonte: autoria própria.

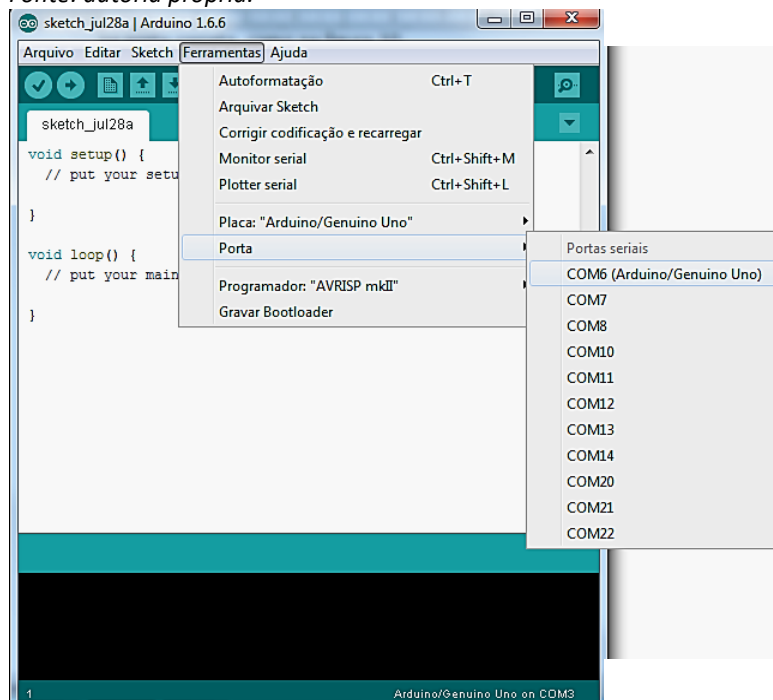


Fig. 10 – Configuração da porta COM.

Para certificar se todo o processo de configuração está correto, você pode utilizar um código de exemplo e tentar carregar. Vá ao *menu>arquivo>exemplos>basics>blink*, conforme a figura 11.

Fonte: autoria própria.

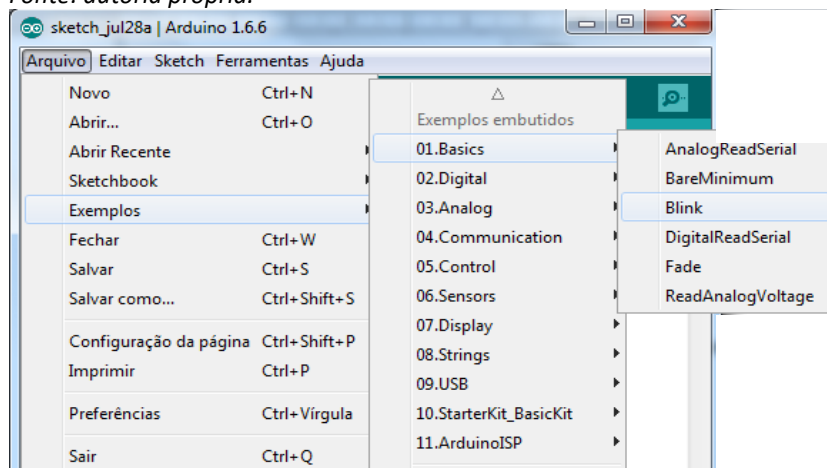


Fig. 11 – Carregando o código de exemplo blink.

Carregue o código e se tudo ocorreu corretamente deverá aparecer uma mensagem na área inferior semelhante à figura 12 e, logo em seguida, o do pino 13 da placa começará a piscar.

Fonte: autoria própria.

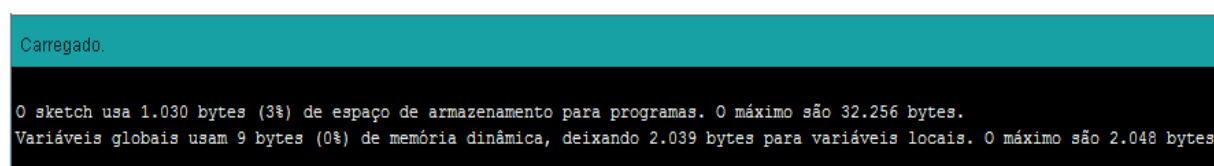


Fig. 12 – Mensagem de carregamento ao final do processo.

Capítulo IV: Programando o Arduino

Para começarmos a programar o Arduino, temos que saber o que realmente ele precisa fazer, ou seja, quais as “respostas” (saídas) que ele deve nos dar e quais são as “perguntas” (entradas) que ele precisa para que “responda” corretamente àquilo que necessitamos.

Como nas portas lógicas vistas anteriormente, a programação também funciona com o AND, OR, NOT. Dependendo do código de programação os símbolos que os representam podem ser diferentes.

Operador Lógico	Equivalência Arduino	Equivalência Planilhas Excel
AND	&&	E
OR		OU
NOT	!	NÃO

4.1 Variáveis

As variáveis são palavras que nós usamos para armazenar um valor. Elas devem ser declaradas no início ou durante o programa e precedidas do tipo de dados que ela pertence. Elas não podem ser iguais a palavras especiais que o programa usa. Para se atribuir um valor a uma variável devemos usar o sinal de igual (=). Dependendo do lugar onde são declaradas elas podem ser **variáveis globais** (que são utilizadas em qualquer bloco ou função do programa e devem ser declaradas no início do código, antes da função setup) e as **variáveis locais** (que são declaradas dentro de uma função ou bloco de controle e só poderão ser utilizadas dentro do bloco ou função onde se encontram).

Exemplo: `int valor = 0;`

Neste exemplo declaramos uma variável chamada “valor” que vale zero, esta variável é do tipo inteira.

4.2 Tipos de Dados

Os tipos de dados são referentes à variável que está sendo utilizada, pois têm relação com a capacidade (números de bits) que elas suportam. Os tipos mais usados durante nosso curso são:

int (para números inteiros de dois bytes, abrange entre -32.768 e 32.768);

unsigned int (números inteiro sem sinal, abrange de 0 a 65535);

long (números inteiros sinalizados com 32 bits, abrange de -2.147.483.648 a 2.147.483.647);

unsigned long (números inteiros sem sinal com 32 bits, abrange 0 a 4.294.967.295);

float (para valores com ponto flutuante, utilizam 4 bytes ou 32 bits, abrange $3,4 \times 10^{-38}$ a $3,4 \times 10^{38}$);

4.3 void setup() e void loop()

void setup(): o programa roda apenas uma vez antes de entrar no void loop(). Usamos para configurar os pinos e declarar as variáveis.

void loop(): depois do programa rodar o void setup() ele entra no void loop() que irá rodar o que estiver dentro deste bloco repetitivamente até a placa ser “resetada” ou cessada a sua alimentação.

4.4 Funções

4.4.1 Entradas e saídas digitais

pinMode(pino, modo): utilizado para dizer que o pino está como INPUT(entrada) ou OUTPUT(saída).

Exemplo: `pinMode(13, OUTPUT)` - o pino 13 está como saída (alto ou baixo)

digitalWrite(pino, valor): escreve em um pino especificado o valor HIGH (alto) ou LOW (baixo). Esse valor é uma tensão de 5V (HIGH) ou 0V (LOW).

Exemplo: `digitalWrite(13, HIGH)` - o pino 13 tem uma tensão de 5V.

digitalRead(pino): Lê o valor do pino digital e diz seu estado HIGH ou LOW.

Exemplo: `valor = digitalRead(7)` - a variável “valor” agora vale o estado do pino 7

4.4.2 Entradas e saídas analógicas

analogRead(pino): Lê o valor do pino analógico. Esses pinos analógicos na placa UNO vão de A0 a A5, em algumas placas há mais pinos de entrada analógica. O valor lido é um número inteiro de 0 a 1023. Essa função mede o valor da tensão no pino, mas não mostra diretamente o valor em *volts*

e sim um número inteiro de 0 a 1023. Deve-se calcular o valor da tensão através de uma conta simples.

$$V_{\text{pin}} = \frac{V_{\text{ref}} \times \text{valor lido}}{1023}$$

V_{ref} é a tensão de referência do Arduino (podemos utilizar 1.1, 3.3 ou 5 Volts. No Arduino Uno o mais utilizado é $V_{\text{ref}} = 5\text{V}$). Podem ser utilizados outros valores de referência que não são preestabelecidos, ligando o pino V_{ref} do Arduino a uma tensão que se quer de referência.

Exemplo: se o valor lido no pino analógico foi de 500, para saber a tensão correspondente usamos:

Para $V_{\text{ref}} = 5\text{V}$

$$V_{\text{pin}} = \frac{V_{\text{ref}} \times \text{valor lido}}{1023} = \frac{5 \times 500}{1023} = 2,44 \text{ volts}$$

analogWrite(pino, valor): escreve um valor entre 0 e 255 no pino especificado. Esse valor é referente ao rendimento do ciclo (duração do ciclo ativo), ou seja, vai de 0 (sempre desligado) a 255 (sempre ligado). Por exemplo, se quisermos uma tensão próxima da metade do valor de alimentação da placa (2,5V) precisamos especificar um valor de 50% do ciclo ativo (ex.: **analogWrite(9, 127)**) o que faz com que ele fique a metade do tempo ligado e a metade do tempo desligado numa frequência de aproximadamente 490Hz. A função *PWM* (modulação por largura de pulso) está presente apenas em alguns pinos digitais da placa.

4.5 Funções de Tempo

delay(valor): é usado para que o programa espere um tempo especificado para que continue a executar. O valor deve ser especificado em milisegundos (ms).

Exemplo: `delay(500)` - o programa para, conta até 0,5s e retorna a execução.

delayMicroseconds(valor): Suspende a execução por um valor especificado em microsegundos (μs).

millis() e micros(): mostra o tempo total que o programa está em execução, pode mostrar em *milisegundos* ou *microsegundos*.

4.6 Estruturas de Controle e Comparadores

if, if ... else

Uma estrutura muito utilizada quando programamos o Arduino é o **if** (se). Ela é usada junto com operadores de comparação. Verifica se a sentença é verdadeira ou falsa. Se for verdadeira executa o comando que estão entre as chaves, se for falsa os ignora. Se utilizarmos junto com o **else** (senão), no caso da sentença for falsa são executado os comandos entre as chaves do else. Dentro do **if** e **else** podem ser feito mais outros **if** e **else**, dependendo de quantas condições queremos verificar.

while(condição)

Faz um bloco de comandos que esta dentro das chaves do **while** ser executado continuamente até que a condição dentro dos parênteses não seja mais verdadeira. Ele pode ser parado também por um comando **break**, independente da condição ser verdadeira.

Exemplo:

```
b = 10;
a = 0;
while (a<b) {
a = a+1;
}
```

Neste exemplo o programa vai entrar no bloco do **while**, executar os comandos até o fim e voltar ao início, até que a condição (a<b) não seja mais verdadeira. Depois que ele sair do bloco a variável **a** vale 10. Note que sempre ele vai verificar a condição colocada em parênteses, por isso as variáveis devem ter um valor armazenado antes de entrar no **while**.

Para verificar uma condição do **if** e **while** utilizamos os operadores:

Operadores de comparação

igual a	==
diferente de	!=
menor que	<
maior que	>
menor e igual a	<=
maior e igual a	>=

Capítulo V: Praticando a Programação do Arduino

Para começarmos a programar devemos identificar os alguns botões do programa e suas partes principais. A figura 13 mostra algumas características importantes da janela que usaremos para escrever nosso código.

Fonte: autoria própria.

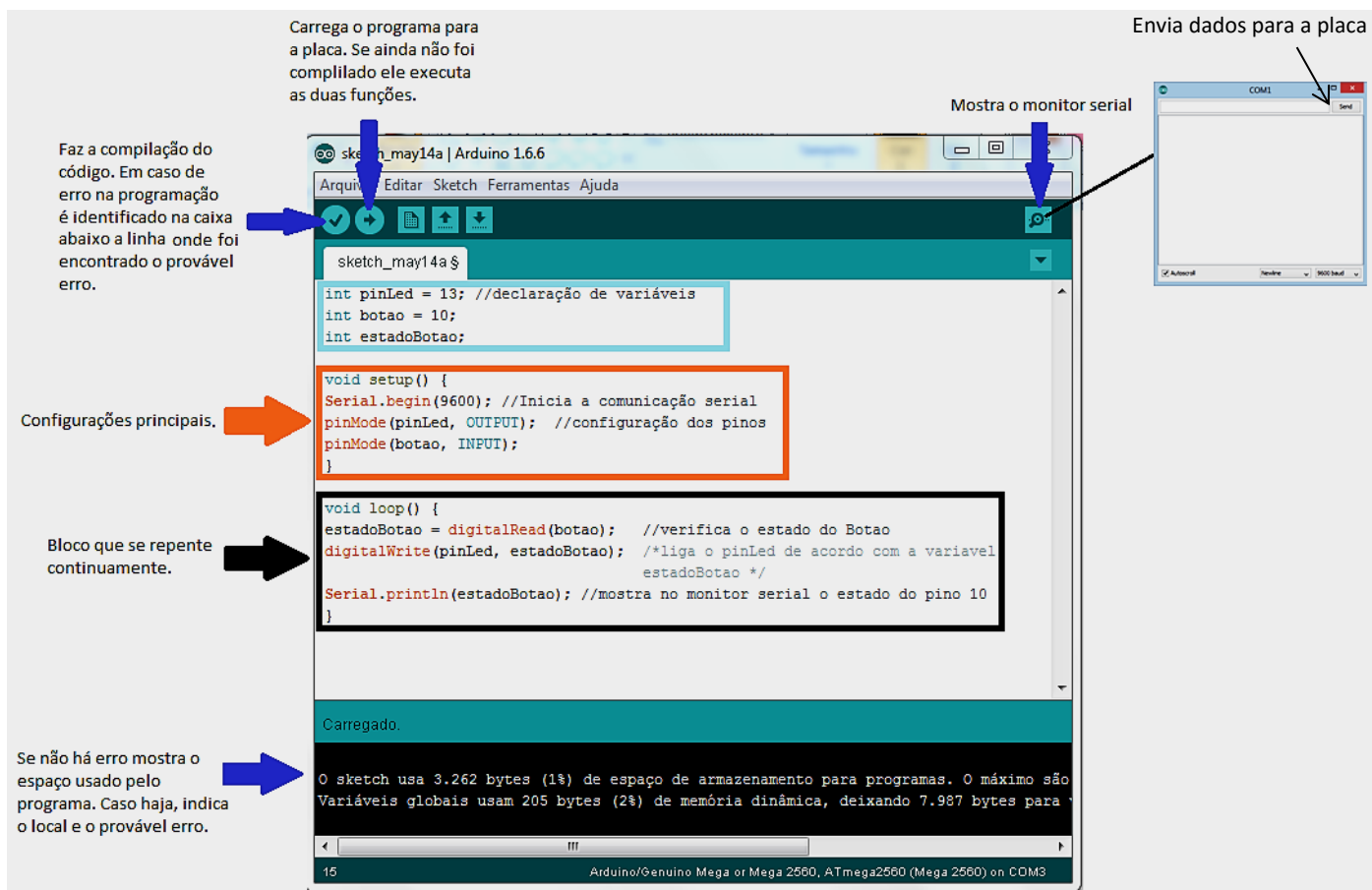


Fig. 13 – Imagem da janela usada para programar o Arduino.

Exemplo:

Queremos fazer com que o Arduino ligue um LED quando um botão for pressionado. Uma das formas mais simples de fazer isso é utilizando uma variável que armazena o estado do pino de entrada (configurado no `void setup()`) e quando o botão estiver pressionado ele modifica seu estado de baixo para alto (ou 0 para 1) e o comando `digitalWrite(pinLed, estadoBotao)` escreve no pino 13 o valor atual da variável. Para vermos a mudança do estado do pino na tela do computador acrescentamos no final de nosso código o comando `Serial.println(estadoBotao)` para mostrar no monitor serial o valor atual da variável `estadoBotao`. Abaixo o código:

```
int pinLed = 13;

int botao = 10;

int estadoBotao;

void setup() {

Serial.begin(9600);

pinMode(pinLed, OUTPUT);

pinMode(botao, INPUT);

}

void loop() {

estadoBotao = digitalRead(botao);

digitalWrite(pinLed, estadoBotao);

Serial.println(estadoBotao);

}
```

É importante lembrarmos que o nome de variáveis não deve possuir acentos, por isso **botao** e **estadoBotao** estão escritos dessa forma.

5.1 Comentários

Conforme se observa na figura 13, o código também pode conter comentários a respeito do que foi digitado em cada linha de comando e que não serão interpretados pelo programa. Isso pode ser útil para não esquecermos o que uma linha ou bloco de instrução faz. Esses comentários são digitados depois de // no caso de ser apenas uma linha. Para várias linhas de comentários podemos utilizar o /* no início e */ no final, ou seja, tudo que estiver dentro de /*...*/ será tratado como comentário. Como no exemplo:

```

estadoBotao = digitalRead(botao);    //verifica o estado do Botão
digitalWrite(pinLed, estadoBotao);  /*liga o pinLed de acordo com a variável
                                     estadoBotao */

```

└──────────────────────────────────┘
└──────────────────────────────────┘

Linhas de comando
Comentários

Observe que os comentários podem ser escritos com acentos.

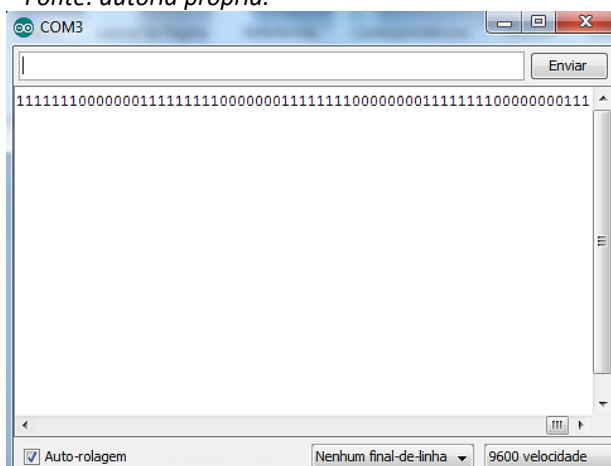
5.2 Monitor Serial e o Comando Serial.print()

Um comando bastante usado como forma de visualizar no **Monitor Serial** o que está acontecendo com o Arduino é o **Serial.print()**. Podemos usá-lo para obter um valor de uma variável, cálculo, medida ou digitar um texto colocando entre aspas. O comando `Serial.print()` escreve na mesma linha, já o comando `Serial.println()` escreve e troca o cursor para linha seguinte.

Para iniciar a comunicação serial usamos o comando **Serial.begin(9600)**. Esse valor dentro dos parênteses é a velocidade de comunicação em bits por segundo (bps).

Depois de compilar e carregar o código podemos abrir o monitor clicando no ícone da lupa, localizado na parte superior direita do programa do Arduino. A caixa de texto da parte superior pode enviar valores de entrada para a placa. O monitor serial não escreve textos com acentos.

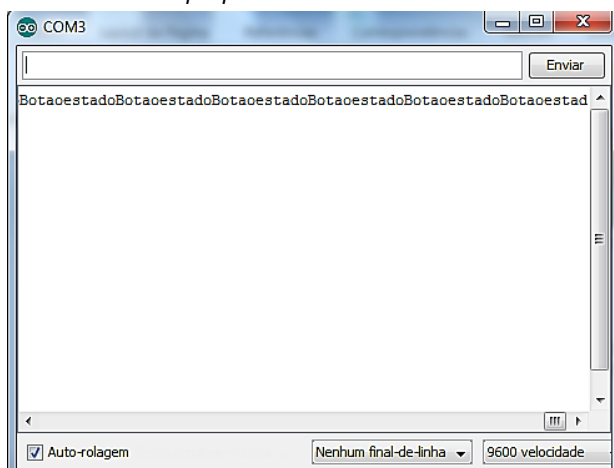
Fonte: autoria própria.



Serial.print(estadoBotao): mostra no monitor o valor da variável `estadoBotao` sempre na mesma linha.

Fig. 14 – Monitor serial com `Serial.print()` usando variável.

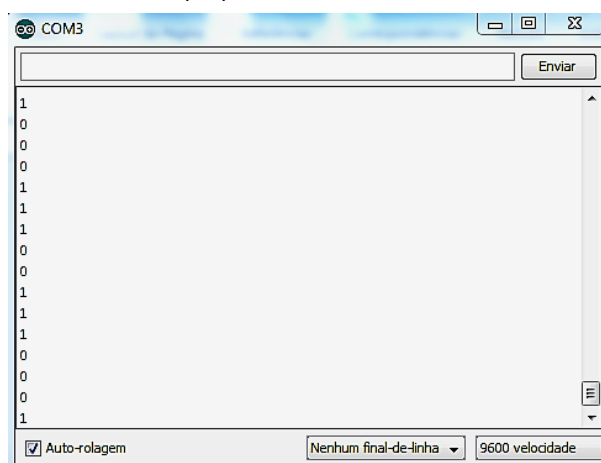
Fonte: autoria própria.



Serial.print("estadoBotao"): mostra no monitor a palavra estadoBotao repetidamente na mesma linha;

Fig. 15 – Monitor serial com Serial.print() usando texto.

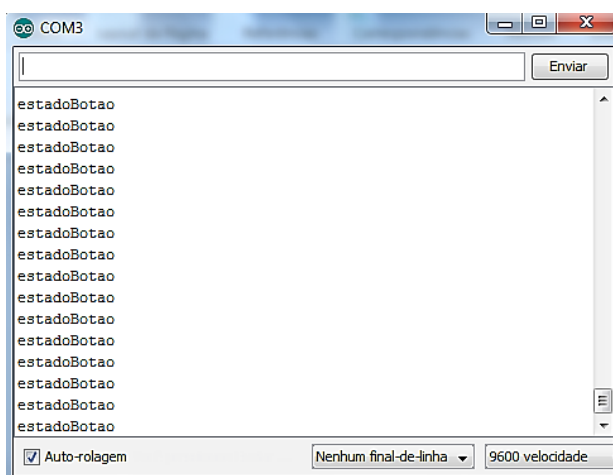
Fonte: autoria própria.



Serial.println(estadoBotao): mostra no monitor o valor da variável estadoBotao e troca de a linha;

Fig. 16 – Monitor serial com Serial.println() usando variável.

Fonte: autoria própria.



Serial.println("estadoBotao"): mostra no monitor a palavra estadoBotao repetidamente em linhas diferentes.

Fig. 17 – Monitor serial com Serial.println() usando texto.

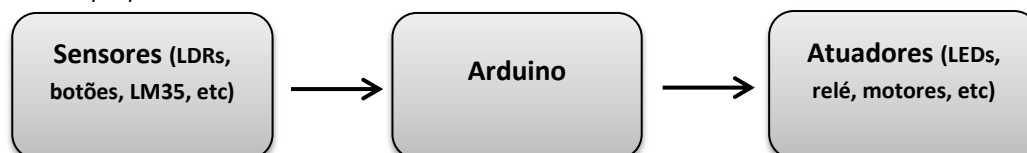
Atividades:

- 1) Faça um programa que o pino digital 13 (LED fixo na placa) ligue e desligue continuamente a uma frequência de 2Hz. Deve mostrar no monitor serial cada vez que o pino de saída mudar seu estado. Dica: usar a função `delay()`.
- 2) Modifique o programa da atividade 1 e faça o Arduino executar continuamente os seguintes passos: contar até 10; mostrar a frase "O LED será ligado!"; ligar o LED; contar até 5; mostrar a frase "O LED será desligado!"; e desligar o LED. Faça antes um projeto no papel com os passos que o programa deve executar.
- 3) Faça um programa para teste de pilhas. Usando um pino analógico do Arduino ele deve verificar a tensão da pilha. Caso estiver com a tensão acima de 1,4V o LED do pino 13 (fixo na placa) deverá ficar ligado. Se estiver abaixo de 1,4V o LED fica piscando em uma frequência de 1Hz. No monitor serial deverá aparecer a tensão da pilha em volts cada vez que ela for conectada à placa. Faça também um esquema no papel antes de programar. Dica: usar as funções `analogRead()`; `if ... else` e `Serial.println()`.

Capítulo VI: Montando Projetos com o Arduino

Começaremos a estudar alguns conceitos importantes que iremos utilizar para começar a fazer nossos primeiros projetos com o Arduino. Na construção de projetos mais elaborados, precisamos saber montar alguns circuitos que funcionarão integrados a nossa placa Arduino. Para isso, utilizaremos *protoboards*, *jumpers*, *resistores*, *LEDs*, *relés*, *transistores*, *sensores* e demais componentes que serão comentados assim que aparecerem no texto. Normalmente, um projeto é composto basicamente por três partes:

Fonte: autoria própria.



Podemos entender melhor essa relação quando pensamos em uma atividade do nosso dia a dia, como consumir uma bebida quente. Normalmente não bebemos antes de testar a temperatura para garantir que não iremos nos queimar. Colocamos nos lábios ou tocamos com a mão devagar (nossos sensores); logo nosso cérebro faz o papel do Arduino recebendo e analisando os dados (verificando se está quente a ponto de nos queimar); em seguida executamos a ação de beber (atuadores) ou de esperar até que ela esfrie mais.

6.1 Sensores

Quando precisamos implementar projetos de automação, usando microcontroladores e circuitos eletrônicos, é necessário que medidas externas sejam feitas para que, com base nelas, ocorram ações. Podemos dizer que automatizar um sistema é agir conforme estímulos exteriores sejam eles físicos ou químicos de qualquer processo. Por exemplo, ligar uma resistência elétrica quando a medida de temperatura de um tanque de água atingir certo valor. Para realizarmos essas medidas precisamos de **sensores** que são dispositivos capazes de modificarem suas características de forma automática. Há vários tipos de grandezas que sensores podem medir como: temperatura, pressão, umidade, distância e outros. Podemos classificá-los com discretos e contínuos.

- **Sensores discretos:** possuem apenas dois estados ligado/desligado. Uma simples chave fim de curso de um portão elétrico é um exemplo.
- **Sensores contínuos:** monitoram uma grandeza física numa faixa de valores contínuos e a saída do sensor é proporcional a essa grandeza. Por exemplo, o botão de volume (potenciômetro) de um rádio ou TV antiga (os mais modernos possuem botões digitais).

Os sensores também se classificam pelos princípios físicos com que funcionam podendo ser resistivos, capacitivos, indutivos, magnéticos, fotoelétricos, ultrassônicos, etc.

6.2 Atuadores

A partir das medidas realizadas pelos sensores e passando pelo Arduino (que possui os códigos que decidem como ele deve “agir”) precisamos ter algo que atue no meio físico. Os atuadores são qualquer tipo de elemento pelo qual o Arduino irá responder externamente, ou seja, podemos dizer que um atuador é a resposta final a partir de parâmetros preestabelecidos por nós através da programação e usando as variáveis verificadas pelos sensores. Podem ser LEDs indicadores, motores, relés, válvulas, *Buzzers* (pequenos indicadores sonoros), etc.

6.3 Divisor de Tensão

O divisor de tensão é um circuito que, como o nome já diz, faz a divisão de uma tensão aplicada em seus extremos. Nas figuras 18a e 18b vemos duas formas de representação desse circuito. No exemplo da fig. 18b, a tensão de 10V está sendo dividida pelas resistências que têm o mesmo valor, fazendo a tensão de saída do divisor ser de 5V em relação ao terra (GND) do circuito. Essa proporção de tensão de saída (V_{out}) depende do valor das resistências usadas e da tensão de entrada (V_{in}), dado pela equação:

$$V_{out} = \frac{R_2}{(R_1+R_2)} V_{in} \rightarrow (R_1 \neq R_2)$$

$$V_{out} = \frac{1}{2} V_{in} \rightarrow (R_1 = R_2)$$

Fonte: autoria própria.

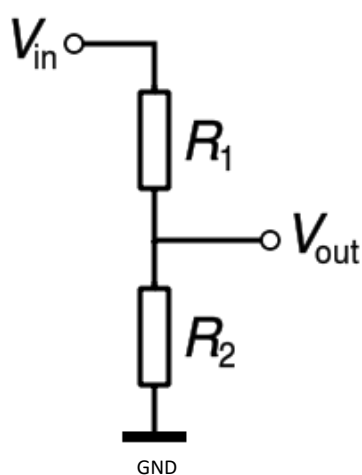


Fig. 18a – Divisor de tensão

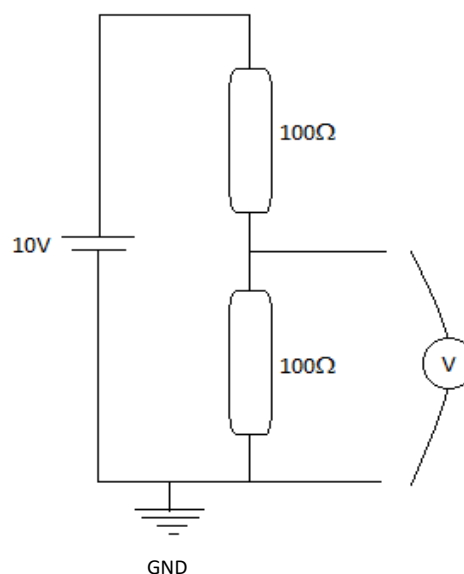


Fig. 18b – Fonte de 10V ligada a um divisor de tensão com resistências iguais.

Os pinos de entrada (*input*) do Arduino, tanto os digitais como os analógicos não reconhecem nenhuma grandeza a não ser tensão elétrica. Então, qualquer sensor ligado a ele deve enviar uma medida em *volts* que deverá ser 0 ou 5V (LOW ou HIGH) para entradas digitais e de 0 até 5V para entradas analógicas (com a precisão de 0,0049V ou 4,9mV). As tabelas 1 e 2 mostram a relação dos valores de tensão reais e como o Arduino interpreta essa tensão.

Entradas Digitais		
Tensão (V)	0	5
Arduino	0	1
	LOW	HIGH

Fonte: autoria própria.

Entradas Analógicas									
Tensão (V)	0	0,0049	0,0098	0,0147	0,0196	0,0245	0,0294	...	5
Arduino (10bits)	0	1	2	3	4	5	6	...	1023

Fonte: autoria própria.

Multiplicando **0,0049V** pelo valor lido no Arduino (**analogRead(pino)**) fica fácil encontrar a tensão real a partir do valor mostrado pela função **Serial.println(pino)**.

Exemplo:

1) Monte um circuito divisor de tensão.

- uma 2 resistores de 1500Ω;
- ligue uma extremidade no pino 5V e outra no GND;
- ligue a junção dos dois resistores no pino A0;

Fonte: autoria própria.

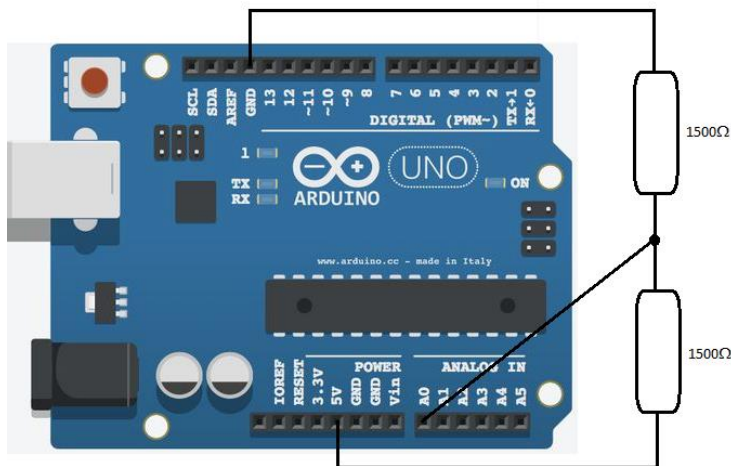


Figura 19 – Ligação do divisor de tensão à placa Arduino.

- 2) Calcule a tensão de saída do divisor de tensão.

$$V_{out} = \frac{R_2}{(R_1 + R_2)} V_{in}$$

- 3) Construa o código no Arduino e com dois resistores diferentes verifique o número que aparece no monitor serial, como na figura 20.

Fonte: autoria própria.

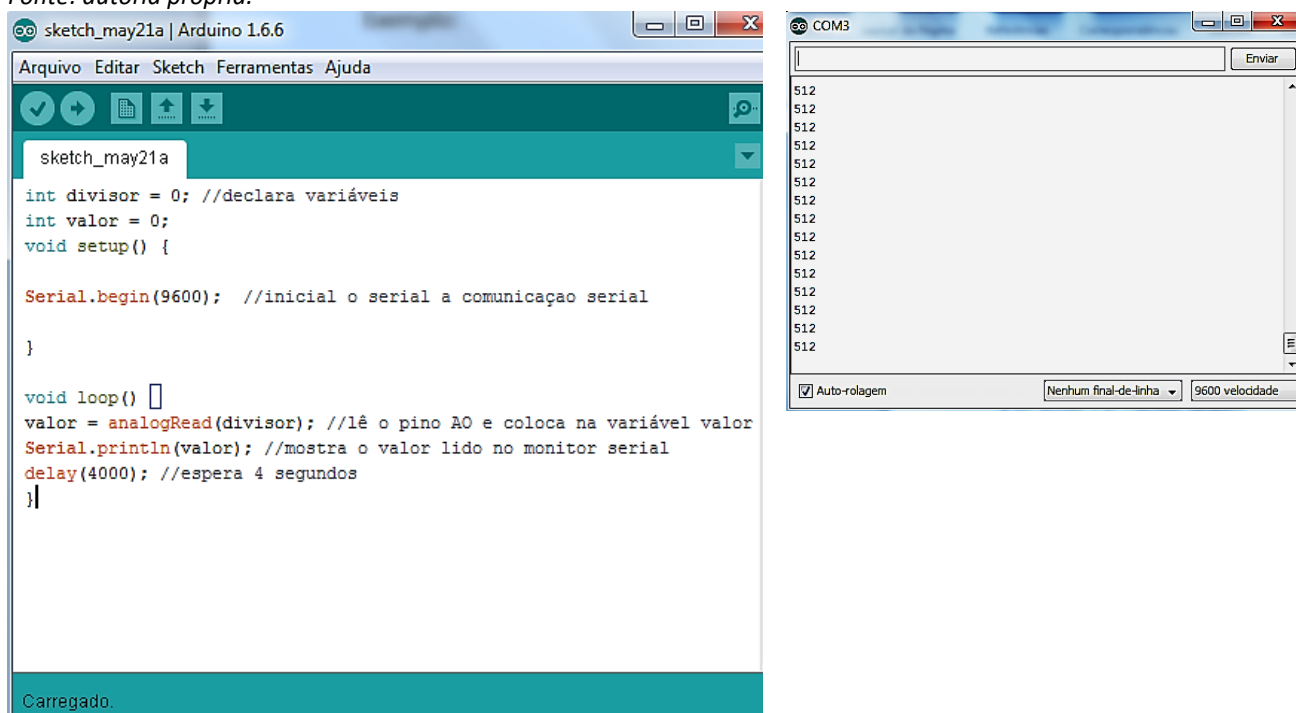


Fig. 20 – Código do exemplo acima e o monitor serial com o valor do pino A0.

- 4) Modifique o programa acima para aparecer no monitor a tensão real do divisor de tensão.

$$V_{real} = \text{valor lido} \times 0,0049$$

6.4 LDR e Resistor

O LDR (*Light Dependent Resistor*) ou em português *resistor dependente de luz* é um componente eletrônico passivo que diminui sua resistência conforme aumenta a intensidade de luz que incide sobre ele. Diferentemente dos resistores que possuem um valor de resistência fixo que geralmente são indicados por um código de faixas de cores impressas no próprio componente. A fig. 9 mostra as simbologias mais utilizadas para o LDR e para o resistor serem identificados nos desenhos de circuitos eletrônicos.

Fonte: autoria própria.

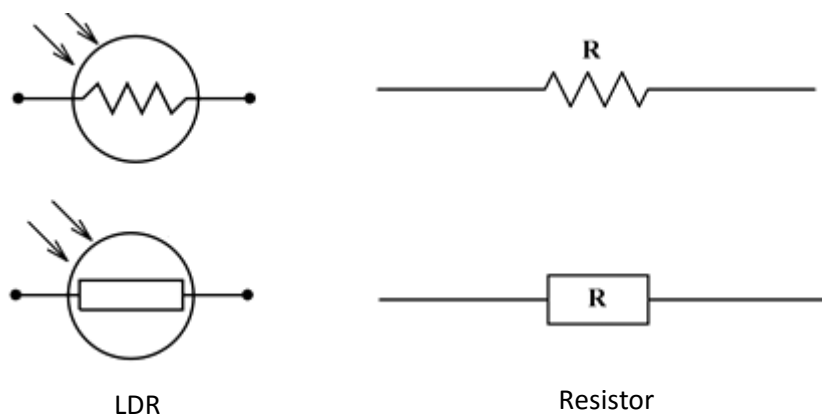
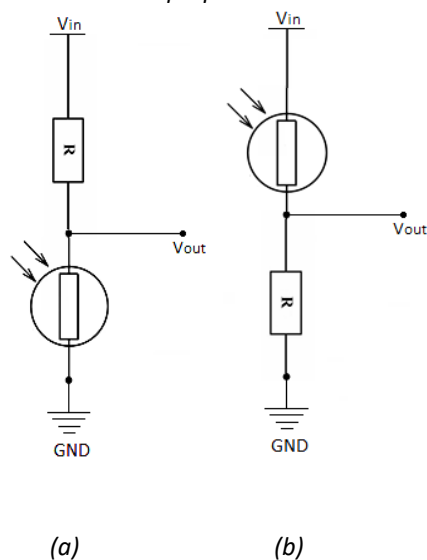


Fig. 21 – Simbologias do LDR e do resistor mais utilizadas.

Como podemos ver o LDR é um sensor que podemos utilizar para várias aplicações em projetos com o Arduino. Mas como ele apenas varia sua resistência e o Arduino mede apenas tensão elétrica, como podemos fazer para que atue como um sensor em uma entrada analógica?

Então, precisamos fazer uma variação de resistência elétrica ter uma saída de variação de tensão elétrica. Para isso usamos um divisor de tensão como o da fig. 6, em que colocamos um resistor fixo junto com LDR, assim cada vez que variar a resistência de um deles (no caso só poderá variar o LDR, pois o resistor se manterá fixo) a tensão V_{out} também varia.

Fonte: autoria própria.



A fig. 10 mostra as duas formas de montar o circuito. Quando o LDR é ligado ao GND (fig. 22a) a tensão de saída V_{out} diminui com o aumento da luminosidade, pois o LDR diminui sua resistência. Quando ligado como na fig. 22b a tensão V_{out} aumenta com o aumento a luminosidade, pois ao diminuir a resistência do LDR mais tensão ficará no resistor. De acordo com a equação, vista anteriormente:

$$V_{out} = \frac{R_2}{(R_1 + R_2)} V_{in}$$

Figura 22 – Divisor de tensão com resistor e LDR em duas configurações.

Atividades:

- 1) Faça um divisor de tensão com resistores diferentes e calcule a tensão de saída V_{out} usando como tensão de entrada $V_{in} = 5V$. Ligue o divisor na placa Arduino e faça um programa que leia a tensão de saída e compare com o valor calculado.
- 2) Faça um programa usando um LDR que funcione como um botão digital e quando pressionado uma vez ligue o LED da placa, pressionando duas vezes o LED desligue. Faça dois circuitos:
 - a) um com o LDR ligado ao GND;
 - b) um com o resistor ligado ao GND.
 O que mudou? O que é preciso fazer no programa para que os dois circuitos tenham o mesmo resultado (ligar e desligar o LED ao pressionar o botão)?
- 3) Faça um programa usando o LDR que mostre no monitor serial a situação de claridade do ambiente entre: “Muito Claro”; “Claro”; “Pouco Claro”; “Escuro”.

6.5 Sensor de Temperatura LM35

O sensor LM35 (fig. 23a) é um CI (circuito integrado) que possui tensão de saída linearmente proporcional à temperatura em graus Celsius. A fig. 23b mostra o circuito típico do LM35 que pode medir temperaturas de 2 a 150°C. A tensão de alimentação V_{in} pode ser de 4 a 20V e a tensão de saída V_{out} será de 0,01V/°C ou 10mV/°C.

Fonte: autoria própria.

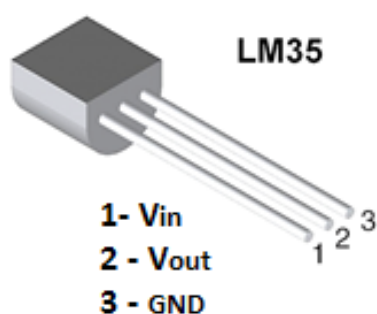


Fig. 23a – Circuito integrado LM35 com identificação de seus pinos.

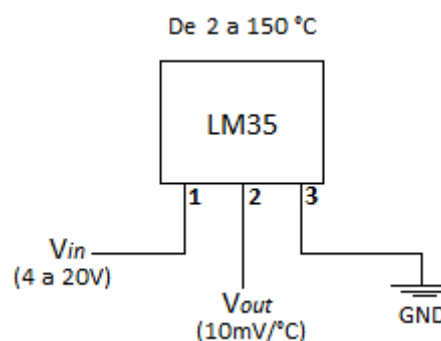
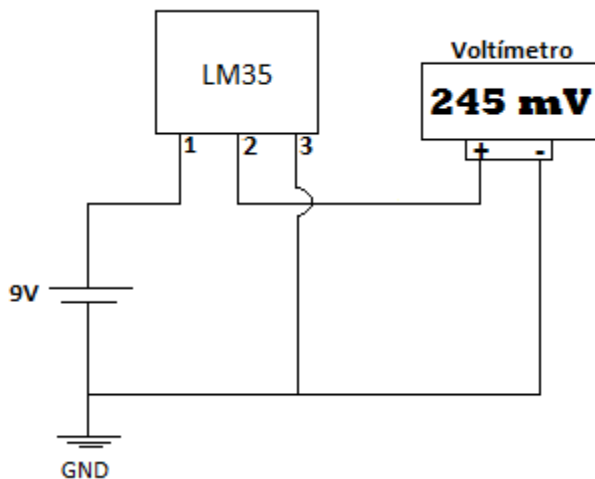


Fig. 23b – Circuito típico para medir temperaturas de 2 a 150°C.

6.5.1 Aplicação do LM35

Se utilizarmos apenas um voltímetro e uma bateria de 9V, já podemos obter a temperatura montando um circuito bem simples.

Fonte: autoria própria.



Na figura 24, temos o pino 1 (V_{in}) ligado ao polo positivo da bateria e o pino 3 (GND) ligado ao polo negativo da bateria. O voltímetro está medindo a tensão entre o pino 2 (V_{out}) e o pino 3. Como a medida é 245mV, temos:

$$Temperatura (^{\circ}C) = \frac{Tensão\ pino\ 2}{10mV}$$

$$Temperatura (^{\circ}C) = \frac{245mV}{10mV} = 24,5^{\circ}C$$

Fig. 24 – Circuito para medir temperatura utilizando uma bateria de 9V e um voltímetro.

6.5.2 Utilizando o LM35 com o Arduino

Agora que já sabemos como o sensor de temperatura LM35 funciona, podemos ligá-lo a uma entrada analógica da placa Arduino, obter a temperatura e utilizar essa informação para controlar algum atuador, como por exemplo, acionar um *cooler* extra quando a temperatura do seu computador atingir certo valor ou apenas mostrar essa temperatura no monitor serial.

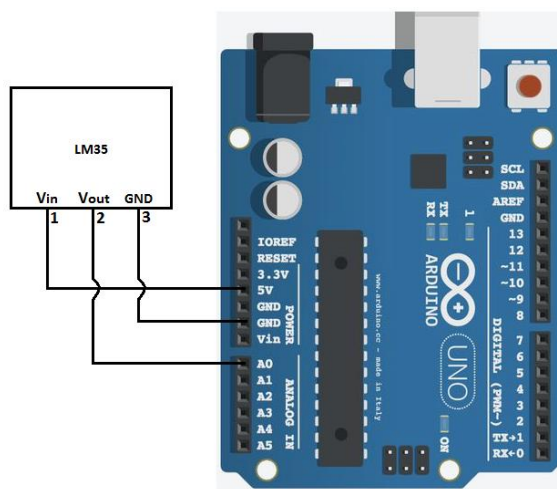


Fig. 25 - Ligação básica do LM35 ao Arduino.

Exemplo:

Vamos mostrar no monitor serial o valor da temperatura em graus Celsius ($^{\circ}C$). Primeiro ligamos os pinos 1, 2 e 3 do LM35 respectivamente nos pinos 5V, A0 e GND do Arduino (fig. 25). Depois digitamos o código da figura 26, compilamos e carregamos.

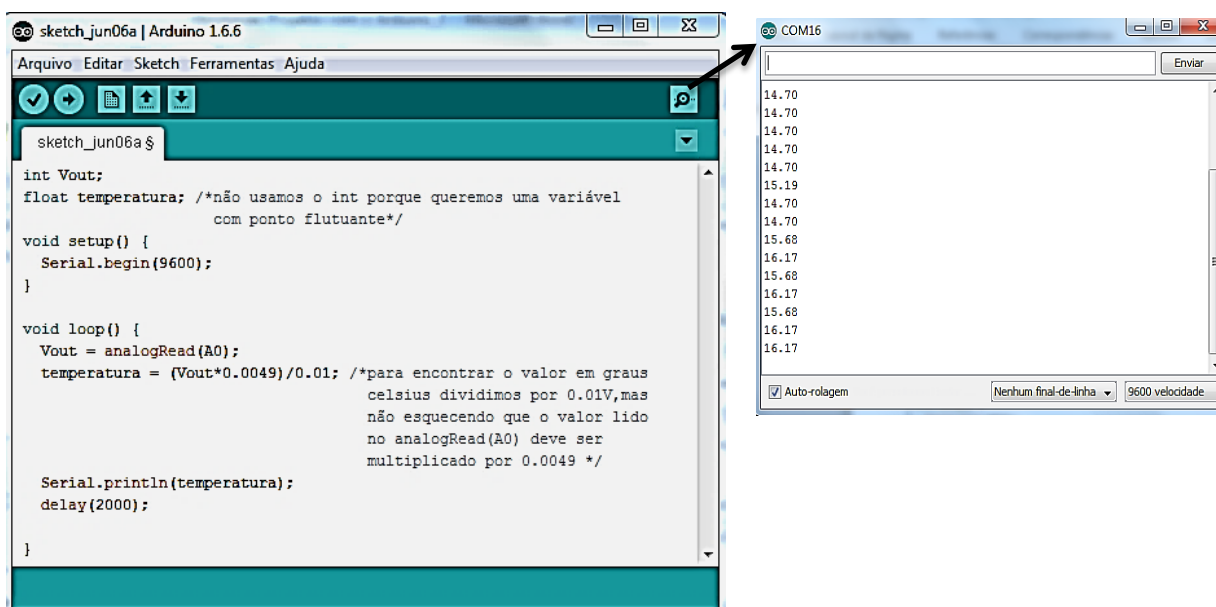


Fig. 26 – Código para mostrar a temperatura (em °C) e o monitor serial.

Atividades:

- 1) Monte o circuito para medir a temperatura usando um voltímetro. Dica: use a tensão de 5V e o GND da placa Arduino.
- 2) Faça um programa que mostre a temperatura no monitor em três escalas: Celsius, Fahrenheit e Kelvin. Utilize a relação: $\frac{T_C}{5} = \frac{T_F - 32}{9} = \frac{T_K - 273}{5}$
- 3) Faça um programa que controle o aquecimento da água do chimarrão. Quando atingir a temperatura ideal ($\pm 70^\circ\text{C}$) deverá soar um sinal sonoro no *buzzer*. Dica: Utilize uma temperatura de referência menor para não ter que utilizar água muito quente.
- 4) Faça um circuito usando o LDR e o LM35 que deverá ligar um pequeno motor quando a temperatura aumentar 5°C e desligar quando diminuir 6°C . O sistema todo somente deverá funcionar se a sala estiver com as luzes acesas.
- 5) Faça um programa que controle um pequeno motor por temperatura, ou seja, quanto mais quente maior sua rotação.

Referências

APOSTILA ARDUINO com aplicações baseada na placa: ARDUINO UNO. Disponível em:
<<http://www.valdick.com/files/ApostilaArduinoIntroducao.pdf>> acesso em 29 de julho de 2016.

<<http://arduino.cc/en/Reference/HomePage>> acesso em 26 de maio de 2016.

<<http://playground.arduino.cc/Portugues/HomePage>> acesso em 21 de maio de 2016.

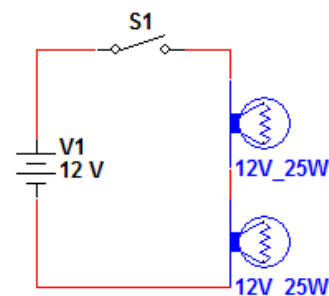
<<http://slides.com/jurasec/nodebots--2#/3>> acesso em 23 de maio de 2016.

<<http://www.embarcados.com.br/arduino-primeiros-passos>> acesso em 21 de maio de 2016.

APÊNDICE C – Questionário 1

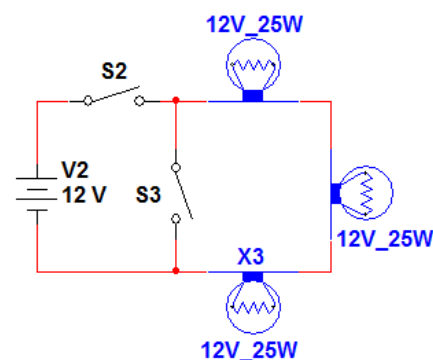
Responda as perguntas com base nas figuras:

- 1) A figura ao lado é um circuito formado por uma bateria, uma chave liga/desliga e dois resistores. A chave está aberta, então podemos dizer que:



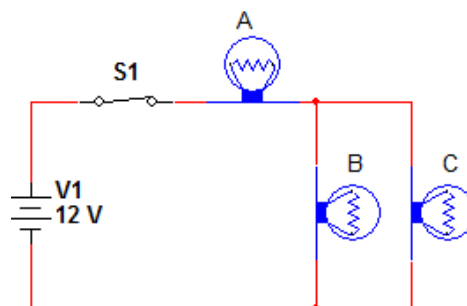
- a) Não há corrente elétrica fluindo no circuito
 b) A corrente elétrica é muito pequena, mas passa por R1 e R2
 c) Se fecharmos a chave S1 haverá um curto-circuito
 d) Não é possível saber sem medir com instrumentos

- 2) A figura ao lado é um circuito formado por uma bateria, duas chaves liga/desliga e três lâmpadas. Se fecharmos a chave S2 e S3 o que acontecerá?



- a) As três lâmpadas ficarão acessas com seu brilho normal
 b) Haverá um curto-circuito
 c) As lâmpadas ficarão acessas, mas com um brilho muito fraco
 d) Somente uma lâmpada ficará acessa

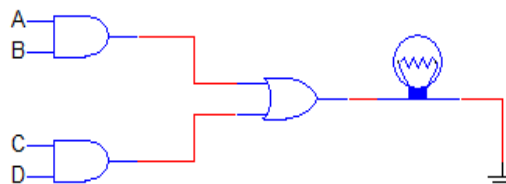
- 3) O circuito ao lado possui as lâmpadas A, B e C e a chave S1 está fechada, então podemos dizer que:




- a) Se a lâmpada C queimar as outras não acenderam
 b) Se a lâmpada B queimar a lâmpada A não acenderá
 c) Se as lâmpadas B e C queimarem a lâmpada A continuará acessa
 d) Se a lâmpada A queimar as lâmpadas B e C não acenderão

- 4) A lâmpada poderá acender se as entradas A, B, C e D forem respectivamente iguais a:

- a) 1 0 1 0
 b) 0 0 0 1
 c) 0 0 1 1
 d) 0 1 1 0



- 5) No código ao lado queremos fazer o LED do pino 13 piscar de 1 em 1 segundo. Este programa vai funcionar corretamente? Há ajustes a fazer? Quais são eles?



```
Arquivo Editar Sketch Ferramentas
sketch_may22a $
int Led = 13;

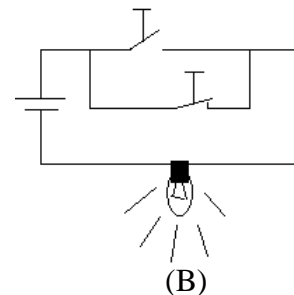
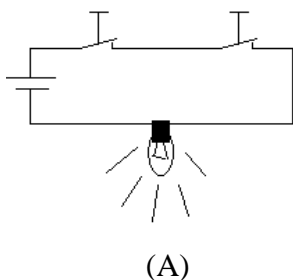
void setup() {
  pinMode(Led, INPUT);
}

void loop() {
  digitalWrite(led, 1);
  delay(1000);
  digitalWrite(led, 0);
  delay(1000);
}
```

APÊNDICE D – Questionário 2

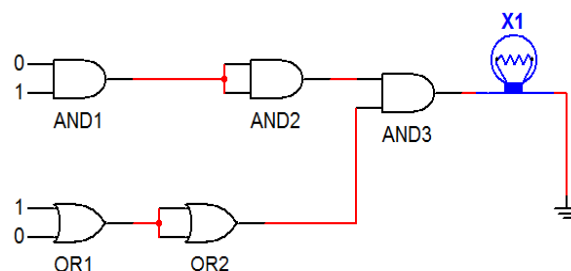
Nome: _____ Data: __/__/2016

- 1) As figuras A e B abaixo são representações de circuitos elétricos que simulam portas digitais (os interruptores representam as entradas e as lâmpadas representam as saídas). Marque a alternativa que você considera correta. Obs.: Considere que as duas lâmpadas estão ligadas.



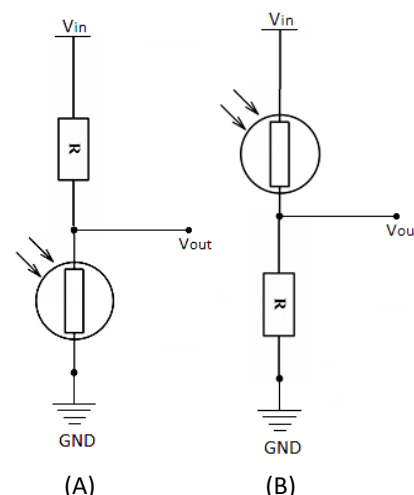
- a) A figura A se trata de uma porta digital OR (OU), pois se um interruptor abrir a lâmpada apagará.
- b) A figura B representa uma porta digital AND (E), pois tem como entradas 0 e 1 (desligado e ligado) e, mesmo assim, a lâmpada está acesa.
- c) A figura B deveria estar com a lâmpada apagada, pois uma porta AND (E) precisa de todas as suas entradas ligadas para ter uma saída ligada.
- d) A figura A se trata de uma porta AND (E), pois sendo suas entradas 1 e 1 (ligado e ligado) a lâmpada ficará acesa.
- 2) O circuito digital abaixo possui 4 entradas e sua saída é representada por uma lâmpada. Marque a alternativa correta.

- a) Para que a lâmpada acenda temos que colocar na porta **OR1** as entradas 1 e 1.
- b) Assim como está na figura a lâmpada irá acender.
- c) Trocando a porta **AND3** por uma **OR** a lâmpada acenderá.
- d) Trocando a porta **OR1** por uma **AND** a lâmpada acenderá.



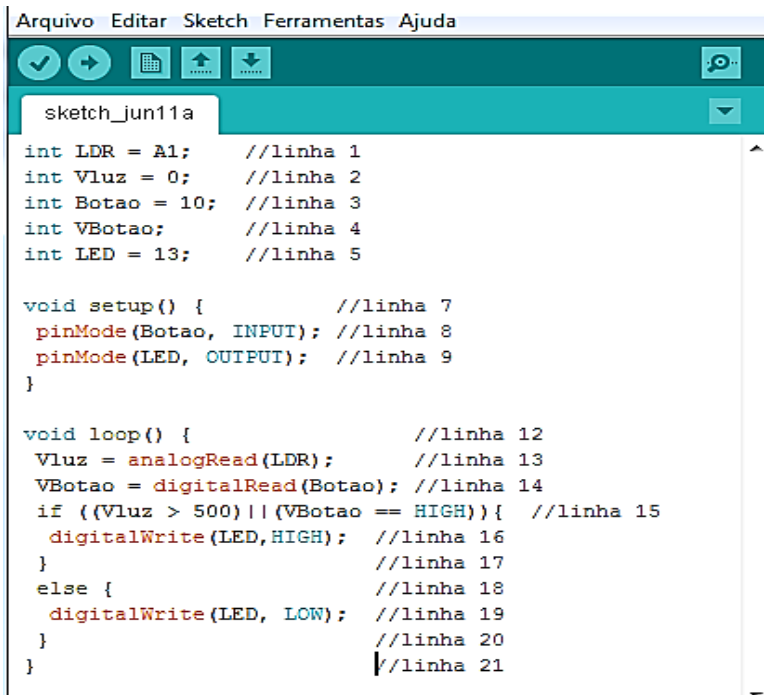
- 3) Nos circuitos A e B temos um sensor LDR ligado a um resistor fixo R. Considerando que o LDR diminui sua resistência conforme aumenta a intensidade de luz que incide sobre nele podemos dizer que:

- a) No circuito A, se a incidência de luz aumentar a tensão de saída (V_{out}) irá aumentar.
- b) No circuito B, se a incidência de luz aumentar a tensão de saída (V_{out}) irá aumentar.
- c) Nos dois circuitos a tensão de saída (V_{out}) aumenta quando aumenta a incidência de luz.



- d) Mesmo variando a resistência do LDR a tensão de saída (V_{out}) será sempre a metade da tensão de entrada (V_{in}), pois é um divisor de tensão.

De acordo com o código da figura abaixo, responda as questões de 4 a 6.



```

Arquivo  Editar  Sketch  Ferramentas  Ajuda
sketch_jun11a
int LDR = A1; //linha 1
int Vluz = 0; //linha 2
int Botao = 10; //linha 3
int VBotao; //linha 4
int LED = 13; //linha 5

void setup() { //linha 7
  pinMode(Botao, INPUT); //linha 8
  pinMode(LED, OUTPUT); //linha 9
}

void loop() { //linha 12
  Vluz = analogRead(LDR); //linha 13
  VBotao = digitalRead(Botao); //linha 14
  if {(Vluz > 500)|| (VBotao == HIGH)}{ //linha 15
    digitalWrite(LED,HIGH); //linha 16
  } //linha 17
  else { //linha 18
    digitalWrite(LED, LOW); //linha 19
  } //linha 20
} //linha 21

```

4) As linhas 8 e 9 tem a função de:

- Identificar os pinos e declarar as variáveis.
- Configurar os pinos como entrada e saída digitais.
- Configurar os pinos como entrada e saída analógicas.
- Poderiam ser colocadas acima do “voidsetup()” (linha 6, por exemplo)

5) Sobre as linhas que estão dentro do “voidloop()”, podemos dizer que:

- Se **Vbotao** estiver em estado alto, o LED ligará, independente do valor de **Vluz**
- Se a variável **Vluz** for igual 400 (**Vluz** = 400), não há a possibilidade do LED ligar.
- O comando “if” (linha 15) não depende dos valores das variáveis **Vluz** e **VBotao**.
- O comando “else” (linha 18) será executado sempre, independente do comando da linha 15.

6) Complete com o número da linha correspondente:

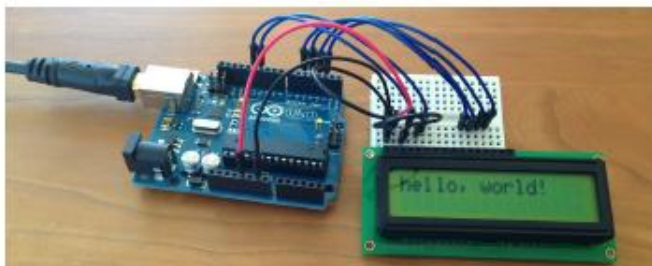
- I) Liga o LED (linha ____);
- II) Declara uma variável de um pino analógico (linha ____);
- III) Configura um pino como entrada digital (linha ____);
- IV) Coloca em uma variável o valor lido no pino analógico (linha ____);
- V) Coloca em uma variável o valor lido no pino digital (linha ____);
- VI) Usa um operador lógico **OU** (linha ____);
- VII) Configura um pino como saída digital (linha ____);
- VIII) Declara a variável que receberá o estado do botão (linha ____);

APÊNDICE E – Formulário online de inscrição

14/11/2017

Inscrição Curso de Arduino para Professores

Inscrição Curso de Arduino para Professores

***Obrigatório**

1. Nome: *

2. Sobrenome: *

3. E-mail de contato: *

4. Telefone de contato: *

5. Cidade: *

6. Formação: *

Marcar apenas uma oval.

- Graduação *Ir para a pergunta 20.*
- Especialização *Ir para a pergunta 17.*
- Mestrado *Ir para a pergunta 18.*
- Outro: _____ *Ir para a pergunta 19.*

7. Está atuando em sala de aula? *

Marcar apenas uma oval.

- Sim *Ir para a pergunta 15.*
- Não

https://docs.google.com/forms/d/10vN6lG8bc-mvPfl_TwWOiHcrSk8EWOJ2coOsNW2D2SI/edit

14/11/2017

Inscrição Curso de Arduino para Professores

8. Tem experiência com alguma linguagem de programação? *

Marcar apenas uma oval.

- Sim *Ir para a pergunta 21.*
- Não *Ir para a pergunta 9.*

9. Comente sobre sua experiência com o uso de tecnologias em sala de aula ou no dia a dia (computadores, tablets, celulares etc.)

10. Qual sua experiência com o uso de plataformas de ensino a distância (cursos a distância)? *

11. Tem disponibilidade de horários aos sábados de manhã e tarde? *

Marcar apenas uma oval.

- Sim
- Não

12. Tem disponibilidade de horários na sua escola para aplicar algumas atividades com seus alunos? *

Marcar apenas uma oval.

- Sim. Posso dispor de algumas aulas para a aplicação de atividades.
- Sim. Posso dispor de encontros com os alunos em turnos inversos.
- Não. Mas posso aplicar em horários de outro professor.
- Não há possibilidade de aplicar atividades na minha escola.

13. Quantas horas durante a semana você poderá se dedicar ao curso? *

Marcar apenas uma oval.

- menos de 2 horas
- de 2 a 4 horas
- de 4 a 6 horas
- mais de 6 horas
- Somente fins de semana

14/11/2017

Inscrição Curso de Arduino para Professores

14. Você dispõe de notebook ou netbook para usar durante o curso? **Marcar apenas uma oval.* Sim Não*Ir para a pergunta 22.***15. Qual a disciplina? ***

16. Quais os conteúdos que está trabalhando? *

*Ir para a pergunta 8.***17. Especialização em: ***

*Ir para a pergunta 7.***18. Mestrado em: ***

*Ir para a pergunta 7.***19. Titulação ***

*Ir para a pergunta 7.***20. Graduação em: ***

*Ir para a pergunta 7.***21. Comente sobre suas experiências com programação. ***

Ir para a pergunta 9.

14/11/2017

Inscrição Curso de Arduino para Professores

22. Já quer enviar sua inscrição? **Marcar apenas uma oval.* Sim Não. Quero voltar ao início e revisar minhas respostas [Comece este formulário novamente.](#)

Powered by
 Google Forms

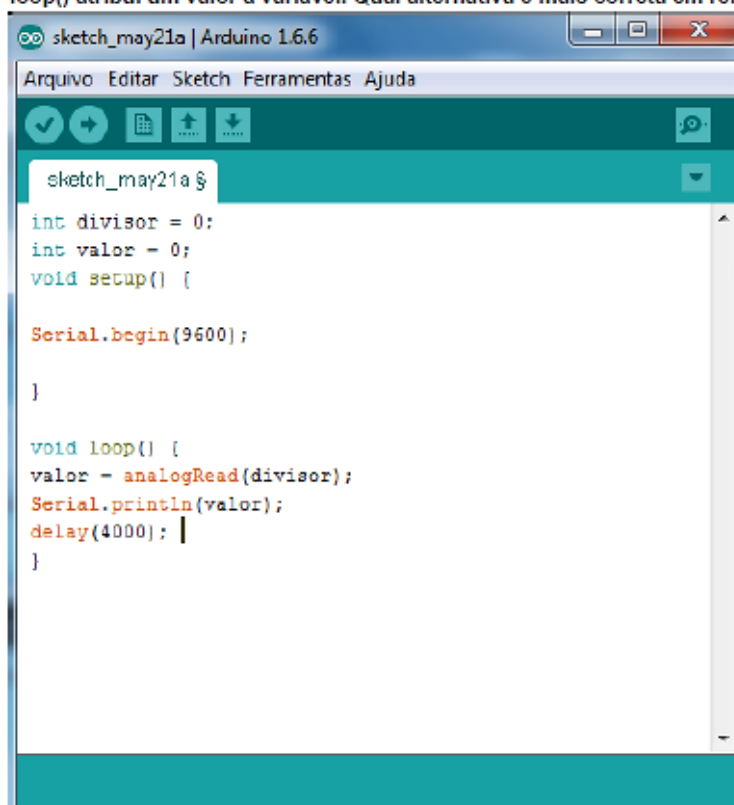
APÊNDICE F – Questionário 3 (online)

Responda as questões relacionadas ao que vimos nas três primeiras aulas

*Obrigatório

1. Endereço de e-mail *

2. Sabendo que este código não possui erro de sintaxe, responda: A linha abaixo do void loop() atribui um valor à variável. Qual alternativa é mais correta em relação a esta linha? *



```
sketch_may21a §
int divisor = 0;
int valor = 0;
void setup() {
  Serial.begin(9600);
}

void loop() {
  valor = analogRead(divisor);
  Serial.println(valor);
  delay(4000); |
}
```

Marcar apenas uma oval.

- a palavra divisor da linha tem armazenado o valor da leitura do pino analógico
- a variável valor sempre vale zero quando o loop() se repete
- trocando a variável divisor por A0 o comportamento do programa não se altera em nada
- valor é uma variável que pode valer 512,5 o que equivaleria 2,5 volts aproximadamente
- a variável valor vale a tensão (em volts) do pino A0

04/11/2017

Atividade 2

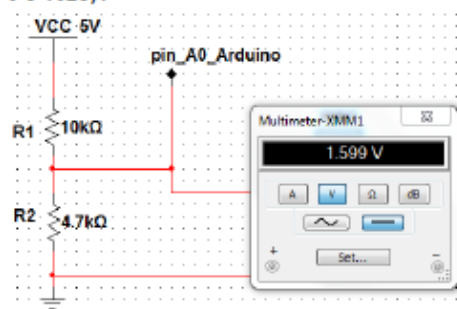
3. Usando a mesma imagem da questão 1 responda qual alternativa está mais correta em relação a linha: `Serial.println(valor);` *

Marcar apenas uma oval.

- a função desta linha é escrever na tela a palavra valor
- ela escreve a palavra valor na tela e troca de linha de 4 em 4 segundos
- o valor que aparece na tela é zerado depois dos 4 segundos, pois no início do loop() ela volta a ser zero
- o comportamento do programa é o mesmo caso colocássemos dentro dos parênteses a variável divisor
- posso escrever essa linha e todo o programa de outra forma sem utilizar nenhuma variável

4. Ao ligar um divisor de tensão a um pino analógico do Arduino o valor que eu recebo na tela é um número inteiro igual a 854. Qual a tensão em volts que esse número representa? *

5. A imagem abaixo é um divisor de tensão ligado ao pino A0 do Arduino e sendo medido por um voltímetro. Se eu "ler" esse pino qual o valor inteiro que ele valerá (entre 0 e 1023)? *



6. Ainda utilizando a imagem anterior, se eu trocar o resistor de 4,7k (R2) por um resistor de 20k, posso afirmar que: *

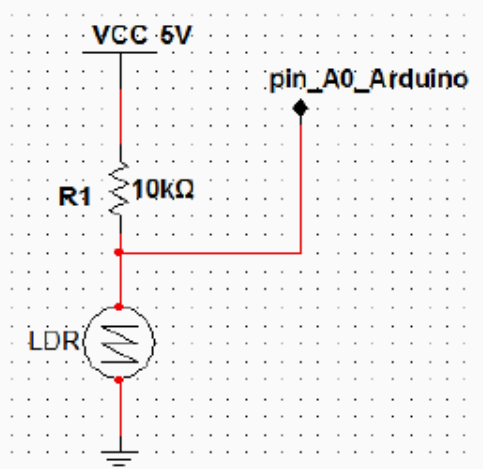
Marcar apenas uma oval.

- a tensão do voltímetro vai ser um valor bem mais baixo que 1,599V
- a tensão será 20 vezes mais baixa que a atual
- a tensão será 20 vezes mais alta que a atual
- a tensão será 2,5V, ou seja a metade dos 5V do Arduino
- a tensão valerá o 2/3 (dois terços) dos 5V do Arduino

04/11/2017

Atividade 2

7. No divisor de tensão abaixo temos um resistor de 10k (R1) e um LDR. Sabendo que o LDR diminui sua resistência quanto mais luminosidade incidir sobre ele, é correto afirmar que: *



Marcar apenas uma oval.

- iluminando o LDR a tensão do pino A0 aumenta
- trocando o R1 por outro LDR igual e com as mesmas características que o de baixo e iluminando igualmente os dois, o valor lido no pino A0 será aproximadamente 512
- trocando o R1 por outro LDR igual e com as mesmas características que o de baixo e iluminando igualmente os dois, o valor lido no pino A0 será aproximadamente 1023
- colocando dois LDRs iguais e de mesmas características e cobrindo o de cima com o dedo teremos uma tensão no pino A0 maior que 2,5V
- enquanto mais escuro estiver maior será a tensão no pino A0 e maior será a resistência de R1
8. Qual a principal diferença entre a entrada analógica e a entrada digital do Arduino. Quando devo usar uma ou outra? *

04/11/2017

Atividade 2

9. Faça um breve comentário de cada linhas de comando do código abaixo e explique o que o programa faz. *

```
int pinLed = 8;
int botao = 10;
int estadoBotao = 0;

void setup() {
  Serial.begin(9600);
  pinMode(pinLed, OUTPUT);
  pinMode(botao, INPUT);
}
void loop() {
  estadoBotao = digitalRead(botao);
  digitalWrite(pinLed, estadoBotao);
  Serial.print(estadoBotao);

}
```

APÊNDICE G – Código de programação para medir o painel solar

```

int R1 = 8;
int R2 = 9;
int R3 = 10;
int R4 = 11;
int R5 = 12;
int cont;
int tempo = 1; //tempo em min
const float R = 56.0;
float divisor, ajuste = 4.87;
float TensaoPlaca;
float Potencia;

void setup(){
  pinMode(R1, OUTPUT);
  pinMode(R2, OUTPUT);
  pinMode(R3, OUTPUT);
  pinMode(R4, OUTPUT);
  pinMode(R5, OUTPUT);
  Serial.begin(9600);
  Serial.print( "Tensao da Placa (V)      ");
  Serial.println( "Potencia Dissipada na Carga (W)");
}

void loop(){
  while(millis() < tempo*60000){ // o while() controla o tempo em que os relés são ligados
    pela variavel "tempo" (o 60000 transforma minutos em milisegundos)
    divisor = analogRead(A0)*0.0048; //divisor de tensao (resistores de 47k e 12k)
    TensaoPlaca = ajuste*divisor; //calculo para obter a tensao da placa
    Serial.println(TensaoPlaca);
    delay(1000);
  }
}

```

```

while(millis() < 2*tempo*60000) {
divisor = analogRead(A0)*0.0048;
TensaoPlaca = ajuste*divisor;
Serial.print(TensaoPlaca);
Serial.print("          ");
digitalWrite(R1, HIGH);          //liga o relé do resistor 1
Serial.print(TensaoPlaca*TensaoPlaca/R);
Serial.println("      R1");
delay(1000);
}

```

```

while(millis() < 3*tempo*60000) {
divisor = analogRead(A0)*0.0048;
TensaoPlaca = ajuste*divisor;
Serial.print(TensaoPlaca);
Serial.print("          ");
digitalWrite(R2, HIGH);          //liga o relé do resistor 2
Serial.print(TensaoPlaca*TensaoPlaca/(0.5*R));
Serial.println("      R2");
delay(1000);
}

```

```

while(millis() < 4*tempo*60000) {
divisor = analogRead(A0)*0.0048;
TensaoPlaca = ajuste*divisor;
Serial.print(TensaoPlaca);
Serial.print("          ");
digitalWrite(R3, HIGH);          //liga o relé do resistor 3
Serial.print(TensaoPlaca*TensaoPlaca/(0.33*R));
Serial.println("      R3");

```

```

delay(1000);

```

```

}

```

```

while (millis() < 5*tempo*60000) {

```

```

divisor = analogRead(A0)*0.0048;
TensaoPlaca = ajuste*divisor;
Serial.print(TensaoPlaca);
Serial.print("          ");
digitalWrite(R4, HIGH);          //liga o relé do resistor 4
Serial.print(TensaoPlaca*TensaoPlaca/(0.25*R));
Serial.println("      R4");

delay(1000);
}
while(millis() < 6*tempo*60000) {
divisor = analogRead(A0)*0.0048;
TensaoPlaca = ajuste*divisor;
Serial.print(TensaoPlaca);
Serial.print("          ");
digitalWrite(R5, HIGH);          //liga o relé do resistor 5
Serial.print(TensaoPlaca*TensaoPlaca/(0.2*R));
Serial.println("      R5");

delay(1000);
}
digitalWrite(R1, LOW); //desliga todos os relés
digitalWrite(R2, LOW);
digitalWrite(R3, LOW);
digitalWrite(R4, LOW);
digitalWrite(R5, LOW);
if (cont < 1){
Serial.println();
Serial.println("      DESLIGADO!!!");
Serial.println();
cont++;
}
}

```


ANEXO A – Pôster feito pelo PPE 9 e seus alunos

mostra de Ciências da
Escola CAICIrrigação Automatizada de uma Horta Utilizando
S4A e Placa Arduino

E.E.E.M. Professor Leopoldo Maieron – CAIC

Disciplina de Tecnologias Educacionais – turmas 80 e 81

Alunos: Lidiane Domingues Marques; João Victor Machado Leonel dos Santos; Joaquim Ribeiro Cardoso e Maria Eduarda Madruga Maia

Professor Orientador: Ricardo Nunes Rodrigues

Bolsistas: Bárbara Quartieri e Luciane Machado (PIBID da Física – UNIPAMPA)

Introdução

O presente trabalho é uma construção feita pelos alunos da disciplina de Tecnologias Educacionais utilizando os conhecimentos na linguagem de programação em blocos do *Scratch* (software que se utiliza de blocos lógicos para programar). O projeto foi desenvolvido utilizando o *Arduino* (placa microcontrolada que interage com o mundo físico através de sensores e atuadores) e o software *S4A* (*Scratch for Arduino* - um *Scratch* modificado que interage com a placa *Arduino* para a construção de projetos).

Objetivo

Executar a irrigação automática de uma horta controlada pelo *Arduino* (fig.1) através do código criado no *S4A* (fig. 2) e utilizando as medidas de umidade e intensidade da luz solar verificada pelos sensores (fig. 3a e 3b).



Fig. 1 – Placa Arduino

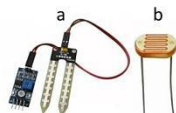
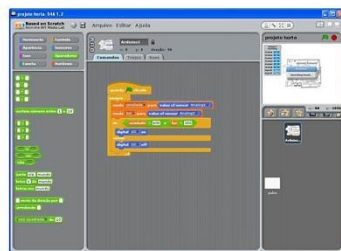
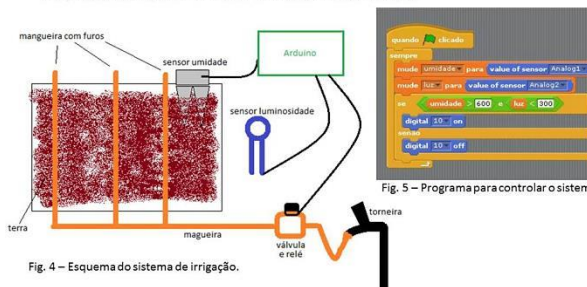
Fig. 3 – (a) Sensor de umidade do solo
(b) sensor de luminosidade.

Fig. 2 – Software S4A com a programação para funcionamento do Arduino.

Metodologia

Foram utilizados, além dos sensores e *Arduino*, outros materiais como: mangueira, conexões de pvc, fios, válvula solenoide, relé, resistores e protoboard.

O funcionamento do sistema ocorre pelo controle “liga/desliga” da válvula acionada pelo relé. O *Arduino* faz a leitura do sensor que fica em contato com a terra. Quando verifica um valor de umidade abaixo do especificado no programa ele aciona o relé e a válvula é aberta, liberando a água. Mas, para que a válvula seja aberta, há outra condição: a intensidade da luz solar medida pelo sensor deve estar abaixo da especificada, garantindo que a água não seja liberada com o sol muito intenso.



Conclusões

O sistema pode ser eficaz para evitar o desperdício de água em uma horta ou irrigação de uma planta. A resposta dos sensores se mantiveram com boa precisão. Como o sensor pode ser calibrado, há varias possibilidade que poderão ser exploradas em projetos futuros.



Fig. 6 – Alunos programando o S4A e testando o sensor

Agradecimentos: À UNIPAMPA pelo material utilizado no projeto e pela impressão do pôster.

ANEXO B – Termo de consentimento livre e esclarecido

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Dados de identificação

Título do Projeto: Curso de Arduino para Professores

Pesquisador Responsável: Januário Dias Ribeiro

Instituição: Universidade Federal do Pampa - UNIPAMPA

Telefones para contato: 53 32403600 (Ramal 9650) cel: 53 81440946 ou 53 91094960

Nome do voluntário: _____

R.G. _____

Januário Dias Ribeiro é aluno regularmente matriculado no Programa de Pós-Graduação em Ensino de Ciências. Este programa visa à diversificação e qualificação do ensino de ciências na Educação Básica, proporcionando a seus alunos contato com o uso de novas tecnologias e novas práticas pedagógicas. Visando cumprir com os requisitos do programa, o professor necessita aplicar, em sala de aula, uma metodologia inovadora. Estas metodologias não irão, de forma alguma, expor os participantes a situações desconfortáveis ou inseguras, assim como eventuais filmagens, gravações em áudio e fotografias serão utilizadas exclusivamente para a análise, por parte do pesquisador, da eficácia de sua proposta didática inovadora. Em casos de dúvidas, os voluntários poderão telefonar para o pesquisador responsável (5381440946) ou enviar mensagem eletrônica para o endereço: januarioribeiro@unipampa.edu.br

A participação dos alunos é voluntária e este consentimento poderá ser retirado a qualquer tempo, sem prejuízos à continuidade da pesquisa. As informações prestadas serão de caráter confidencial e a sua privacidade será garantida.

Eu, _____, RG nº _____
 declaro ter sido informado e concordo em participar, como voluntário, do projeto de pesquisa acima descrito.

(Local) _____, _____ de _____ de _____

 Nome do aluno

 Assinatura do Responsável