



UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

**Robótica Educativa em uma abordagem  
investigativa: uma proposta para o ensino de  
linguagem de programação**

Danielle Martins Tostes

Rio de Janeiro  
Agosto/2021



UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

# **Robótica Educativa em uma abordagem investigativa: uma proposta para o ensino de linguagem de programação**

Danielle Martins Tostes

Monografia apresentada ao Instituto de Física da  
Universidade Federal do Rio de Janeiro como parte  
dos requisitos necessários para conclusão do curso  
de Licenciatura em Física.

Orientador: Prof. Dr. Sidnei Percia da Penha  
(CAp-UFRJ)

Rio de Janeiro

Agosto/2021

## CIP - Catalogação na Publicação

M386r      Martins Tostes, Danielle  
Robótica Educativa em uma abordagem  
investigativa: uma proposta para o ensino de  
linguagem de programação / Danielle Martins Tostes.  
- Rio de Janeiro, 2021.  
79 f.

Orientador: Sidnei Percia da Penha.  
Trabalho de conclusão de curso (graduação) -  
Universidade Federal do Rio de Janeiro, Instituto  
de Física, Licenciado em Física, 2021.

1. Robótica Educativa. 2. Ensino por Investigação.  
3. Linguagem de Programação. I. Percia da Penha,  
Sidnei, orient. II. Título.

# Resumo

A tecnologia ocupa um lugar de destaque em nossa sociedade e, neste cenário, a robótica educativa surge como uma maneira de despertar nos estudantes o interesse pelo conhecimento científico e incentivar a criação de um ambiente de pesquisa colaborativa. Sua implementação em sala de aula, aliada a uma abordagem de ensino por investigação, tem muito a acrescentar ao programa de Ensino de Ciências. Uma ferramenta importante na robótica educativa é a linguagem de programação, necessária para automação dos protótipos através do uso de placas controladoras. O desafio encontrado é como ensinar essa parte técnica sem tirar o protagonismo do estudante no desenvolvimento da atividade. Nesse contexto está inserido este trabalho cujo objetivo é tentar aproximar a maneira como é feito o ensino de programação a uma abordagem de ensino por investigação. Ele está vinculado às atividades do Laboratório Didático de Ensino de Física (LaDEF) do CAP-UFRJ que, desde 2016, tem realizado as Oficinas de Acionamento e Robótica com estudantes do Ensino Médio e último ano do Ensino Fundamental. Partindo das atividades que já eram feitas anteriormente nas oficinas para ensino das estruturas de controle, foi proposta uma mudança na abordagem de modo que seja percebida a necessidade dessas estruturas para execução da tarefa proposta, antes que seja apresentada a sintaxe na linguagem de programação. Aqui se torna importante a noção de algoritmo. Esta atividade foi aplicada nas oficinas dos estudantes e no curso de extensão oferecido para professores da rede pública de ensino.

**Palavras-chave:** Robótica Educativa; Ensino por Investigação; Linguagem de Programação.

# Abstract

Technology plays an important role in our society and, in this scenario, educational robotics born to increase both: students' interest in scientific knowledge and the creation of a collaborative research environment. Its implementation in the classroom with an inquiry-based teaching approach, has a lot to add to the Science teaching program. An important tool in educational robotics is the programming language we use to build autonomous prototypes controlled by Arduino boards. The challenge faced is how to teach the technical concepts without preventing the students from being the main characters in the activity development. In this context, it is inserted this work which aims to try to approximate the way how programming language teaching is done to an approach of inquiry-based teaching. Since 2016, these robotics workshops have been offered to High School students in the city of Rio de Janeiro. Starting from the activities that had been previously used in the workshops to teach the control structures, a change in the approach was proposed to allow the students to realize the need to use these structures to solve the proposed task, before the introduction of the syntax in the programming language. Here it becomes important to introduce the concept of algorithm. This activity was applied to the student workshops and to the extension course offered to teachers from the Public School System.

**Keywords:** Educational Robotics, Inquiry-based teaching, Programming Language.

# Agradecimentos

Agradeço ao meu orientador, professor Sidnei Percia, por partilhar sua experiência em ensino, pela orientação, paciência, incentivo e amizade ao longo desses anos e por me deixar ser a “dona” da oficina 2.

Agradeço ao prof. Wanderley Jr. por todas as conversas e conselhos.

Agradeço ao restante da equipe das Oficinas de Robótica, Luciana, Rafael, Laura, Marcella, Lucas, Gabriel e Gleison, pela parceria nesse trabalho.

Agradeço aos professores Beto e Rodrigo do CAp-UFRJ, que tive a honra de acompanhar durante o estágio, por compartilharem sua experiência em sala de aula.

Agradeço os colegas que me fizeram companhia durante o estágio supervisionado, Rodrigo, Rodolfo, Dandara, Leandro, Bruno, Jessica, Nina, Leandro e Renã, pela convivência e troca de experiências.

Sou grata aos estudantes da oficina de Robótica do CAp por serem minhas “cobaias” na elaboração desse trabalho e por tudo que aprendi com eles.

Agradeço ao corpo docente e aos funcionários do IF-UFRJ por propiciarem um excelente ambiente de formação.

Agradeço a todos aqueles que direta ou indiretamente contribuíram na minha trajetória acadêmica: professores, orientadores e grupos de pesquisa dos quais tive o prazer de fazer parte.

Agradeço a minha amiga Ihani Junger por todas as vezes em que foi meu lembrete de escrita da monografia (e não foram poucas).

Não citarei nomes dessa vez, mas sou imensamente grata a TODOS os meus amigos pelo incentivo, aconselhamento, apoio, e por toda alegria proporcionada por poder contar com pessoas tão maravilhosas ao meu redor.

Agradeço a minha família por me amar e me apoiar sempre.

E, acima de tudo, agradeço a Deus por me permitir concluir mais essa etapa e a Nossa Senhora por sua intercessão e amparo.

# Lista de Figuras

3.1	Trecho da apostila utilizada pelos estudantes da oficina 2 mostrando um exemplo de como utilizar a função <i>while</i> . . . . .	29
5.1	Primeira versão do código escrito a partir do algoritmo elaborado pelos estudantes para a solução do problema proposto. . . . .	48
5.2	Versão final do código escrito a partir do algoritmo elaborado pelos estudantes para a solução do problema proposto, obtida após realizadas as devidas correções. . . . .	52



# Lista de Tabelas

2.1	Quatro classes de abordagem comunicativa (MORTIMER; SCOTT, 2002).	21
2.2	Intervenções do professor. Modificado de Mortimer & Scott (2002).	22
2.3	Indicadores da Alfabetização Científica. Modificado de Penha, Carvalho & Vianna (2015).	24

# Sumário

<b>1</b>	<b>Introdução</b>	<b>11</b>
<b>2</b>	<b>Referencial Teórico</b>	<b>13</b>
2.1	Alfabetização Científica . . . . .	14
2.2	Ensino por investigação . . . . .	16
2.3	Robótica Educativa . . . . .	19
2.4	Ferramentas para análise da atividade . . . . .	20
2.4.1	Análise das interações entre professor e estudantes . . . . .	20
2.4.2	Indicadores da Alfabetização Científica . . . . .	23
<b>3</b>	<b>Aspectos sobre o uso de programação em atividades de ensino</b>	<b>25</b>
3.1	Pensamento computacional e programação . . . . .	26
3.2	Algoritmo . . . . .	27
3.3	Contexto da atividade . . . . .	27
3.4	Abordagem proposta . . . . .	29
<b>4</b>	<b>Material didático de apoio ao estudante</b>	<b>32</b>
4.1	Introdução . . . . .	32
4.1.1	Algoritmo . . . . .	33
4.1.2	Exemplo . . . . .	33
4.2	Atividade . . . . .	34
4.2.1	Utilizando a linguagem de programação . . . . .	34
4.2.1.1	Estruturas de controle . . . . .	34

4.2.1.2	Realizando uma contagem . . . . .	35
4.2.2	Colocando em prática . . . . .	37
<b>5</b>	<b>Considerações sobre a aplicação da atividade nas oficinas de robótica</b>	<b>38</b>
5.1	Oficina com estudantes do Ensino Médio (2017) . . . . .	39
5.2	Comentários sobre outras aplicações . . . . .	52
5.2.1	Oficina remota com estudantes . . . . .	52
5.2.2	Oficina com professores . . . . .	53
<b>6</b>	<b>Considerações Finais</b>	<b>55</b>
	<b>Referências Bibliográficas</b>	<b>57</b>
	<b>Anexo 1 - Trecho da apostila da Oficina 1</b>	<b>60</b>
	<b>Anexo 2 - Trecho da apostila da Oficina 2</b>	<b>72</b>

# Capítulo 1

## Introdução

O uso da tecnologia, cada vez mais, se torna indispensável em nosso cotidiano e, nesse cenário, a robótica educativa aparece como uma forma de despertar nos estudantes o interesse pelo conhecimento científico e incentivar a criação de um ambiente de pesquisa colaborativa. Desta maneira, sua implementação em sala de aula, aliada a uma abordagem de ensino por investigação, tem muito a acrescentar ao programa de Ensino de Ciências.

O trabalho apresentado nesta monografia está vinculado às atividades do Laboratório Didático de Ensino de Física (LaDEF) do CAP-UFRJ que procura envolver os estudantes de Licenciatura em Física nas etapas de planejamento, desenvolvimento, implementação e avaliação de suas atividades. No desenvolvimento das atividades do projeto de Ensino, Pesquisa e Extensão intitulado *Oficinas de Acionamento e Robótica*, segue-se uma proposta de ensino por investigação que tem por objetivo trazer para o espaço social da sala de aula de ensino de ciências alguns dos aspectos da cultura científica. Neste projeto, iniciado em 2016, são realizadas oficinas onde os estudantes do Ensino Médio e último ano do Ensino Fundamental são desafiados a trabalharem colaborativamente no desenvolvimento de projetos que exigem a utilização de microprocessadores para gerenciamento de componentes e circuitos eletrônicos e eletromecânicos bem como para montagem de sua estrutura e controle. Para isso, eles devem aprender a manusear adequadamente essas ferramentas. Entre

outubro e dezembro de 2017, foi realizado também o curso de extensão para professores *Oficinas de Robótica: propostas de laboratório investigativo para o ensino de Física* com o intuito de estimular o uso dessas ferramentas em sala de aula e envolver ainda mais profissionais do ensino nas linhas de pesquisa desenvolvidas no LaDEF.

Na segunda etapa das oficinas, para desenvolvimento de projetos mais complexos, os estudantes passam a utilizar a placa Arduino. O manuseio desta placa controladora demanda um conhecimento técnico que deve ser repassado aos estudantes, incluindo a linguagem de programação necessária para comunicação com a mesma. Uma questão que surge é a de como ensinar a parte técnica sem tirar o protagonismo do estudante no desenvolvimento da atividade. Nesse contexto está inserida a atividade proposta neste trabalho. O objetivo é tentar aproximar a maneira como é feito o ensino de programação a uma abordagem de ensino por investigação. Partindo das atividades que já eram feitas anteriormente nas oficinas para ensino das estruturas de controle, propõe-se uma mudança na abordagem de modo que seja percebida a necessidade dessas estruturas para execução da tarefa proposta, antes que seja apresentada a sintaxe na linguagem de programação. Aqui se torna importante a noção de algoritmo.

Nesta monografia, as bases do ensino por investigação, referencial escolhido para o trabalho, e os indicadores que norteiam a nossa análise são apresentados no capítulo 2. No capítulo 3, são apresentados alguns aspectos relacionados a programação, a atividade proposta neste trabalho e o contexto em que está inserida. O capítulo 4 contém um texto de apoio que pode ser disponibilizado para o estudante. No capítulo 5, são feitas considerações baseadas no registro de vídeo das aplicações da atividade nas oficinas com os estudantes do CAP-UFRJ e no curso de extensão para professores. As considerações finais do trabalho são apresentadas no capítulo 6. Foram acrescentados como anexos, trechos das apostilas das oficinas 1 e 2 relacionados aos assuntos estudados anteriormente pelos estudantes e que são relevantes para a atividade.

## Capítulo 2

# Referencial Teórico

Ensinar é uma tarefa desafiadora e, ensinar ciências, um desafio ainda maior. A discussão sobre como ensinar ciências passa pela compreensão do que é ciência e o que se deseja atingir ao ensiná-la. Sabe-se que nem todos os estudantes se tornarão cientistas mas que a Alfabetização Científica é necessária a todos, mesmo aos que não serão profissionais da área.

Moreira & Osterman (1993) apontam em seu artigo algumas concepções errôneas encontradas na maneira como é ensinado o chamado método científico nas escolas: que este começa pela observação, é um procedimento lógico, algorítmico, rígido, e é indutivo; que a produção do conhecimento científico é cumulativa e linear, que o conhecimento científico é definitivo. Os autores criticam cada um desses pontos que estão longe de refletir a complexidade do fazer científico. Eles ressaltam que “se o conhecimento científico é produzido por indivíduos que pensam, sentem e fazem — como, de fato, o é — e se há uma constante interação entre pensar, sentir e fazer — como, de fato, há — então, o método científico tal como é ensinado, não existe” (MOREIRA; OSTERMAN, 1993).

Uma discussão similar é feita por Chalmers (1993) em seu livro “O que é a Ciência afinal?” onde associa essa visão do método científico ao que ele chama “indutivismo ingênuo”. Dentre as críticas que faz a essa concepção de ciência, questiona a validade e justificabilidade do princípio de indução e discute as suposições relativas ao *status* e ao papel da própria observação. Ele argumenta que “a ciência não começa

com proposições de observação porque algum tipo de teoria as precede” e que “as proposições de observação não constituem uma base firme na qual o conhecimento científico possa ser fundamentado porque são sujeitas a falhas” (CHALMERS, 1993).

Para Thomas Kuhn, uma teoria científica é uma estrutura complexa, que não se desenvolve por acúmulos sucessivos de novos conhecimentos. Ele percebeu que os relatos tradicionais da ciência não se adequavam ao testemunho histórico. Sua teoria dá ênfase ao caráter revolucionário do progresso científico e considera a importância das características históricas e sociológicas das comunidades científicas (CHALMERS, 1993).

Um conceito importante na teoria de Kuhn que ajuda a caracterizar os diferentes períodos no progresso da ciência é o de *paradigma*, que consiste num conjunto de suposições teóricas gerais e de leis e técnicas para a sua aplicação adotadas por uma comunidade científica. Quando num período de *ciência normal*, os cientistas trabalham para articular um paradigma com o objetivo de melhorar a correspondência entre ele e a natureza. Quando surge uma quantidade significativa de anomalias sérias e há dificuldade de explicar os fenômenos da natureza pelo paradigma vigente é constituído um período de *crise*, levando a necessidade de criação de um novo paradigma. Uma *revolução científica* corresponde ao abandono de um paradigma e adoção de um novo pela comunidade científica (CHALMERS, 1993).

É necessário, então, considerar alguns dos aspectos relacionados a Natureza da Ciência ao desenvolver atividades e estratégias de ensino.

## 2.1 Alfabetização Científica

Os termos “Letramento Científico”, “Enculturação Científica” e “Alfabetização Científica” são utilizados para se referir ao ensino de ciências que almeja a formação cidadã dos estudantes para o domínio e uso dos conhecimentos científicos e seus desdobramentos nas mais diferentes esferas de sua vida (SASSERON; CARVALHO, 2011).

A escolha do termo a ser adotado carrega consigo os motivos que guiam o planejamento do ensino de ciências. Ao utilizar o termo “enculturação” se reconhece a existência de uma cultura científica com noções, ideias, valores, linguagem e conceitos próprios na qual a participação deve ser promovida.

Ao adotar a expressão “Alfabetização Científica”, Sasseron & Carvalho (2011) se baseiam na ideia de alfabetização concebida por Paulo Freire:

“...a alfabetização é mais que o simples domínio psicológico e mecânico de técnicas de escrever e de ler. É o domínio destas técnicas em termos conscientes. (...) Implica numa autoformação de que possa resultar uma postura interferente do homem sobre seu contexto.” (FREIRE, 1980 apud SASSERON; CARVALHO, 2011)

Entendendo que a alfabetização deve formar pessoas capazes de organizar seu pensamento de maneira lógica, e auxiliar na construção de uma consciência mais crítica em relação ao mundo que as cerca. Esta é a expressão que será adotada ao longo desta monografia.

A sociedade depende dos conhecimentos que a ciência constrói, então, é preciso que ela saiba mais sobre isso. Considerar somente os trabalhos sem suas aplicações dificulta o entendimento.

Sasseron & Carvalho (2008) definiram três **Eixos Estruturantes da Alfabetização Científica** capazes de fornecer bases para o planejamento de atividades que visam a alfabetização científica. São eles: *a compreensão básica dos termos, conhecimentos e conceitos científicos fundamentais*, relacionado à importância de se compreender conceitos-chave para poder entender situações do dia-a-dia, *a compreensão da natureza das ciências e dos fatores éticos e políticos que circundam a sua prática*, considerando as reflexões e análise necessárias em determinadas circunstâncias, e *o entendimento das relações existentes entre ciência, tecnologia, sociedade e meio-ambiente*, reconhecendo que os fatos da vida são influenciados pelas ciências e tecnologias.

Santos (2007) aponta três aspectos considerados nos estudos sobre as funções da alfabetização científica: *natureza da ciência*, que implica conhecimentos sobre



história, filosofia e sociologia da ciência, *linguagem científica*, que possui características próprias que a distingue da linguagem cotidiana, e *aspectos sociocientíficos*, que referem-se às questões ambientais, políticas, econômicas, éticas, sociais e culturais relativas à ciência e tecnologia.

Penha (2012) destaca que apesar de adotarem classificações ligeiramente diferentes, Sasseron & Carvalho (2008) e Santos (2007) contemplaram nestas classificações a identificação de atributos semelhantes.

Nesse contexto, Gil-Pérez & Vilches-Peña (2001 apud SASSERON; CARVALHO, 2011) propõem o ensino por investigação como uma excelente maneira de favorecer a alfabetização científica.

## 2.2 Ensino por investigação

Rodrigues & Borges (2008) apresentam em seu trabalho uma reconstrução histórica da metodologia de ensino por investigação com o objetivo de “demonstrar que a ideia de ensinar como investigação sofreu diversas modificações” (RODRIGUES; BORGES, 2008).

Nesta seção, pretende-se apresentar as bases do ensino por investigação e as principais características desta metodologia.

Carvalho (2013) destaca dois fatores que vieram modificar a maneira como era feita a transmissão do conhecimento: o aumento exponencial do conhecimento produzido e os trabalhos de epistemólogos e psicólogos sobre a construção dos conhecimentos em nível individual e social.

Para os psicólogos interacionistas, as crianças procuram ativamente compreender o que vivenciam e explicar o que lhes é estranho, construindo hipóteses que lhes pareçam razoáveis. Os seus conhecimentos são construídos por meio de sua interação com o meio; nessa interação, fatores internos e externos se interrelacionam

continuamente e experiências anteriores servem de base para novas construções (DAVIS; OLIVEIRA, 1994). Dentro da corrente interacionista, destacam-se as teorias de Piaget e Vygotski.

Para Piaget, o indivíduo procura manter um estado de equilíbrio ou de adaptação com o seu meio, agindo de forma a superar perturbações na relação que estabelece com ele. Desta maneira, o desenvolvimento cognitivo do indivíduo ocorre através de constantes *desequilíbrios* e *equilibrações*. Ao se deparar com uma nova possibilidade orgânica no indivíduo ou a mudança de alguma característica do ambiente, um desequilíbrio é criado. Dois mecanismos “ocorrem” ao mesmo tempo para atingir o novo estado: *assimilação*, que atribui significações aos elementos do ambiente sem alterar estruturas, a partir de sua experiência anterior; e *acomodação*, no qual se modifica para se ajustar à demanda do ambiente (DAVIS; OLIVEIRA, 1994).

Na teoria de Vygotski, defende-se a ideia de que o pensamento é construído num ambiente, em essência, social. Possui especial importância as possibilidades que o indivíduo dispõe a partir do acesso que tem a “instrumentos” físicos (como objetos) e simbólicos (como a cultura, valores, crenças, costumes, tradições, conhecimentos) (DAVIS; OLIVEIRA, 1994).

Segundo Vygotski, a referência do indivíduo com parceiros mais experientes cria uma *zona de desenvolvimento potencial*<sup>1</sup>, que se refere à distância entre o desenvolvimento atual do indivíduo, relacionado a sua capacidade de resolver problemas sem ajuda, e seu potencial de desenvolvimento, relacionado a sua capacidade de solucionar problemas com orientação ou em colaboração (DAVIS; OLIVEIRA, 1994).

Dentro da perspectiva do ensino por investigação, as teorias de Piaget e Vygotski se complementam e servem como fundamentação para sua proposta (CARVALHO, 2013).

Munford & Lima (2007) apresentam em seu trabalho algumas considerações sobre o ensino de ciências por investigação. Dentre elas, o que consta no documento

---

<sup>1</sup>Também traduzida como *zona de desenvolvimento proximal*.

“Investigação e os Parâmetros Curriculares Nacionais de Ciências: Um Guia para Ensino e Aprendizagem” (*Inquiry and the National Science Education Standards: A Guide for Teaching and Learning*), elaborado em 2000, que “propõe como essencial ao ensino de ciências que os aprendizes: engajem-se com perguntas de orientação científica; dêem prioridade às evidências ao responder questões; formulem explicações a partir de evidências; avaliem suas explicações à luz de outras alternativas, em particular as que refletem o conhecimento científico; comuniquem e justifiquem explicações propostas” (MUNFORD; LIMA, 2007).

Borges (2002), ao discutir o papel do laboratório escolar de ciências, aponta que “os processos educacionais devem respeitar e favorecer a atividade do estudante, e que esta deve ser o centro do processo de aprendizagem”, destaca a necessidade de haver planejamento e clareza dos objetivos das atividades propostas para que estas sejam eficazes, e propõe que as atividades de laboratórios devem ser estruturadas como investigações ou problemas práticos mais abertos (BORGES, 2002).

Capecchi & Carvalho (2006) reconhecem que existem diferenças entre a forma que o conhecimento científico é produzido e transmitido, mas indicam, ao analisar uma atividade de laboratório aberto aplicada, que é possível observar a utilização de ferramentas importantes na construção de uma explicação científica, como o problema, as hipóteses e a realização de teste empírico como forma de contrastá-los com a realidade, identificando episódios de enculturação científica.

As chamadas *Sequências de Ensino Investigativas* (SEI) são propostas por Carvalho (2013) e visam proporcionar aos alunos: condições de trazer seus conhecimentos prévios para iniciarem os novos, terem ideias próprias e poder discuti-las com seus colegas e com o professor passando do conhecimento espontâneo ao científico e tendo condições de entenderem conhecimentos já estruturados por gerações anteriores. As SEIs possuem uma estrutura que inclui: *problema contextualizado* (experimental ou teórico), *sistematização do conhecimento construído* e *contextualização do conhecimento no dia a dia* (CARVALHO, 2013).

Estes trabalhos e reflexões permitem caracterizar a metodologia de ensino por investigação e foram considerados na elaboração da atividade proposta nesta monografia.

## 2.3 Robótica Educativa

Vivemos numa sociedade cada vez mais tecnológica e, nesse cenário, é natural que diferentes tecnologias sejam inseridas na sala de aula e que se tornem ferramentas a serem exploradas no contexto da educação.

A *robótica educativa*, também conhecida como *robótica educacional* ou *robótica pedagógica*, já vem sendo reconhecida como uma excelente ferramenta de ensino-aprendizagem e tem sido aplicada em diversas áreas como, por exemplo, matemática (MALIUK, 2009) e engenharia de produção (VARGAS *et al.*, 2012), e também como ferramenta para promover a inclusão (LOPES *et al.*, 2015) e incentivo a carreiras tecnológicas (VALLIM *et al.*, 2009).

Segundo D’Abreu *et al.* (2012), a robótica educativa “se torna mais efetiva e atraente porque além de se ampliar as possibilidades de recursos digitais associa-se a este processo o design, concepção, construção, e o controle via computador de dispositivos que os próprios alunos podem desenvolver e compartilhar com seus colegas via rede”.

Desta maneira, ela aparece como uma forma de despertar nos estudantes o interesse pelo conhecimento científico e incentivar a criação de um ambiente de pesquisa colaborativa e sua implementação em sala de aula, aliada a uma abordagem de ensino por investigação, tem muito a acrescentar ao programa de Ensino de Ciências.

Uma questão que dificulta a implementação da Robótica Educativa, principalmente nas instituições públicas de Ensino, é o elevado custo dos materiais utilizados na elaboração dos protótipos. A busca por materiais alternativos e confecção própria de peças ajuda a viabilizar a realização de oficinas de robótica (REZENDE *et al.*, 2019b).

Para propiciar o uso da Robótica Educativa, é necessário que haja um empenho para formar professores e profissionais de ensino capacitados para realização das oficinas. Com o objetivo de colaborar nesse esforço e capacitar professores para utilizar a Robótica Educativa em suas salas de aula, o LaDEF (CAP-UFRJ) realizou um curso de extensão destinado aos professores da rede pública do Rio de Janeiro<sup>2</sup>.

## 2.4 Ferramentas para análise da atividade

Para análise dos dados da pesquisa, serão utilizadas algumas das ferramentas apresentadas por Mortimer & Scott (2002), voltadas para análise das interações, e por Sasseron & Carvalho (2008) e Penha, Carvalho & Vianna (2015), que introduzem os chamados indicadores da alfabetização científica. Essas ferramentas serão brevemente discutidas nesta seção.

### 2.4.1 Análise das interações entre professor e estudantes

O trabalho de Mortimer & Scott (2002) introduz uma ferramenta para analisar as formas de interação entre professores e estudantes que resultam na construção de significados em salas de aula de ciências. A estrutura analítica que apresentam é baseada em cinco aspectos interrelacionados: *intenções do professor*, *conteúdo*, *abordagem comunicativa*, *padrões de interação* e *intervenções do professor*. Estes aspectos focalizam o papel do professor e podem ser agrupados em termos de focos de ensino, abordagem e ações. Os dois aspectos mais relevantes para a análise apresentada nessa monografia são abordagem comunicativa e intervenções do professor.

A *abordagem comunicativa* fornece uma perspectiva sobre como são trabalhados as intenções e o conteúdo de ensino através das diferentes intervenções pedagógicas resultando em diferentes padrões de interação (MORTIMER; SCOTT, 2002). Quatro classes de abordagem comunicativas podem ser identificadas e são apresentadas na tabela 2.1.

---

<sup>2</sup>O curso foi realizado no final de 2017 e seus resultados foram apresentados no SNEF 2019 (REZENDE *et al.*, 2019a).

	INTERATIVO	NÃO-INTERATIVO
DIALÓGICO	Interativo/dialógico	Não-interativo/dialógico
DE AUTORIDADE	Interativo/de autoridade	Não-interativo/de autoridade

Tabela 2.1: Quatro classes de abordagem comunicativa (MORTIMER; SCOTT, 2002).

Cada uma das classes é melhor explicada no trecho a seguir:

**a. Interativo/dialógico:** professor e estudantes exploram ideias, formulam perguntas autênticas e oferecem, consideram e trabalham diferentes pontos de vista.

**b. Não-interativo/dialógico:** professor reconsidera, na sua fala, vários pontos de vista, destacando similaridades e diferenças.

**c. Interativo/de autoridade:** professor geralmente conduz os estudantes por meio de uma sequência de perguntas e respostas, com o objetivo de chegar a um ponto de vista específico.

**d. Não-interativo/de autoridade:** professor apresenta um ponto de vista específico.

(MORTIMER; SCOTT, 2002)

Essas classes se aplicam não somente ao papel do professor ao elaborar sua fala em sala de aula mas também às interações entre os estudantes.

Com relação às *intervenções do professor*, de acordo com o trabalho de Scott (1998 apud MORTIMER; SCOTT, 2002), é possível identificar seis formas de intervenção pedagógica. Estas formas são apresentadas na Tabela 2.2, onde são especificados também o foco e as ações do professor relacionados a cada uma delas.

<b>Intervenção do Professor</b>	<b>Foco</b>	<b>Ação - o professor:</b>
<b>Dando forma aos significados</b>	Explorar as ideias dos estudantes / Trabalhar os significados no desenvolvimento da estória científica.	introduz um termo novo; parafrasea uma resposta do estudante; mostra a diferença entre dois significados.
<b>Selecionando significados</b>		considera a resposta do estudante na sua fala; ignora a resposta de um estudante.
<b>Marcando significados chave</b>		repete um enunciado; pede ao estudantes que repita um enunciado; estabelece uma sequência I-R-A (Iniciação-Resposta-Avaliação) com um estudante para confirmar uma ideia; usa um tom de voz particular para realçar certas partes do enunciado.
<b>Compartilhando significados</b>	Tornar os significados disponíveis para todos os estudantes da classe	repete a ideia de um estudante para toda a classe; pede a um estudante que repita um enunciado para a classe; compartilha resultados dos diferentes grupos com toda a classe; pede aos estudantes que organizem suas ideias ou dados de experimentos para relatarem para toda a classe.
<b>Checando o entendimento dos estudantes</b>	Verificar que significados os estudantes estão atribuindo em situações específicas	pede a um estudante que explique melhor sua ideia; solicita aos estudantes que escrevam suas explicações; verifica se há consenso da classe sobre determinados significados.
<b>Revedo o progresso da estória científica</b>	Recapitular e antecipar significados	sintetiza os resultados de um experimentos particular; recapitula as atividades de uma aula anterior; revê o progresso no desenvolvimento da estória científica até então.

Tabela 2.2: Intervenções do professor. Modificado de Mortimer &amp; Scott (2002).

## 2.4.2 Indicadores da Alfabetização Científica

O trabalho de Sasseron & Carvalho (2008) introduz os chamados indicadores da alfabetização científica como “competências comuns desenvolvidas e utilizadas para resolução, discussão e divulgação de problemas em quaisquer das Ciências quando se dá a busca por relações entre o que se vê do problema investigado e as construções mentais que levam ao entendimento dele.” Eles têm como função mostrar algumas das destrezas que devem ser trabalhadas quando se deseja colocar a Alfabetização Científica em processo de construção entre os estudantes (SASSERON; CARVALHO, 2008).

Estes indicadores podem ser organizados em três grupos de acordo com as diferentes dimensões a que estão relacionados: *trabalho com os dados obtidos numa investigação, estruturação do pensamento e procura do entendimento da situação analisada*. Os indicadores de Alfabetização Científica estão apresentados na tabela 2.3, extraída do trabalho de Penha, Carvalho & Vianna (2015).



<b>Indicadores da Alfabetização Científica</b>		
<b>Tratamento dos dados de uma investigação</b>	<i>seriação de informações</i>	Indicador que não necessariamente prevê uma ordem a ser estabelecida, mas pode ser um rol de dados, uma lista de dados trabalhados.
	<i>organização de informações</i>	ocorre nos momentos em que se discute sobre o modo como um trabalho foi realizado.
	<i>classificação de informações</i>	ocorre quando se busca conferir hierarquia às informações obtidas.
<b>Estruturação do pensamento</b>	<i>raciocínio lógico</i>	compreende o modo como as ideias são desenvolvidas e apresentadas e está diretamente relacionada à forma como o pensamento é exposto.
	<i>raciocínio proporcional</i>	mostra como se estrutura o pensamento, e refere-se também à maneira como variáveis têm relações entre si, ilustrando a interdependência que pode existir entre elas.
<b>Entendimento da situação analisada</b>	<i>levantamento de hipóteses</i>	aponta instantes em que são alçadas suposições acerca de certo tema (pode surgir da forma de uma afirmação ou de uma pergunta).
	<i>teste de hipóteses</i>	colocar à prova as suposições anteriormente levantadas (pode ocorrer tanto diante da manipulação direta de objetos quanto no nível das ideias).
	<i>justificativa</i>	quando em uma afirmação qualquer proferida lança mão de uma garantia para o que é proposto.
	<i>previsão</i>	é explicitado quando se afirma uma ação e/ou fenômeno que sucede associado a certos acontecimentos
	<i>explicação</i>	quando se busca relacionar informações e hipóteses já levantadas. (Estão relacionadas à justificativa para o problema).

Tabela 2.3: Indicadores da Alfabetização Científica. Modificado de Penha, Carvalho & Vianna (2015).

## Capítulo 3

# Aspectos sobre o uso de programação em atividades de ensino

Ao se trabalhar com robótica educativa, uma série de conhecimentos técnicos são necessários para realização dos projetos, dentre eles, a programação, que exerce um papel fundamental na automação dos protótipos através do uso de placas controladoras.

Na descrição prévia dos referenciais, foi apontada a importância de favorecer a atividade do estudante que deve planejar e executar as ações necessárias para o desenvolvimento da investigação. Nesse sentido, ele necessita apropriar-se de uma nova linguagem formada por um conjunto de novos símbolos, com formato, propriedades e significados específicos. Muitas propostas didáticas abordam estas características como se fossem um manual técnico que desconsidera o protagonismo do estudante para atuar nestas atividades. A hipótese de que é possível estabelecer condições mínimas para autonomia dos estudantes em seus trabalhos de investigação, mesmo em situações que exigem conhecimento técnico (PENHA, 2019), é o fator motivador na busca por atividades que propiciem essa autonomia.

Neste capítulo, são apresentadas algumas características do pensamento computacional e da programação e os benefícios relacionados a eles, o conceito de algoritmo,

o contexto no qual está inserida a atividade proposta neste trabalho, a abordagem utilizada e mais detalhes sobre o seu desenvolvimento e aplicação em sala de aula.

### 3.1 Pensamento computacional e programação

Uma das vantagens do ensino de programação e algoritmos é promover o pensamento computacional (OLIVEIRA, 2013).

*Pensamento computacional* é integrar o poder do pensamento humano com as capacidades dos computadores. Sua essência é pensar sobre dados e ideias, e usar e combinar esses recursos para resolver problemas (PHILLIPS, 2009). De acordo com Wing (2006), pensamento computacional é reformular um problema aparentemente difícil em um que saibamos resolver, talvez por redução, incorporação, transformação ou simulação, usar abstração e decomposição quando lidando com uma tarefa complexa e pensar em termos de prevenção, proteção e recuperação de cenários complicados. Envolve planejamento, aprendizado e planejamento em meio a incertezas.

Gomes, Henriques & Mendes (2008) apontam que a *programação* é muito mais do que a escrita de um conjunto de linhas de código numa dada linguagem, pois envolve criatividade, apresenta um conjunto de regras orientadoras e utiliza a lógica.

Sebesta (2006) apresenta como um dos benefícios de aprender os conceitos de linguagem de programação o aumento da capacidade de expressar ideias. A programação também promove a interdisciplinaridade pois estudantes que aprendem a programar, além de adquirirem domínio de um instrumento tecnológico, entram em contato com conteúdos programáticos de diferentes disciplinas, tais como: a ciência, a matemática e a arte da construção de modelos intelectuais (PAPERT, 1980 apud OLIVEIRA, 2013).

Um programa escrito para resolução de um problema com um computador exige pelo menos os seguintes passos: 1) definição ou análise do problema; 2) projeto do

algoritmo; 3) transformação do algoritmo em um programa; 4) execução e validação do programa (AGUILAR, 2008).

Essas etapas serão importantes para discussão da atividade proposta mas, antes de apresentá-la, é necessário introduzir o conceito de algoritmo.

## 3.2 Algoritmo

A palavra *algoritmo* deriva da tradução do latim da palavra *Alkhôwarizmi*, nome de um matemático e astrônomo árabe que escreveu um tratado sobre manipulação de números e equações no século IX (AGUILAR, 2008). Segundo Martins & Cravo (2011 apud OLIVEIRA, 2013), um algoritmo é uma sequência de instruções que podem ser executadas de modo a atingir um determinado objetivo, associado à solução de um problema. Ele deve ser *preciso*, *definido* e *finito* (AGUILAR, 2008).

O conceito de algoritmo é fundamental em informática, considerando que os computadores só fazem aquilo que foi mandado, e não necessariamente o que se deseja que façam (MARTINS; CRAVO, 2011 apud OLIVEIRA, 2013). Existem três formas de descrever algoritmos: através da *descrição narrativa*, método de expressar o passo a passo verbalmente, de *pseudolinguagens*, que consiste na escrita de instruções através de palavras semelhantes em inglês ou português, e de *linguagens gráficas*, que permitem uma representação gráfica de um algoritmo (e.g., diagrama de fluxo) (MARTINS; CRAVO, 2011 apud OLIVEIRA, 2013).

O conceito de algoritmo também pode ser relacionado a situações cotidianas. Uma receita de bolo pode ser considerada um exemplo de algoritmo. A introdução deste conceito é essencial para a abordagem proposta.

## 3.3 Contexto da atividade

A atividade proposta está inserida no projeto *Oficinas de Acionamento e Robótica* realizado no CAP-UFRJ, desde 2016, e que oferece oficinas para estudantes do Ensino Médio e do último ano do Ensino Fundamental. Na segunda etapa das oficinas,

chamada oficina 2, é introduzida a placa controladora Arduino que é utilizada na automação dos protótipos. Para o correto manuseio da placa, é necessário que sejam adquiridos conceitos básicos de programação.

Tendo chegado na oficina 2, os estudantes já passaram pela oficina 1 onde aprenderam noções de eletricidade e o manuseio de alguns componentes mecânicos e eletrônicos. Foi incluída como Anexo 1 a parte da apostila da oficina 1 referente ao conteúdo que terá relevância para atividade proposta. É também suposto que os estudantes já foram apresentados, em aulas anteriores, à placa Arduino e a suas funções básicas. Este conteúdo, da forma como é apresentado na apostila da oficina 2, constitui o Anexo 2.

As *estruturas de controle* exercem um papel fundamental na linguagem de programação. Elas são métodos de especificar a ordem em que instruções de um algoritmo são executadas. As estruturas de controle básico são: *sequência*, *seleção* e *repetição* (AGUILAR, 2008). A atividade proposta está centrada no ensino dessas estruturas. Anteriormente nas oficinas, elas eram ensinadas da seguinte maneira: era apresentada a estrutura e sua sintaxe na linguagem de programação, explicando a sua função e a maneira correta de escrevê-la, e um código-exemplo era disponibilizado para que o estudante pudesse vê-la em uso. Um exemplo utilizado após a explicação sobre a função *while*, que é uma estrutura de repetição, é mostrado na Figura 3.1.

Até esse momento, o estudante atua de forma passiva sendo somente exposto a esse novo conhecimento. A questão que surge é “*seria possível aumentar a participação do estudante e dar-lhe uma maior autonomia nessa atividade?*”. Propõe-se, então, uma mudança na abordagem utilizada de modo a propiciar uma maior participação do estudante já na apresentação das estruturas.

Para exemplificar o uso da função `while`, veja a estrutura do programa abaixo que faz o LED piscar três vezes, depois esperar cinco segundos, piscar mais três vezes e assim por diante.

```
// Variável para contar o número de vezes que o LED piscou
int i = 0;

// Pisca o LED três vezes
while(i < 3) {
    digitalWrite(led, HIGH); // Atribui nível lógico alto ao pino do LED, acendendo-o
    delay(1000);              // Espera 1000 milissegundos (um segundo)
    digitalWrite(led, LOW);  // Atribui nível lógico baixo ao pino do LED, apagando-o
    delay(1000);              // Espera 1000 milissegundos (um segundo)
    i = i + 1;                // Aumenta o número de vezes que o LED piscou
}

delay(5000);                 // Espera 5 segundos para piscar o LED de novo
```

Figura 3.1: Trecho da apostila utilizada pelos estudantes da oficina 2 mostrando um exemplo de como utilizar a função `while`.

### 3.4 Abordagem proposta

Partindo das atividades que já eram feitas anteriormente nas oficinas para ensino das estruturas de controle, é proposta uma mudança na abordagem de modo que os estudantes percebam a necessidade dessas estruturas para execução da tarefa, antes de ser apresentada a sintaxe na linguagem de programação. A primeira etapa nessa nova abordagem é introduzir a noção de algoritmo, que pode ser exemplificada através de situações cotidianas.

Definido o conceito, para ajudar a entender melhor as características do algoritmo, os estudantes são convidados a criar um algoritmo para uma atividade simples como, por exemplo, abrir a porta. Isto é feito coletivamente usando a descrição narrativa e, ao mesmo tempo, o professor (ou aplicador da atividade) vai seguindo as etapas propostas pelos estudantes de modo que possam perceber se as instruções foram claras e detalhadas, e possam fazer as devidas correções. Este exercício os ajudará a entender que as instruções devem ser objetivas e precisas para que a tarefa

seja realizada corretamente.

Depois de apresentada aos estudantes a definição de algoritmo, reforçada por alguns exemplos e por este primeiro exercício, é possível propor que criem um algoritmo para executar uma determinada tarefa. É importante ressaltar que, nesse momento, o estudante não precisa ter nenhum conhecimento prévio sobre programação. É necessário somente pensar no procedimento detalhado (algoritmo) para solução do problema proposto, o que pode ser feito através da descrição de ações simples como, por exemplo, ligar, desligar, acender, apagar, esperar, etc. Quando, na descrição das etapas, o estudante perceber que há a necessidade de realizar uma repetição ou testar uma condição, ele poderá ser apresentado ao *comando* na linguagem de programação que o permitirá executar essa tarefa. Dessa maneira, é mais fácil que assimile a nova linguagem introduzida.

Considerando os passos da programação descritos da seção 3.1, somente na etapa 3, transformação de um algoritmo em programa, é que se torna essencial a intervenção do professor uma vez que o estudante ainda não domina a linguagem de programação. De posse do algoritmo construído, o professor auxiliará os estudantes na elaboração do código correspondente na linguagem de programação, que pode ser implementado e testado com o protótipo robótico ou circuito eletrônico. Caso o teste não produza o resultado esperado, o estudante pode reavaliar o seu algoritmo, identificar a possível causa e propor soluções.

A estrutura da atividade proposta pode ser resumida nas seguintes etapas:

1. Apresentação do conceito de algoritmo;
2. Exemplos e exercício de fixação;
3. Tarefa em grupo (ou com toda a turma) para elaboração de algoritmo;
4. Apresentação das estruturas na linguagem C++ necessárias para implementação;
5. Montagem do circuito e execução do programa.

Procura-se, ao longo de toda a atividade, seguir uma abordagem comunicativa predominantemente interativa/dialógica (seção 2.4.1), no entanto, em alguns momentos utiliza-se, por exemplo, a de autoridade, principalmente quando é necessária a apresentação de um novo conceito. Isso será melhor explorado no capítulo 5.

Na aplicação realizada nas oficinas do CAp-UFRJ, optou-se por utilizar como tarefa o exemplo que era proposto anteriormente, mostrado na Figura 3.1, mas, na nova abordagem, o código não será apresentado. O problema proposto aos estudantes é elaborar um algoritmo para que o LED pisque 3 vezes, espere 5 segundos e volte a piscar.

A atividade seguindo a abordagem aqui proposta foi aplicada na oficina dos estudantes no CAp e no curso de extensão oferecido para professores da rede pública, realizados presencialmente em 2017, e na oficina remota oferecida aos estudantes em 2020. Algumas adaptações foram necessárias para aplicação na modalidade remota.



## Capítulo 4

# Material didático de apoio ao estudante

Este capítulo apresenta um texto que pode ser disponibilizado para o estudante com o objetivo de auxiliar na aplicação da atividade ou servir para consulta posterior. Ele não foi planejado para ser um material independente, mas é parte da atividade aplicada. Deste modo, é importante respeitar o andamento da aula e as intervenções dos estudantes na condução da atividade, que propõe-se que seja feita usando uma abordagem comunicativa predominantemente interativa/dialógica. Vale ressaltar que essa atividade está inserida no contexto apresentado na seção 3.3 e que serão utilizados alguns conteúdos estudados em aulas anteriores. O código apresentado como exemplo ao final do texto é apenas uma das maneiras de se solucionar a tarefa proposta.

### 4.1 Introdução

No último encontro, você foi apresentado à placa Arduino, viu algumas de suas funções e aprendeu que a comunicação com a placa é feita utilizando a linguagem de programação C++. Então, para controlar o Arduino precisamos aprender mais sobre essa linguagem mas, além disso, precisamos saber claramente o que queremos que ele faça. Um recurso que pode nos ajudar com isso é o algoritmo.

### 4.1.1 Algoritmo

O *algoritmo* pode ser definido simplesmente como uma *sequência finita de instruções bem definidas utilizadas para executar uma tarefa*, é um passo a passo que deve ser seguido. Isso é algo que muitas vezes utilizamos inconscientemente no nosso dia a dia. Um exemplo que podemos considerar é uma receita culinária. Seguindo uma receita de bolo, conseguimos fazer um bolo.

### 4.1.2 Exemplo

*Para compreender melhor essa ideia, converse com seus colegas para juntos elaborarem uma lista de ações que seriam necessárias para uma pessoa abrir uma porta.*

Você percebeu como teve que considerar algumas situações em que não tinha pensado inicialmente? Apesar de ser uma ação que realizamos frequentemente, a fazemos de maneira natural sem pensarmos em todas as etapas envolvidas. Quando programamos, precisamos considerar todas as etapas necessárias para podermos “ensinar” ao Arduino o que queremos que ele faça.

Vamos ver um outro exemplo para ajudar a fixar a ideia<sup>1</sup>.

*Imagine o trabalho de um recepcionista de cinema, ele deve conferir os bilhetes e direcionar o cliente para a sala correta. Além disso, se o cliente estiver 30 minutos atrasado o recepcionista deve informar que a entrada não é mais permitida. E quando o cliente estiver 30 minutos adiantado o recepcionista deve informar que a sala do filme ainda não está aberta.*

Você consegue pensar numa maneira de descrever o trabalho do recepcionista? Juntamente com seus companheiros de grupo, procure elaborar uma lista de procedimentos e ações que o recepcionista deve cumprir para realizar sua tarefa.

---

<sup>1</sup>Este exemplo foi retirado da página Dicas de Programação (<https://dicasdeprogramacao.com.br/o-que-e-algoritmo/>).

Uma das maneiras de descrever o algoritmo correspondente a situação proposta é mostrada a seguir:

```

Algoritmo Recepcionista de Cinema
Início
  1 – Solicitar ao cliente o bilhete do filme.
  2 – Conferir a data e o horário do filme no bilhete.
  Se data/hora atual > data/hora do filme + 30 minutos Então
    3 – Informar ao cliente que o tempo limite para entrada foi excedido.
    4 – Não permitir a entrada.
  Senão Se data/hora atual < data/hora do filme – 30 minutos Então
    5 – Informar ao cliente que a sala do filme ainda não foi liberada para entrada.
    6 – Não permitir a entrada.
  Senão
    7 – Permitir a entrada.
    8 – Indicar ao cliente onde fica a sala do filme.
Fim-Se
Fim

```

## 4.2 Atividade

Agora vamos propor um pequeno desafio: elabore, junto com seus colegas, um algoritmo para fazer um LED piscar três vezes, esperar 5 segundos e depois reiniciar a piscar. Nesse momento, você não precisa utilizar nenhuma função do Arduino nem código de programação. Basta descrever as ações que devem ser feitas (ex.: ligar, desligar, acender, apagar, esperar, etc.)

### 4.2.1 Utilizando a linguagem de programação

A partir do algoritmo que elaborou, serão apresentadas as estruturas da linguagem de programação necessárias para implementá-lo e, com isso, escreveremos o código.

#### 4.2.1.1 Estruturas de controle

Uma das coisas que você e seus colegas devem ter indicado no seu algoritmo é que é necessário fazer uma repetição, já que o LED precisa piscar três vezes. Para

conseguir fazer isso, precisaremos das estruturas de controle.

*Estruturas de controle* são blocos de instruções que alteram o fluxo de execução do código de um programa. Com elas é possível fazer coisas como executar comandos diferentes de acordo com uma condição ou repetir uma série de comandos várias vezes, por exemplo. Vamos ver ao longo da oficina algumas das estruturas de controle mais comuns usadas nas linguagens de programação em geral. Para fazer a repetição que precisamos, usaremos o *while*.

O *while* é uma estrutura que executa um conjunto de comandos repetidas vezes enquanto uma determinada condição for verdadeira. *While*, em inglês, quer dizer “enquanto”. Ele segue o seguinte formato:

```
while (condicao){  
    ...  
}
```

Nessa estrutura, a palavra *while* deve ser seguida de uma condição definida entre parênteses. Enquanto essa condição for verdadeira, todo o bloco de comandos que estiver entre os caracteres “{” e “}” é executado repetidamente. No nosso caso, a repetição está relacionada a quantidade de vezes que o LED piscou, então, antes de determinarmos a condição que usaremos, precisamos “ensinar” o Arduino a contar.

#### 4.2.1.2 Realizando uma contagem

A primeira coisa que precisamos para contar, é saber como guardar o valor da contagem. Para isso, utilizaremos o que chamamos de *variável*. Uma variável em C, sempre precisa ser declarada. Essa declaração é feita simplesmente indicando o seu tipo e o nome escolhido. Para contar, utilizamos números inteiros, então, trabalharemos com uma variável do tipo *int*. O nome é como você vai se referir a ela ao longo do programa. Pode ser qualquer um, mas o ideal é que seja um que te ajude a lembrar pra que você está usando a variável. Como, nesse caso, ela vai ser nossa contadora e nos ajudar a contar quantas vezes o LED piscou, uma boa opção

de nome é *cont*. A linha de código da declaração dessa variável fica da seguinte maneira.

```
int cont;
```

Quando declaramos uma variável, o programa separa um espaço na memória do computador para guardar a informação que desejamos, e acessamos esse espaço usando o nome escolhido. Podemos atribuir um valor inicial já na declaração da variável. No início do programa, o LED ainda não piscou nenhuma vez, então, vamos inicializar a variável com zero. A declaração com atribuição de um valor inicial fica:

```
int cont = 0;
```

Apesar de não pensarmos muito nisso conscientemente, contar é basicamente somar 1. Começamos com 0 que, somando 1, vira 1 que, somando 1, vira 2 que, somando 1, vira 3, e assim por diante. Então, para contar o número de “piscadas” do LED, basta somar 1 à nossa variável de contagem cada vez que ele piscar. Isso é feito na seguinte linha de código:

```
cont = cont + 1;
```

Aqui estamos dizendo que o valor de *cont* passa a ser o valor anterior mais 1.

Agora que sabemos contar, já podemos estabelecer a condição que usaremos no nosso *while*. Enquanto a quantidade de piscadas do LED (representado pela variável *cont*) for menor do que três, nós continuamos a executar os comandos que fazem o LED piscar. Escreveríamos, então:

```
while (cont < 3){  
    ...  
}
```

Utilizamos essa condição porque quando *cont* for igual a 3, quer dizer que ele já piscou as três vezes necessárias e não é necessário repetir mais uma vez a sequência de comandos. As reticências (...) que estão entre os caracteres { e }, devem ser substituídas pelos comandos utilizados para fazer o LED piscar e para somar 1 ao *cont* após a piscada.

Tendo aprendido a fazer uma repetição e a contar o número de piscadas do LED, e utilizando os comandos para acender, apagar e esperar aprendidos na aula anterior, é possível escrever o programa para realizar a tarefa proposta.

## 4.2.2 Colocando em prática

Agora, monte o circuito necessário, digite o programa no IDE do Arduino e execute-o.

Ele funcionou da maneira que você esperava? Caso tenha surgido algum problema e não tenha funcionado conforme o que foi proposto, tente pensar nas possíveis causas e em propostas de como solucionar. Discuta com seus colegas e/ou com seu professor e teste os ajustes.

Um possível código para executar o que foi proposto é mostrado a seguir:

```
int cont = 0;           // variável inteira para fazer a contagem
void setup() {
    pinMode(13,OUTPUT); // configurando o pino 13 como saída (p/LED)
}

void loop() {
    while(cont<3){
        digitalWrite(13,HIGH); // acende o LED
        delay(500);           // espera 0,5 s
        digitalWrite(13,LOW);  // apaga o LED
        delay(500);           // espera 0,5 s
        cont = cont + 1;       // aumenta 1 na contagem
    }
    delay(4500);              // espera 4,5 s
    cont = 0;                  // reinicia a contagem
}
```

## Capítulo 5

# Considerações sobre a aplicação da atividade nas oficinas de robótica

A abordagem apresentada no capítulo 3 tem sido utilizada nas oficinas oferecidas pelo LaDEF desde que foi proposta, tendo sido registradas em vídeo-gravações as aplicações nas oficinas com estudantes em 2017 e 2020, esta última realizada na modalidade remota, e no curso de extensão realizado com professores. Todas as considerações apresentadas neste capítulo são baseadas nestas gravações.

As oficinas presenciais foram filmadas com uma câmera que, na maior parte do tempo, ficava posicionada de modo a enquadrar a professora aplicadora da atividade, o quadro e parte dos estudantes. Nos momentos em que os estudantes trabalhavam em grupos, ela era deslocada de modo a acompanhar a atividade de um dos grupos; no entanto, ela também conseguia captar o áudio de outros grupos trabalhando. Para oficina remota, foi utilizado o recurso de gravação do Google Meet.

A oficina presencial realizada com os estudantes foi escolhida como referência e será estudada mais detalhadamente. A metodologia empregada consistiu em assistir as vídeo-gravações que registraram a aplicação da atividade e nelas selecionar trechos, aqui chamados de episódios de ensino, que pudessem ser relacionados ao suporte teórico adotado e analisados à luz dos critérios e indicadores apresentados na seção 2.4. Transcrições destes episódios de ensino serão intercaladas com descrições de como foi desenvolvida a atividade de modo que estas possam ser localizadas den-

tro deste contexto e que possam ser identificadas as etapas propostas na seção 3.4. Esse estudo é apresentado na seção 5.1.

Para as demais aplicações, optou-se por somente pontuar as principais diferenças observadas com relação à aplicação já analisada. Esses comentários constituem a seção 5.2 e reforçam algumas particularidades encontradas no ensino remoto e ao se trabalhar com um público-alvo diferente.

## **5.1 Oficina com estudantes do Ensino Médio (2017)**

No momento da aplicação da atividade, os estudantes já foram apresentados às funções básicas do Arduíno e já realizaram um projeto simples utilizando a placa. Por esse motivo, a aula é iniciada com uma recapitulação do que já foi visto e comentando o projeto que foi realizado na aula anterior. Isso é importante não só porque estes conceitos serão utilizados para o desenvolvimento da atividade que será proposta, mas também para que os estudantes que não estiveram na aula anterior possam ser introduzidos a esses conceitos. Nessa recapitulação, reforça-se a importância de fazer de maneira correta a comunicação com o Arduino, que só executará o que for programado para fazer, e a necessidade de se indicar detalhadamente as instruções. Isso dá o gancho para introdução do conceito de algoritmo. A intervenção da professora nessa parte introdutória da aula pode ser classificada de acordo com a Tabela 2.2 como “revendo o progresso da história científica”, pois uma das ações a ela relacionada é justamente recapitular a aula anterior.

Inicia-se, então, a atividade seguindo a abordagem proposta na seção 3.4, sendo sua primeira etapa a apresentação do conceito de algoritmo.



05:30	<b>Professora</b>	<i>Então, hoje a gente vai aprender a fazer um pouco mais... algumas outras coisas, algumas outras estruturas de C mas, antes disso, eu quero a gente pense nessa questão da sequência, porque se é importante que a gente mande passo a passo, pra gente mandar certo, é bom gente pensar antes o que a gente quer mandar, certo? Então, uma coisa que ajuda pra gente é algo que a gente chama de algoritmo. Alguém já ouviu falar em algoritmo?</i>
06:03	<b>Estudantes</b>	<i>[comentários inaudíveis e risadas]</i>
06:09	<b>Professora</b>	<i>Não tem nada a ver com logaritmo, fique tranquilo [...]</i>
06:15	<b>Estudante A</b>	<i>Assim, é só trocar as quatro primeiras letras de posição que já [...]</i>
06:21	<b>Professora</b>	<i>É, mas aí vira uma coisa completamente diferente.</i>
06:24	<b>Professora</b>	<i>Então, o algoritmo nada mais é que uma sequência de passos bem definida que você precisa seguir para fazer uma tarefa. Tipo uma receita de bolo. Se você tem a receita de bolo você vai fazer: acrescente a farinha, bata, coloque o ovo... Se você fizer tudo bonitinho... vai ao forno, 50 min. Se você fizer tudo direito o que você tem no final?</i>
06:43	<b>Estudantes</b>	<i>O bolo.</i>
06:44	<b>Professora</b>	<i>O bolo. Basicamente é só um passo a passo. No dia a dia, a gente até sem pensar a gente está fazendo as coisas e sabe qual a sequência do que a gente vai fazer. Tudo bem? Então, o algoritmo é isso. É um passo a passo. É uma sequência de passos bem definida.</i>

Como pode ser visto na transcrição acima, a professora optou por fazer isso de maneira simples e usando o exemplo da receita de bolo para esclarecer a ideia que queria passar. Nesse trecho, é possível identificar as intervenções “dando forma aos significados” e “marcando significados chaves” (ver Tabela 2.2). Para fixação, o exercício proposto aos estudantes a seguir foi descrever o algoritmo para abrir a porta. Este exercício foi feito seguindo uma abordagem comunicativa interativa/dialógica e, além disso, a professora que está aplicando a atividade foi seguindo as instruções dadas pelos estudantes de modo que pudessem verificar se foram claras e suficientes para realização da tarefa e ajustá-las quando necessário. O exercício

seguiu-se até que a ação proposta, abrir a porta, fosse concluída.

Em seguida, foi proposta aos estudantes a tarefa de elaborar um algoritmo para fazer um LED piscar três vezes e depois esperar 5 segundos. Esclarece-se que, nesse momento, eles não precisam pensar nos comandos, somente na ação que precisa ser feita e que a “tradução” do português para a linguagem C++ será feita posteriormente.

Logo após a apresentação da tarefa, um dos estudantes, de imediato, já propõe:

09:47	<b>Estudante A</b>	<i>liga, desliga, liga, desliga, liga, desliga e aí espera 5 segundos.</i>
-------	--------------------	--

Após repetir a proposta do estudante, a professora faz a seguinte intervenção:

09:54	<b>Professora</b>	<i>Beleza, mas três vezes é mole. E se fossem 100 vezes?</i>
-------	-------------------	--

Aqui tem-se um exemplo do uso da abordagem comunicativa interativa/de auto-riedade. A professora, com essa intervenção, está tentando fazer com que o estudante perceba que, ao invés de ele colocar as etapas novamente pela quantidade de vezes que foi pedida, ele pode indicar que é necessário que haja uma repetição. A interação com os estudantes continua e, conforme eles vão indicando as etapas, a professora vai escrevendo no quadro. A primeira versão do algoritmo elaborado pelos estudantes é mostrada a seguir:

liga

desliga

repete 3 vezes

espera 5 segundos

De posse do algoritmo, é necessário identificar o que os estudantes já sabem como fazer e o que precisam aprender.

11:32	<b>Professora</b>	<i>Bom, o que que a gente já sabe fazer disso aí? Depois a gente vai descobrir se isso vai ter problema ou não. (...)</i>
11:37	<b>Estudante A</b>	<i>A gente já sabe ligar e desligar e esperar 5 s.</i>
11:40	<b>Professora</b>	<i>A gente já sabe ligar e desligar e sabe esperar 5 s. O que que a gente não sabe?</i>
11:45	<b>Estudante A</b>	<i>Repetir.</i>
11:46	<b>Professora</b>	<i>Repetir três vezes.</i>

Neste trecho está presente a forma de intervenção “selecionando significados” pois a professora considera a resposta do estudante na sua fala. Inicia-se, então, a quarta etapa da atividade que consiste na apresentação das estruturas na linguagem C++ necessárias para implementação do algoritmo.

Diante da problemática de que é necessário aprender como repetir três vezes, um estudante apresenta uma proposta:

11:50	<b>Estudante A</b>	<i>Assim, [...] eu sei uma maneira teórica de fazer isso aí com... criando uma variável e fazendo um ciclo if-else [...]</i>
-------	--------------------	--

Este comentário do estudante indica que ele já teve contato com alguma linguagem de programação. A partir dessa intervenção, a professora introduz o conceito de variável. Ao ser levantado o questionamento do porquê se precisa de uma variável, o próprio estudante consegue indicar que ela vai estar relacionada à contagem. Para aproveitar o conhecimento prévio dos estudantes, a professora faz uma analogia com a variável utilizada em matemática, lembrando que está relacionada a algo que pode mudar, que pode variar. Isto pode ser observado no trecho transcrito a seguir:

12:07	<b>Professora</b>	<i>Vamos começar pela primeira palavra que ele falou: variável. Por que que eu preciso de uma variável?</i>
12:11	<b>Estudante A</b>	<i>Pra você saber quantas vezes você tem [...]</i>
12:13	<b>Professora</b>	<i>Exatamente. Eu quero saber quando que eu cheguei a três vezes. Então, pra fazer isso eu preciso...</i>
12:18	<b>Estudante A</b>	<i>de uma variável</i>
12:19	<b>Professora</b>	<i>...contar, então, eu preciso de uma coisa que vai mudar de valor. Ele não está errado, não. Eu vou precisar usar uma variável. O que que é uma variável? [...] É igual ao nosso bom e velho e amado <math>x</math></i>
12:32	<b>Estudantes</b>	<i>[sons de reclamação]</i>
12:34	<b>Professora</b>	<i>o <math>x</math> não é uma variável? O que que é uma variável? O que que define uma variável? É uma coisa que pode ter um valor diferente. Variável é uma coisa que...</i>
12:44	<b>Estudante A</b>	<i>muda</i>
12:45	<b>Professora</b>	<i>...muda. Então, ela é justamente o que a gente precisa fazer.</i>

Na sequência da atividade, a professora faz uma síntese apresentando os aspectos necessários para utilizar uma variável na programação: definir um tipo e um nome. Estes dois elementos são definidos na declaração. Os estudantes conseguem identificar que, para contar, a variável deve ser do tipo inteira (*int*).

13:01	<b>Professora</b>	<i>A gente pode ter variáveis de vários tipos em programação. Então, a primeira coisa que a gente tem que dizer é se é uma variável inteira, se é real, se é um caracter. Eu posso usar uma variável pra armazenar letra também, pra escrever palavra? Posso. Mas, pra contar, que tipo de variável será que eu preciso?</i>
13:18	<b>Estudante A</b>	<i>Inteira</i>
13:19	<b>Professora</b>	<i>Inteira, né? 1, 2, 3, 4, ... Inteiro serve pra mim? Então, uma coisa que a gente vai fazer é sempre que a gente for usar uma variável, a gente tem que fazer uma coisa primeiro que é declarar. O que que é isso? Você vai dar um nome pra variável [...] o valor muda mas você tem que saber do que está chamando ela. Você vai dar um nome para ela e vai dizer de que tipo ela é.</i>

Ao saber que o nome é de sua escolha, os estudantes acabam optando por usar *pimba*. Apesar da estranheza do nome, respeitar a escolha dos estudantes reforça o seu protagonismo e o fato de que eles é que estão realizando a atividade e a professora exerce o papel de auxiliar e mediadora.

14:03	<b>Professora</b>	<i>Então, pra representar uma variável inteira, a gente escreve int. Int de inteiro. Vocês vão lembrar? E aí, depois, eu tenho que escolher um nome para variável.</i>
14:14	<b>Estudante A</b>	<i>Número.</i>
14:15	<b>Professora</b>	<i>O nome pode ser qualquer nome, tá? Você quer chamar a variável de número, então, vou chamar a variável de número. Vamos chamar de número?</i>
14:23	<b>Estudante A</b>	<i>Pode ser pagode, samba, ...</i>
14:26	<b>Professora</b>	<i>Pagode, samba, ...</i>
14:27	<b>Estudante B</b>	<i>Pimba</i>
14:29	<b>Professora</b>	<i>Pimba? Vamos chamar de pimba, então? Nossa variável contadora é pimba.</i>

Apesar de predominar uma abordagem comunicativa interativa durante a atividade, é possível encontrar momentos em que é utilizada uma abordagem não-interativa/de autoridade, principalmente quando é reforçado algum conceito ou se deseja passar alguma instrução, como no trecho a seguir:

14:35	<b>Professora</b>	<i>Então, quando a gente precisar usar uma variável, a gente vai declarar: que é dizer qual é o tipo dela... a nossa é inteira. Se fosse de caracteres a gente usaria char. [...] A nossa é inteira, por enquanto, vamos ficar só com inteiro. Chamei de pimba. Por que que eu coloquei só isso e ponto e vírgula?</i>
15:07	<b>Professora</b>	<i>Quando eu faço isso, eu estou dizendo que eu tenho uma variável inteira que eu vou chamar de pimba, então, ele reserva um espaço de memória que ele vai botar o que quer que eu diga que seja pimba. Essa declaração, eu faço lá no início do meu programa. Quando eu for fazer eu vou botar isso antes até do setup. É só pra ele saber quem é pimba. Agora eu posso usar pimba.</i>

Em seguida, na discussão da atribuição de valor da variável, já é retomada a abordagem comunicativa interativa/de autoridade.

15:25	<b>Professora</b>	<i>Quanto vale pimba nesse momento?</i>
15:27	<b>Estudante C</b>	<i>Zero.</i>
15:28	<b>Estudante A</b>	<i>Ele não tem valor definido.</i>
15:29	<b>Professora</b>	<i>Eu não sei. Eu falei quanto vale? [...] Se eu não falar, o programa vai saber? Então, o que que ele vai me dar se eu mandar ele dizer que que é pimba? Pode ser qualquer coisa. Ele pegou um espaço de memória e vai ter alguma coisa lá de algum programa que rodou antes. Então, se eu quiser dar um valor inicial, eu posso dar um valor inicial já na declaração, ou eu posso dizer depois. Então, eu posso dizer ó... pimba = 0. Agora, quanto vale pimba?</i>
15:55	<b>Estudantes</b>	<i>Zero.</i>
15:56	<b>Professora</b>	<i>Zero. Eu posso mudar o valor de pimba?</i>
15:58	<b>Estudantes</b>	<i>Pode.</i>
15:59	<b>Professora</b>	<i>Posso. É uma variável, eu posso variar ao longo do programa. Ou eu poderia só declarar o pimba...</i>
16:04	<b>Estudante A</b>	<i>E depois atribuir um valor pro pimba.</i>
16:05	<b>Professora</b>	<i>E depois atribuir um valor pro pimba. Dizer que pimba é igual a zero. Lembra que qualquer coisa que eu escreva tem que botar no final...</i>
16:14	<b>Estudante A</b>	<i>Ponto e vírgula.</i>
16:15	<b>Professora</b>	<i>Ponto e vírgula.</i>

Para finalizar a discussão sobre a contagem, é discutido como é feita a atualização do valor da variável ao se somar 1 ao valor anterior, e que isso é incluído após a “piscada”.

Já sabendo como realizar a contagem, o último elemento que falta para que os estudantes escrevam o código correspondente ao algoritmo que propuseram é a repetição. Eles são, então, introduzidos ao comando *while*, estrutura de controle utilizada para repetição que vai executar uma sequência de comandos determinada enquanto a condição estabelecida for verdadeira. Os estudantes indicam a condição a ser utilizada como sendo  $pimba < 3$ . Um trecho desta etapa é transcrito a seguir:

18:05	<b>Professora</b>	<i>Então, repetir. Aí repete. Pra isso a gente precisa aprender um outro comando. A gente tem alguns comandos em C que permitem a gente fazer isso. Um deles é o que a gente chama de while.</i>
18:20	<b>Estudante A</b>	<i>Enquanto... Enquanto pimba menor que três</i>
18:25	<b>Professora</b>	<i>While é uma palavra em inglês que, como ele adiantou, quer dizer enquanto. Então, o que eu botar dentro do while, ele vai fazer enquanto uma condição for verdadeira. Então, qual é a estrutura do while? Primeiro, eu escrevo while. Segundo, eu tenho que dizer enquanto o quê? Ah, não. Faz aí enquanto isso acontecer. Enquanto o quê? Eu tenho que saber qual é a condição. Qual vai ser a minha condição?</i>
18:47	<b>Estudante A</b>	<i>Pimba menor que... Enquanto pimba menor que três.</i>
18:50	<b>Professora</b>	<i>Por que enquanto pimba menor que três?</i>
18:51	<b>Estudante A</b>	<i>Porque tem que repetir três vezes.</i>
18:52	<b>Professora</b>	<i>Porque eu quero repetir três vezes. Todo mundo concorda com isso? Porque quando pimba for zero, vai a primeira vez. Quando pimba for um, vai a segunda vez. Quando pimba for dois, vai a terceira vez. Quando pimba for três, ele já fez três vezes, aí ele não faz mais. Então, faz sentido? Como é que a gente escreve essa condição? Enquanto, é a estrutura, pimba menor que três. Aí o que que ele vai fazer? [...] Eu boto a condição primeiro e depois eu digo o que que ele vai fazer repetidas vezes, o que que ele vai fazer enquanto pimba for menor que três. Faz sentido isso?</i>

É interessante observar que nesse trecho aparece a segunda ação associada à intervenção “selecionando significados” (Tabela 2.2). Num primeiro momento, a professora escolhe ignorar parte da primeira resposta do estudante A. Que é somente considerada em um momento posterior. É importante ressaltar que essa aula é dada com o auxílio do quadro e nele é escrita a estrutura da maneira que deve ser inserida no código conforme a professora apresenta a explicação.

Uma vez que os estudantes já foram apresentados às estruturas que desconheciam, é possível construir o código associado ao algoritmo proposto. Durante a elaboração do código, um estudante propõe uma alteração:

24:36	<b>Estudante B</b>	<i>Eu acho que o delay tinha que estar antes do digitalWrite(13,LOW).</i>
24:41	<b>Professora</b>	<i>[...] Ele tem aqui uma proposta. Por que você acha que tem que ter um delay ali antes?</i>
24:47	<b>Estudante B</b>	<i>Porque [...] o LED tem que ficar aceso por um tempo.</i>
24:52	<b>Professora</b>	<i>Então o que você acha que vai acontecer se eu não colocar o delay ali?</i>
24:54	<b>Estudante B</b>	<i>[...] Vai acender e apagar muito rápido.</i>
24:58	<b>Professora</b>	<i>E aí vai dar pra ver piscando?</i>
25:00	<b>Estudante B</b>	<i>Não.</i>

Este momento da aula nos permite identificar alguns dos indicadores de alfabetização científica apresentados na Tabela 2.3. O estudante levanta uma hipótese, apresenta a justificativa e faz uma previsão. O teste dessa hipótese poderá ser feito com a montagem do circuito, a inclusão do programa no IDE do Arduino e seu respectivo *upload* na placa, quando, então, ele poderá verificar se o funcionamento foi conforme o esperado. Aqui também é possível observar algumas formas de intervenção apresentadas na Tabela 2.2: “compartilhando significados”, quando a professora pede que o estudante repita seu enunciado para a classe, e “checando o entendimento dos estudantes”, pois a professora pede que o estudante explique melhor a sua ideia.

Apesar de inseri-lo no código escrito acatando a sugestão proposta, a professora deixa a critério de cada estudante incluir ou não esse *delay* e pede que verifiquem se faz diferença essa inclusão. O código elaborado a partir do algoritmo dos estudantes, incluindo essa mudança proposta, é mostrado na Figura 5.1. Os estudantes partem, então, para a montagem do circuito e execução do programa onde podem verificar se foram atingidos os objetivos propostos.



```

int pimba = 0;
void setup() {
  pinMode(13,OUTPUT);
}

void loop() {
  while(pimba<3){
    digitalWrite(13,HIGH);
    delay(500);
    digitalWrite(13,LOW);
    pimba++;
  }
  delay(5000);
}

```

Figura 5.1: Primeira versão do código escrito a partir do algoritmo elaborado pelos estudantes para a solução do problema proposto.

Enquanto os grupos trabalham no projeto, a professora acompanha seu progresso para auxiliá-los, se necessário. A câmera, neste momento da atividade, fica mais próxima da dupla formada pelos estudantes B e C. Um dos erros encontrado é transcrito a seguir:

44:52	<b>Professora</b>	<i>Fez?</i>
44:53	<b>Estudante B</b>	<i>Aham. A gente vai verificar agora. Não. Tá salvando, eu quero verificar.</i>
45:01	<b>Professora</b>	<i>Ele tá reclamando que o pimba não foi declarado. Que que é declarar? Lembra que eu falei que antes de usar eu tenho que dizer de qual tipo é o pimba? Vocês não colocaram.</i>
45:08	<b>Estudante B</b>	<i>E a gente tem que colocar aonde? No void setup?</i>
45:11	<b>Professora</b>	<i>Antes do void setup. Dá um enter aí...</i>
45:13	<b>Estudante C</b>	<i>Ah! int pimba = 0;. A gente esqueceu.</i>
45:16	<b>Professora</b>	<i>Se você não declarar a variável, ele não sabe de que tipo que é, então, ele vai reclamar. Quando você manda ele olhar se pimba é menor do que três, aí ele vai falar: 'bom, não sei quem é pimba'. Aí ele reclama: 'ó, pimba não foi declarado'.</i>

Neste trecho, observa-se que os estudantes esqueceram um elemento fundamental que tinha sido discutido em aula. A professora, ao explicar o erro encontrado, aproveita para reforçar o conceito que tinha sido introduzido e sua importância. A

exclamação do estudante C surge quando ele encontra no quadro a parte que eles esqueceram de incluir no código.

Depois de montado o circuito e do programa ter sido carregado à placa Arduino, os estudantes percebem que o ciclo não reinicia: ao invés de ter uma espera de 5 s, ele fica apagado permanentemente.

49:06	<b>Professora</b>	<i>Já tinha dado as duas piscadas antes?</i>
49:08	<b>Estudante C</b>	<i>Acho que sim. A gente pode botar de novo aí.</i>
49:09	<b>Professora</b>	<i>Porque ele só vai piscar três vezes e vai parar. Por quê? Por que que ele não pisca mais? Por que que agora ele tá apagado? Olha pro código e pensa comigo.</i>
49:19	<b>Estudante C</b>	<i>Pera, tinha que botar...</i>
49:21	<b>Professora</b>	<i>[...] O que que eu mando repetir do loop? Quando ele chega aqui ele volta pra cá.</i>
49:26	<b>Estudante C</b>	<i>Ele vai piscar três vezes e vai apagar porque você falou while é igual a isso...</i>
49:30	<b>Professora</b>	<i>menor 3. Quando ele acaba, o pimba vale quanto?</i>
49:33	<b>Estudante B</b>	<i>Três.</i>
49:34	<b>Professora</b>	<i>Três. E ele vai continuar valendo...</i>
49:36	<b>Estudante B</b>	<i>Três.</i>
49:37	<b>Professora</b>	<i>Três. Então, ele vai acender de novo?</i>
49:38	<b>Estudantes</b>	<i>Não.</i>
49:39	<b>Professora</b>	<i>A menos que você... mude alguma coisa. Você pode... Ah, não. Eu quero que ele faça 3 vezes, espere 5 e faça depois de novo três vezes. O que que eu tenho que fazer, então? Eu tenho que começar a contagem de novo, certo? Então, eu tenho que zerar meu contador. Então, você tem que zerar o pimba se você quiser que faça de novo. Então, ele fez exatamente o que você mandou. É aquilo ele fez 1, 2, 3 vezes, depois da terceira vez ele vai esperar os 5 segundos e depois não vai fazer mais porque essa condição aqui não é mais verdadeira. Aí ele não vai acender mais.</i>

É importante mencionar que esta discussão é feita acompanhada de gestos apontando para partes do código na tela do computador. Através da leitura das instruções contidas no código, a professora vai indicando o que está acontecendo até que os estudantes percebam a causa do problema.

A câmera encerra o primeiro arquivo de vídeo após 52 minutos e 30 segundos de gravação e os 15 minutos finais da aula ficam num arquivo separado. As próximas transcrições foram extraídas desse segundo arquivo.

Um outro problema é encontrado pelos estudantes, conforme mostrado a seguir:

02:14	<b>Estudante B</b>	<i>Ele não tá... ele não tá repetindo três vezes.</i>
02:20	<b>Professora</b>	<i>Ele não tá fazendo três vezes... Por quê? Temos que descobrir isso. O dele também não tá fazendo.</i>
02:27	<b>Estudante C</b>	<i>Ah, então tem alguma coisa aqui que você não falou pra gente que tava errado...</i>

Ao mencionar que outro grupo teve o mesmo problema, a professora dá uma indicação de que o erro pode estar relacionado ao que foi proposto. No entanto, ela opta por não sanar a dúvida imediatamente. Deixa um tempo para refletirem sobre o problema e fazerem testes enquanto atende o outro grupo com problema na placa. Após alguns minutos, ela retorna para auxiliá-los.

05:23	<b>Professora</b>	<i>O problema é o seguinte: o que que ele tá fazendo? Ele acende, espera 1 segundo, apaga. Depois o que que ele faz? Soma 1 e acende de novo. Quanto tempo tem entre esse comando e esse?</i>
05:33	<b>Estudante B</b>	<i>Nenhum.</i>
05:34	<b>Professora</b>	<i>Nenhum. Então você não vê ele apagar. É como se ele ficasse aceso direto. Então, parece que você acendeu 1 vez, esperou 5 segundos... Pelo mesmo motivo que você precisa colocar um delay pra ele ficar aceso, você precisa colocar um delay...</i>
05:45	<b>Estudante C</b>	<i>pra ele ficar apagado.</i>
05:46	<b>Professora</b>	<i>...pra ele ficar apagado. Senão você não vê ele piscar.</i>
05:49	<b>Estudante C</b>	<i>Faz sentido.</i>
05:50	<b>Professora</b>	<i>E aí parece que ele tá fazendo uma vez só mas, na verdade, não.</i>

É possível observar que a professora relaciona a solução desse problema com a proposta que o estudante B fez quando criavam o código a partir do algoritmo. Depois de solucionado o problema, os estudantes resolvem comentar sobre um dos testes que fizeram pra tentar resolver.

06:35	<b>Estudante B</b>	<i>Eu tinha dado ctrl-C ctrl-V ali três vezes pra ver e não deu certo.</i>
06:39	<b>Estudante C</b>	<i>Não deu certo, ficou aceso direto.</i>
06:40	<b>Professora</b>	<i>Pelo mesmo motivo. O espaço entre o LOW e o HIGH era nenhum. É que nem o sinal de trânsito. Quando você manda apagar um e apaga o outro você não vê o tempo entre um apagar e o outro acender porque é quase ao mesmo tempo, isso é muito rápido. Então, era como se ele ficasse aceso direto.</i>

O interessante desse trecho é identificar alguns indicadores da Alfabetização Científica (Tabela 2.3). O *raciocínio proporcional* pode ser associado ao fato de que, ao perceber que o LED não pisca três vezes, ele relaciona o problema à estrutura de repetição (*while*). Então, *levantada essa hipótese*, ele a *testa* substituindo o *while* pela repetição dos comandos que copiou e colou. E, ao perceber que o problema não foi resolvido, ele sabe que sua hipótese inicial estava errada.

Depois que os estudantes testaram e ajustaram os seus códigos, a professora faz o fechamento da atividade para saber os problemas que tiveram e como fica o código corrigido.

09:24	<b>Professora</b>	<i>Agora que vocês já fizeram, fala pra mim o que ficou errado no nosso código aqui do quadro. O que que faltou?</i>
09:30	<b>Estudante B</b>	<i>Faltou o delay pra ficar um tempo apagado</i>
09:30	<b>Estudante A</b>	<i>[inaudível]</i>
09:36	<b>Professora</b>	<i>Faltou um delay aqui antes do pimba++, né? Porque senão ele não pisca. Ele apaga e acende na mesma hora, a gente não vê piscar, certo?</i>
09:44	<b>Estudante C</b>	<i>Certo.</i>
09:45	<b>Professora</b>	<i>Então, vamos botar aqui no mesmo tempo pra fazer a piscada.</i>
09:48	<b>Estudante C</b>	<i>E o pimba = 0 no final também.</i>
09:52	<b>Estudante A</b>	<i>pimba = 0 também depois.</i>

A versão final do código obtida após serem incluídas essas correções é apresentada na Figura 5.2.

```

int pimba = 0;
void setup() {
    pinMode(13,OUTPUT);
}

void loop() {
    while(pimba<3){
        digitalWrite(13,HIGH);
        delay(500);
        digitalWrite(13,LOW);
        delay(500);
        pimba++;
    }
    delay(5000);
    pimba = 0;
}

```

Figura 5.2: Versão final do código escrito a partir do algoritmo elaborado pelos estudantes para a solução do problema proposto, obtida após realizadas as devidas correções.

## 5.2 Comentários sobre outras aplicações

### 5.2.1 Oficina remota com estudantes

Em 2020, devido à pandemia do novo coronavírus, foram suspensas as aulas presenciais e as atividades de ensino do CAp-UFRJ passaram a ser realizadas de maneira remota. Depois de serem estudadas diferentes plataformas que viabilizassem a sua realização, as oficinas de robótica também passaram a ser oferecidas nessa modalidade. Os encontros síncronos foram feitos através do *Google Meet*, a montagem dos circuitos e programação foram feitas utilizando a plataforma *Tinkercad* e o *Google Sala de Aula* foi utilizado para envio das atividades e compartilhamento do material didático. O aplicativo *WhatsApp* também foi utilizado para facilitar a interação com os estudantes.

Com os encontros síncronos, foi possível manter uma abordagem interativa com os estudantes, mas o ambiente remoto possui algumas limitações. Um exemplo relacionado à atividade: uma vez que a visão do professor é limitada ao que é enquadrado na câmera do computador, já não era adequado propor aos estudantes

que descrevessem o algoritmo para abrir a porta uma vez que não seria possível visualizar as etapas sendo executadas. A ação escolhida em substituição foi dar tchau. Esse exemplo funcionou bem e atingiu os objetivos propostos.

Uma outra mudança importante foi que o trabalho dos estudantes passou a ser individual. Nas oficinas presenciais, os estudantes planejavam e construíam os projetos em grupo, trocando ideias e se ajudando. Estando isolados em suas casas e desenvolvendo seus projetos em ambiente virtual, apesar de interagirem, poderem mostrar o andamento do trabalho e tirarem dúvidas durante os encontros síncronos, a maior parte do desenvolvimento dos projetos era feita individualmente. Como consequência, os estudantes demoraram mais tempo para concluir os projetos. O *Google Sala de Aula* foi uma ferramenta importante que auxiliou no acompanhamento individual dos estudantes no desenvolvimento de suas tarefas.

Um fator facilitador foi que os estudantes desta turma tinham tido as primeiras aulas da oficina 2 presencialmente antes da pandemia, incluindo a da aplicação dessa atividade. Houve um intervalo de aproximadamente 7 meses entre a suspensão das aulas presenciais e o início da oficina remota. Diante de tão longo período, não se poderia esperar que os estudantes recordassem o que já tinha sido abordado e ainda era necessário inserir novas ferramentas, então, as atividades foram reiniciadas. Mas os estudantes lembravam de algumas coisas e isso ajudou.

### 5.2.2 Oficina com professores

No segundo semestre de 2017, foi oferecido um curso de extensão para professores de física da rede pública de ensino com o objetivo de ampliar o alcance da robótica educativa e do ensino por investigação no espaço de sala de aula. Além de apresentar as atividades já desenvolvidas no LaDEF e as bases do ensino por investigação, procurava capacitá-los para desenvolver projetos semelhantes em suas escolas.

Nesse curso, a atividade ainda foi aplicada seguindo as diretrizes apresentadas nesse trabalho mas, devido ao público a que se dirigia, foi possível discutir antes o

seu contexto e os seus objetivos. Desta maneira, as intervenções dos professores que participavam do curso incluíam também comentários sobre como esperavam que seus estudantes reagiriam em determinados momentos da atividade, sugestões de como abordar alguns pontos e esclarecimentos sobre a melhor maneira de proceder quando estivessem fazendo a aplicação. Dentre os pontos discutidos, falou-se sobre quais informações devem ser fornecidas e o que o estudante conseguiria resolver somente com o conhecimento prévio que possui e sobre a importância de estar atento a tudo o que ele propor durante a atividade. Foi um momento de troca muito interessante que também ajudou a enriquecer esse trabalho.

## Capítulo 6

# Considerações Finais

O espaço conquistado pela tecnologia na sociedade reforça a necessidade de inseri-la no ambiente de sala de aula e a robótica educativa tem exercido bem esse papel sendo utilizada em diversas áreas, inclusive no ensino de ciências. O trabalho aqui apresentado teve o objetivo de enfrentar o desafio de ensinar as ferramentas tecnológicas sem tirar do estudante a autonomia e o protagonismo na realização das tarefas.

O referencial teórico escolhido foi o do Ensino por Investigação e a primeira etapa do trabalho consistiu no estudo de artigos relacionados para se entender suas principais características e os elementos a serem considerados na atividade a ser proposta.

No contexto das Oficinas de Acionamento e Robótica realizadas pelo Laboratório Didático de Ensino de Física (LaDEF) do CAp-UFRJ, a linguagem de programação se torna necessária na automação de protótipos robóticos utilizando placas controladoras, como a placa Arduino. Foi, então, feita a proposição de uma atividade para ensino das estruturas de controle em C++ utilizando uma nova abordagem. Nessa atividade, o conceito de algoritmo, bastante utilizado na área de informática e relacionado ao ensino de programação, exerceu um papel fundamental.

Esta atividade foi aplicada tanto nas oficinas para estudantes do Ensino Básico quanto no curso de extensão para professores da Rede Pública, ambos oferecidos pelo LaDEF (CAp-UFRJ). Através da análise dos registros das atividades buscou-se



avaliar a abordagem comunicativa e as características das intervenções da professora aplicadora da atividade e identificar, nas interações entre os estudantes, indicadores que evidenciassem um processo de alfabetização científica e tecnológica. Durante a realização da atividade e análise de alguns trechos das gravações, foi possível observar que os estudantes se envolveram em atividades dialógicas discursivas que continham: levantamento e teste de hipóteses, planejamento, organização e seleção de dados para análises da situação, avaliação e elaboração de novas estratégias para atingirem o objetivo desejado. Estas características são amplamente esperadas para um ensino que utiliza metodologia investigativa com o objetivo de promover a formação técnica e científica dos estudantes.

## Referências Bibliográficas

AGUILAR, L. **Fundamentos da Programação – Algoritmos, estruturas de dados e objetos**. 3. ed. São Paulo: Mc-Graw Hill, 2008.

BORGES, A. T. Novos rumos para o laboratório escolar de ciências. **Caderno Brasileiro de Ensino de Física**, v. 19, n. 3, p. 9–30, 2002.

CAPECCHI, M. C. V. M.; CARVALHO, A. M. P. Atividade de laboratório como instrumento para a abordagem de aspectos da cultura científica em sala de aula. **Pro-posições**, v. 17, n. 1(49), 2006.

CARVALHO, A. M. P. O ensino de ciências e a proposição de sequências de ensino investigativas. In: CARVALHO, A. M. P. (Ed.). **Ensino de ciências por investigação: condições para implementação em sala de aula**. São Paulo: Cengage Learning, 2013. p. 1–20.

CHALMERS, A. F. **O que é ciência afinal?** São Paulo: Editora Brasiliense, 1993.

D'ABREU, J. V. V.; RAMOS, J. J. G.; MIRISOLA, L. G. B.; BERNARDI, N. Robótica educativa/pedagógica na era digital. In: **Atas do II Congresso Internacional TIC e Educação**. Lisboa: Instituto de Educação da Universidade de Lisboa, 2012. p. 2449–2465. Disponível em: <http://ticeduca.ie.ul.pt/atas/pdf/158.pdf>.

DAVIS, C.; OLIVEIRA, Z. M. R. **Psicologia na Educação**. São Paulo: Cortez, 1994.

GOMES, A.; HENRIQUES, J.; MENDES, A. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. **Educação, Formação & Tecnologias**, v. 1, p. 93–103, 2008. Disponível em: <https://eft.educom.pt/index.php/eft/article/view/23/16>.

LOPES, L.; SANTOS, L. M. M. d.; SOUZA, L. F. F. d.; BARROSO, M. F. S.; SILVA, C. V. d.; SERPA, B. R.; PEREIRA, E. B. A robótica educacional como ferramenta multidisciplinar: um estudo de caso para a formação e inclusão de pessoas com deficiência. **Revista Educação Especial**, Santa Maria, Brasil, v. 28, n. 53, p. 735–749, 2015. ISSN 1808-270X. Disponível em: <https://www.redalyc.org/articulo.oa?id=313141512018>.

MALIUK, K. D. **Robótica Educacional como cenário investigativo nas aulas de matemática**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2009.

MOREIRA, M. A.; OSTERMAN, F. Sobre o ensino do método científico. **Caderno Catarinense de Ensino de Física**, v. 10, n. 2, p. 108–117, 1993.

MORTIMER, E. F.; SCOTT, P. Atividade discursiva na sala de aula de ciências: uma ferramenta sociocultural para analisar e planejar o ensino. **Investigações para o Ensino de Ciências**, v. 7, n. 3, p. 283–306, 2002.

MUNFORD, D.; LIMA, M. E. C. C. Ensinar ciências por investigação: em que estamos de acordo? **Ensaio Pesquisa em Educação em Ciências (Belo Horizonte)**, v. 09, n. 01, p. 89–111, 2007.

OLIVEIRA, D. F. R. **A robótica educativa no ensino e aprendizagem de conceitos de programação e algoritmos**. Dissertação (Mestrado) — Universidade de Lisboa, Lisboa, 2013. Disponível em: [https://repositorio.ul.pt/bitstream/10451/9892/1/ulfpie044799\\_tm.pdf](https://repositorio.ul.pt/bitstream/10451/9892/1/ulfpie044799_tm.pdf).

PENHA, S. P. **A Física e a sociedade na TV**. Dissertação (Mestrado) — Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ, Rio de Janeiro, 2006.

PENHA, S. P. **Atividades Sociocientíficas em sala de aula de Física: as argumentações dos estudantes**. Tese (Doutorado) — Universidade de São Paulo, São Paulo, 2012.

PENHA, S. P.; CARVALHO, A. M. P.; VIANNA, D. M. Laboratório didático investigativo e os objetivos da enculturação científica: análise do processo. **Revista de Educação, Ciências e Matemática**, v. 5, n. 2, p. 6–23, 2015. ISSN 2238-2380.

PENHA, S. P. da. Como inserir objetos e ferramentas tecnológicas em atividades de investigação: um estudo de caso sobre "oficinas de robótica". In: **XII Encontro Nacional de Pesquisa em Educação em Ciências - XII ENPEC**. Natal/RN: Associação Brasileira de Pesquisa em Educação em Ciências (ABRAPEC), 2019. Disponível em: <http://www.abrapecnet.org.br/enpec/xii-enpec/anais/resumos/1/R1318-1.pdf>.

PHILLIPS, P. Computational thinking a problem-solving tool for every classroom. 2009. Disponível em: <http://ksuweb.kennesaw.edu/dlo2/CT/documents/CompThinking.pdf>. Acesso em 8 de abril de 2021.

REZENDE, L. T. T.; ARAÚJO, L. S. de; SANTOS, G. S.; BATISTA, M. S. K.; TOSTES, D. M.; PENHA, S. P. da. Curso de extensão para professores de física com uma abordagem investigativa. In: **XXIII Simpósio Nacional de Ensino de**

**Física – SNEF**. Salvador/BA: Sociedade Brasileira de Física, 2019. Disponível em: <https://sec.sbfisica.org.br/eventos/snef/xxiii/sys/resumos/T0067-1.pdf>.

REZENDE, L. T. T.; SANTOS, G. S.; TOSTES, D. M.; ARAÚJO, L. S. de; BATISTA, M. S. K.; PENHA, S. P. da. Uso de materiais de baixo custo para oficinas de robótica. In: **XXIII Simpósio Nacional de Ensino de Física – SNEF**. Salvador/BA: Sociedade Brasileira de Física, 2019. Disponível em: <https://sec.sbfisica.org.br/eventos/snef/xxiii/sys/resumos/T0067-2.pdf>.

RODRIGUES, B. A.; BORGES, A. T. O ensino de ciências por investigação: reconstrução histórica. **Atas do XI Encontro de Pesquisa em Ensino de Física**, Sociedade Brasileira de Física, Curitiba, 2008.

SANTOS, W. L. P. Educação científica na perspectiva de letramento como prática social: funções, princípios e desafios. **Revista Brasileira de Educação**, v. 12, n. 36, p. 474–550, 2007.

SASSERON, L. H.; CARVALHO, A. M. P. Almejando a alfabetização científica no ensino fundamental: a proposição e a procura de indicadores do processo. **Investigações para o Ensino de Ciências**, v. 13, n. 3, p. 333–352, 2008.

\_\_\_\_\_. Alfabetização científica: uma revisão bibliográfica. **Investigações em Ensino de Ciências**, v. 16, n. 1, p. 59–77, 2011.

SEBESTA, R. W. **Concepts of Programming Languages**. 7. ed. [S.l.]: Addison-Wesley, 2006.

VALLIM, M. B. R.; HERDEN, A.; GALLO, R.; CARDOSO, L. R.; BITENCOURT, L. C. Incentivando carreiras na área tecnológica através da robótica educacional. In: **XXXVII Congresso Brasileiro de Educação em Engenharia**. Recife/PE: Associação Brasileira de Educação em Engenharia, 2009. Disponível em: <http://www.abenge.org.br/cobenge/arquivos/10/artigos/694.pdf>.

VARGAS, M. N.; MENEZES, A. G. C. d.; MASSARO, C. M.; CALVES, T. D. M. G. Utilização da robótica educacional como ferramenta lúdica de aprendizagem na engenharia de produção: introdução à produção automatizada. In: **XL Congresso Brasileiro de Educação em Engenharia**. Belém/PA: Associação Brasileira de Educação em Engenharia, 2012. Disponível em: <http://www.abenge.org.br/cobenge/arquivos/7/artigos/104401.pdf>.

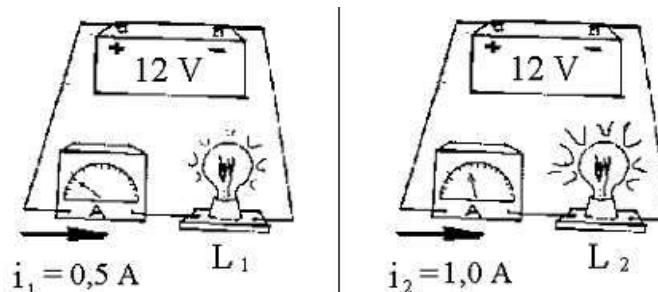
WING, J. Computational thinking. **Communications of the ACM**, v. 49, n. 3, 2006. Disponível em: <http://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>.

# **Anexo 1**

## **Trecho da apostila da Oficina 1**

## 5. Resistência Elétrica

Sejam duas lâmpadas  $L_1$  e  $L_2$ , ligadas a uma bateria de automóvel cuja d.d.p. é de 12 V. Sejam também dois amperímetros utilizados para medir a corrente elétrica que passará em cada uma das lâmpadas.



Observe que a lâmpada 2 “deixa passar” uma maior quantidade de corrente elétrica que a lâmpada 1. Podemos então dizer que o filamento da lâmpada 1 oferece maior dificuldade para a passagem da corrente elétrica que o filamento da lâmpada 2. A esta propriedade que o filamento da lâmpada possui de “**resistir**” (dificultar) a passagem da corrente elétrica dá-se o nome de **Resistência Elétrica**.

Os resistores são os elementos dos circuitos elétricos que transformam a energia elétrica em energia térmica. A este fenômeno damos o nome de Efeito Joule.

Quando uma corrente elétrica está percorrendo um condutor (o filamento da das Lâmpadas acima por exemplo), os elétrons livres são acelerados pela Voltagem estabelecida nos seus terminais e realizarão colisões contra os átomos e moléculas deste condutor. A agitação molecular das partículas que constituem o material irão aumentar ou seja, o condutor aumentará sua temperatura.

A resistência elétrica de um determinado material dependerá entre outras fatores:

- Da estrutura molecular do material que constitui o resistor, ou seja, quanto maior a compactação do material, maior será a dificuldade oferecida a passagem da corrente elétrica.
- Da Temperatura do resistor, ou seja da agitação molecular das partículas deste resistor.
- Do número de elétrons livres (portadores de carga elétrica) que existem neste condutor.



**Definiremos como resistência elétrica de um componente elétrico (resistor) a relação entre a d.d.p. ( V ) aplicada aos seus terminais e a corrente elétrica ( i ) que o percorre.**

$$R = \frac{V}{i} \quad \text{ou} \quad \boxed{V = R i}$$

**UNIDADES:** No S. I. teremos o Volt/ampère, denominado **ohm (  $\Omega$  )**

$$1\Omega = \frac{1V}{1A}$$

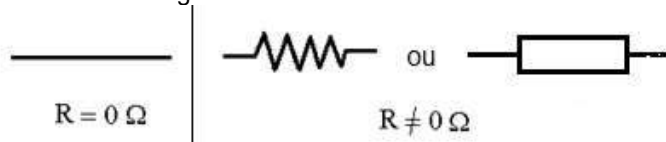
Assim, o valor da resistência elétrica do filamento de cada uma das lâmpadas do exemplo mostrado anteriormente será:

a) Para Lâmpada 1:

b) Para Lâmpada 2:

### 5.1. Representação esquemática dos resistores nos circuitos elétricos

Nos circuitos elétricos, um condutor que tenha resistência elétrica desprezível (como é o caso dos fios condutores) pode ser representado por uma linha contínua em caso contrario representaremos os resistores por uma linha poligonal ou um pequeno retângulo, como mostrado na figura abaixo.



#### a. Resistores para uso em circuitos eletrônicos

Nos circuitos eletrônicos a principal função dos resistores é distribuir e controlar a **corrente** e a **tensão** em partes específicas deste circuito. Para facilitar a identificação destes componentes adotou-se um **código de cores** que nos fornece o valor numérico em Ohms da resistência deste resistor,

Cada resistor possui quatro faixas de cores, as três primeiras faixas estão relacionadas ao valor de sua resistência deste resistor e a quarta faixa a tolerância desta medida. Veja o esquema mostrado a seguir:



$R = 5600 \, \Omega, \pm 1\%$ $R = 5,6 \cdot 10^3 \, \Omega$ $R = 5,6 \, K\Omega$				
COR	1ª Faixa (Número)	2ª Faixa (Número)	3ª Faixa (zeros ou Pot. de 10)	4ª Faixa (Tolerância)
Preto	—	0	—	—
Marrom	1	1	0 ( $\times 10^1$ )	1%
Vermelho	2	2	00 ( $\times 10^2$ )	2%
Laranja	3	3	000 ( $\times 10^3$ )	—
Amarelo	4	4	0000 ( $\times 10^4$ )	—
Verde	5	5	00000 ( $\times 10^5$ )	—
Azul	6	6	000000 ( $\times 10^6$ )	—
Violeta	7	7	—	—
Cinza	8	8	—	—
Branco	9	9	—	—
Ouro	—	—	$\times 0,1$ ( $\times 10^{-1}$ )	5%
Prata	—	—	$\times 0,01$ ( $\times 10^{-2}$ )	10%
<b>Código de cores dos resistores</b>				

**Exercício 1:** Utilizando o código de cores mostrado na figura acima, encontre o valor e a tolerância para cada um dos resistores apresentados na tabela abaixo:

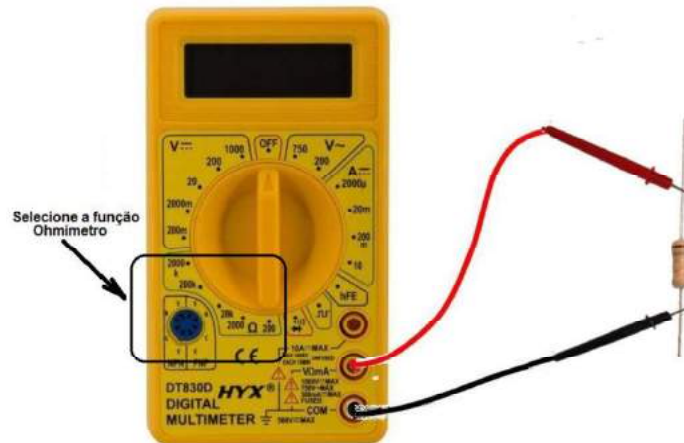
Resistor	Valor da Resistencia em Ohm ( $\Omega$ )	Tolerância da medida
a)	R= _____	_____
b)	R= _____	_____
c)	R= _____	_____
d)	R= _____	_____



**OBS: Você poderá medir a resistência elétrica de um resistor com um instrumento chamado Ohmímetro.**

Gire o botão para função **Ohmímetros**. Você deverá escolher uma das escalas de medida de acordo com o valor aproximado da resistência que deseja medir.

Obs.: Os Ohmímetros não podem ser utilizados para medir componentes que estejam conectados em um circuito elétrico ou em qualquer fonte de tensão.



**Atividade 1:** Encontre o valor da resistência e tolerância para cada uma dos resistores fornecidos pelo professor, e meça o valor com o Ohmímetro. Calcule o erro

Resistor	Valor da Resistência em Ohm ( $\Omega$ ) e tolerância (%) (utilizando o código de cores)	Valor da Resistência em Ohm ( $\Omega$ ) (medida com Ohmímetro)	Erro percentual entre valor especificado e valor medido (%)
a)	R= _____ e tolerância de _____ %	R= _____	_____
b)	R= _____ e tolerância de _____ %	R= _____	_____

**Atividade 2:** Encontre o valor da resistência do filamento de uma lâmpada:

a) com ela desligada e b) com ela em ligada (em funcionamento)

**Atividade 3:** Para o resistor fornecido pelo professor encontre o valor de sua resistência elétrica :

a) indicado no código de cores,

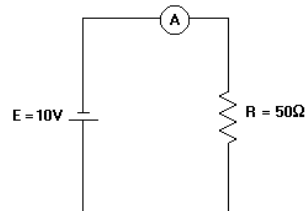
b) medido com um Ohmímetro.

c) quando estiver ligado em uma d.d.p. de aproximadamente 3,0 V.

**Exercícios:**

1) No circuito a seguir, indique qual será a leitura do amperímetro e qual o sentido convencional da corrente elétrica?

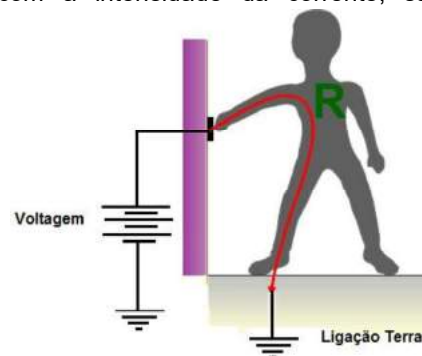
- a)  $I = 0,2 \text{ A}$   
 b)  $I = 10 \text{ A}$   
 c)  $I = 5 \text{ A}$   
 d)  $I = 2 \text{ A}$   
 e)  $I = 500 \text{ A}$



2) Nos choques elétricos, as correntes que fluem através do corpo humano podem causar danos biológicos que, de acordo com a intensidade da corrente, são classificados segundo a tabela a seguir.

	Corrente elétrica	Dano biológico
I	Até 10 mA	Dor e contração muscular
II	De 10 mA até 20 mA	Aumento das contrações musculares
III	De 20 mA até 100 mA	Parada respiratória
IV	De 100 mA até 3 A	Fibrilação ventricular que pode ser fatal
V	Acima de 3 A	Parada cardíaca, queimaduras graves

DURAN, J. E. R. *Biofísica - fundamentos e aplicações*. São Paulo: Pearson Prentice Hall, 2003. p. 178. [Adaptado]



Considerando que a resistência do corpo em situação normal é da ordem de  $1500 \Omega$ , em qual das faixas acima se enquadra uma pessoa sujeita a uma tensão elétrica de 220V?

- a) I      b) II      c) III      d) IV      e) V

3) Um resistor ôhmico é percorrido por uma corrente de 200 mA quando submetido a uma d.d.p. de 1,5 V.

a) que corrente o percorrerá quando ligado em 6V, se o valor de sua resistência permanecer constante?

b) Que d.d.p. (Vtagem) devermos aplicar aos seus terminais para que ele seja percorrido por uma corrente de 1,0 A, se o valor de sua resistência permanecer constante?

4) Um resistor está sendo percorrido por uma corrente  $i_1$ . Se substituirmos por outro, de resistência quatro vezes maior, e mantivermos a mesma d.d.p. nos seus terminais, a corrente  $i_2$  que o percorrerá será?

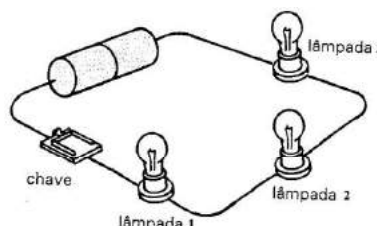
- a)  $i_2 = i_1$       b)  $i_2 = \frac{1}{2} i_1$       c)  $i_2 = \frac{1}{4} i_1$       d)  $i_2 = 2 i_1$       e)  $i_2 = 4 i_1$

## 6. Associação Série e Paralelo

Como já vimos anteriormente, podemos por exemplo acender uma ou mais lâmpadas (resistores) ao mesmo tempo em uma mesma bateria. Verificamos que dependendo do modo como ligamos, estas lâmpadas acendem com diferentes intensidades. Neste capítulo vamos investigar que algumas características dos circuitos são dependentes da forma como seus componentes são associados.

### a. Associação de resistores em Série

Dizemos que dois ou mais resistores estão associados em série quando na montagem do circuito estabelecermos um único caminho para a passagem da corrente elétrica. Assim, toda a corrente elétrica que passar por um resistor deve passar também pelos outros resistores.



#### Atividade 4:

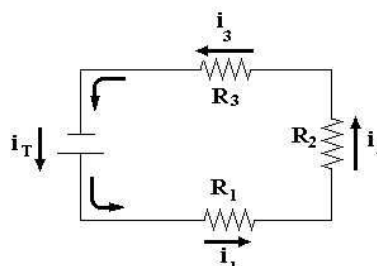
Juntamente com o seu grupo de estudo, investigue as características deste tipo de associação.

1º) Represente o esquema do circuito anterior utilizando os símbolos padronizados e monte o circuito elétrico.

2º) Represente no esquema ao lado, como devemos ligar os amperímetros para medirmos a corrente elétrica que passa pela bateria e por cada uma das lâmpadas. Faça as medições e anote os resultados encontrados.

$$i_T = \underline{\hspace{2cm}}$$

$$i_1 = \underline{\hspace{2cm}} \quad i_2 = \underline{\hspace{2cm}} \quad i_3 = \underline{\hspace{2cm}}$$



3º) Meça também a diferença de potencial d.d.p. nos terminais de cada uma das lâmpadas e da bateria.

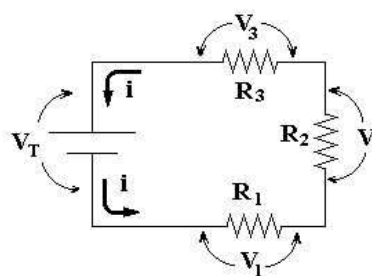
$$V_T = \underline{\hspace{2cm}}$$

$$V_1 = \underline{\hspace{2cm}} \quad V_2 = \underline{\hspace{2cm}} \quad V_3 = \underline{\hspace{2cm}}$$

**Características de uma ligação série:**

**a) Corrente elétrica →**

**b) Voltagem (d.d.p.) →**

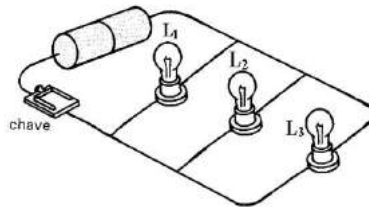


*O que caracteriza uma ligação em série é o fato de cada resistor ser percorrido pela mesma corrente elétrica (que neste caso é a corrente elétrica da bateria)*

*A d.d.p. total da associação é dividida entre os diversos resistores. Quem possuir a maior resistência elétrica terá em seus terminais a maior d.d.p. (Lembre-se que  $V=R.i$ , como o valor da corrente  $i$  será igual para todos os resistores, a maior d.d.p. será obtida nos terminais da maior resistência  $R$ )*

### b. Associação de resistores em Paralelo

Dizemos que dois ou mais resistores estão associados em paralelo quando na montagem do circuito todos os resistores estão submetidos a uma mesma d.d.p.

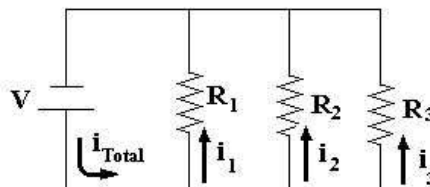


#### Atividade 5:

Gostaríamos agora que você e seu grupo analisassem as características de uma associação em paralelo.

1º) Represente o esquema do circuito anterior utilizando os símbolos padronizados e monte o circuito elétrico.

2º) Represente no esquema ao lado, como devemos ligar os amperímetros para medirmos a corrente elétrica que passa pela bateria e por cada uma das lâmpadas. Faça as medições e anote os resultados encontrados.



$i_T =$  \_\_\_\_\_

$i_1 =$  \_\_\_\_\_  $i_2 =$  \_\_\_\_\_  $i_3 =$  \_\_\_\_\_

3º) Meça a d.d.p. nos terminais de cada lâmpada e anote os resultados.

$V_T =$  \_\_\_\_\_

$V_1 =$  \_\_\_\_\_  $V_2 =$  \_\_\_\_\_  $V_3 =$  \_\_\_\_\_

#### Características de uma ligação em Paralelo:

a) Corrente elétrica →

b) Voltagem (d.d.p.) →

*O que caracteriza uma ligação em Paralelo é o fato de todos os resistores estarem submetidos a uma mesma d.d.p. (que neste caso é a d.d.p. da bateria)*

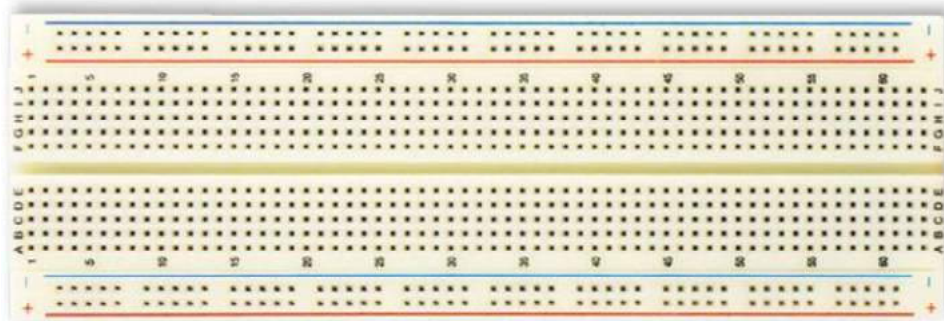
*A corrente elétrica total desta associação será a soma de todas as correntes elétricas que passam por cada um dos resistores. Neste caso o menor resistor desta associação será percorrido pela maior corrente elétrica. ( $i = V/R$ , como o valor da corrente  $V$  será igual para todos os resistores, a maior corrente  $i$  será obtida nos terminais da menor resistência  $R$ ).*

*Os circuitos necessários para a instalação elétrica de uma resistência são ligados em paralelo com o ponto de alimentação instalado pela companhia elétrica em nossas residências (Relógio de força). Isto garante que em todas as tomadas e pontos para ligação de equipamentos elétricos teremos uma mesma d.d.p.*

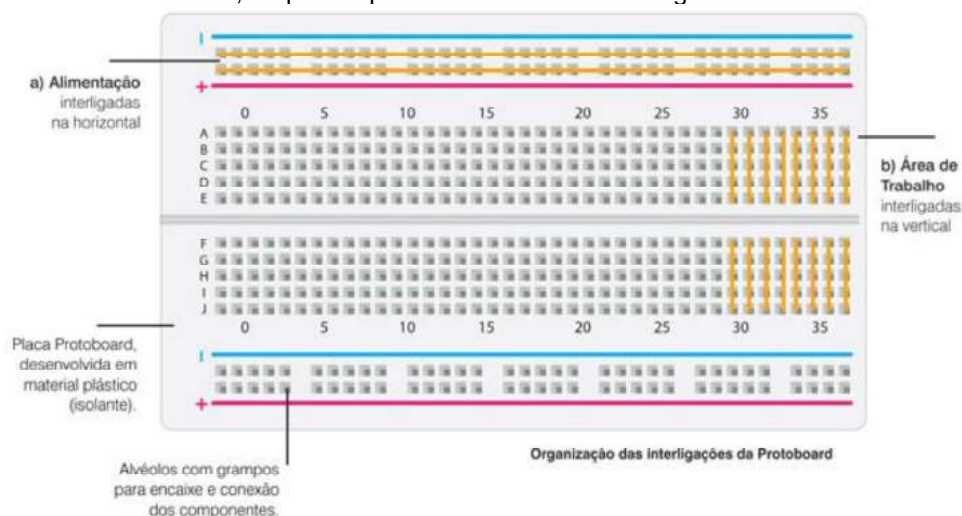
### c. Alguns componentes para facilitar a montagem dos circuitos com componentes eletrônicos

#### a) Protobord

A protobord é uma placa de plástico utilizada para montagem de circuitos contendo componentes eletrônicos. A utilização da protobord facilita a montagem e estruturação de novos projetos uma vez que estes componentes podem ser reaproveitados e desconectados facilmente, evitando a fase de soldagem dos destes componentes.



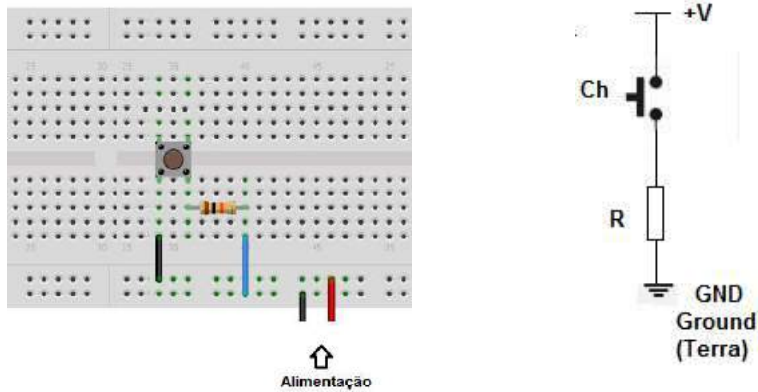
Ela é basicamente uma placa de plástico rígido, contendo pequenos furos que são conectados internamente. A figura abaixo mostra o modo como os furos são conectados internamente. Os furos nas extremidades superior e inferior são ligados entre si na horizontal, enquanto que as barras do meio são ligadas na vertical.



#### Atividade 6:

O circuito da figura abaixo mostra uma parte de uma placa protobord na um resistor será conectado uma fonte de alimentação por meio de uma chave Momentânea.

- Monte este circuito utilizando as pilhas como fonte de tensão
- Utilizando instrumentos de medida (Vôltemetro e um Amperímetro) efetue as medidas de corrente elétrica e tensão nos terminais deste resistor quando estiver em funcionamento
- Calcule o valor da resistência deste resistor quando ligado.
- Desconecte o resistor do circuito e meça com um Ohmímetro o valor de sua resistência.



**Atividade 7:** Utilizando uma fonte de tensão de 5V (Pode ser utilizada uma Placa Arduino), e a Proboborde, faça um esquema e monte o circuito serie e paralelo. Indique neste esquema como devem ser ligados o Voltímetro e o Amperímetro para medir os valores de tensão e corrente nos terminais de cada resistor.

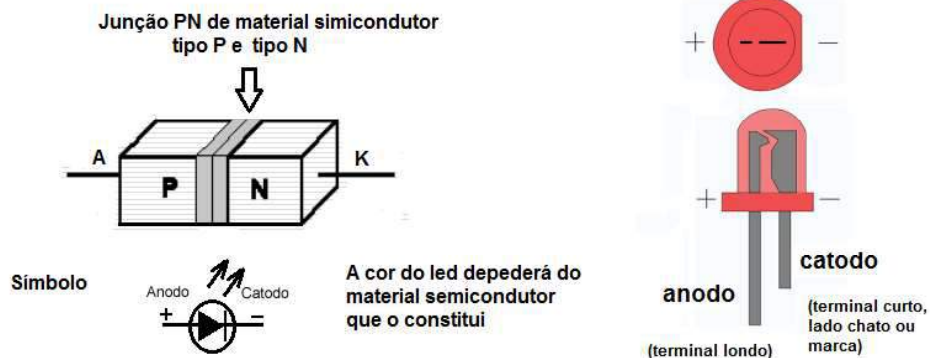
a) Para dois resistores associados em série

b) Para dois resistores associados em paralelo

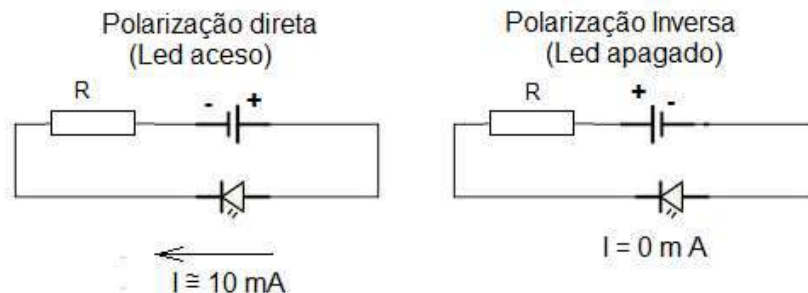
## B)LED

O LED, é um **Diodo Emissor de Luz**, (**Light Emitting Diode**). Ele possui a mesma tecnologia utilizada nos chips dos computadores, que tem a propriedade de transformar energia elétrica em luz.

Diodos são componentes eletrônicos **semicondutores** formados pela junção de dois tipos de materiais semicondutores que foram “dopados” (misturados com outros elementos para aumentarem o número de portadores de cargas em sua estrutura). Uma das mais importantes propriedades dos diodos é conduzir corrente elétrica em um sentido e bloqueá-la pelo outro.

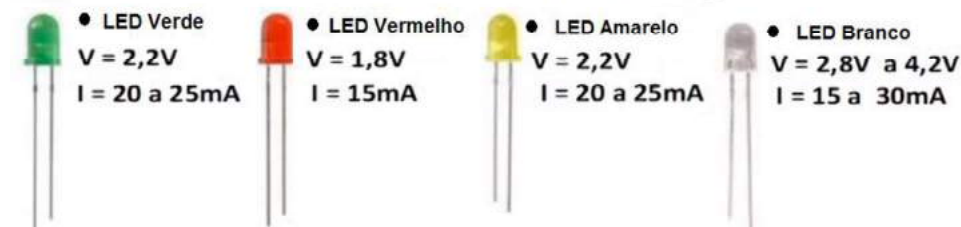


O LED como outros diodos possui um terminal chamado anodo e outro, chamado catodo. Dependendo de como for polarizado, permite ou não a passagem de corrente elétrica e, conseqüentemente, a geração ou não de luz.



Cada tipo de LED possui uma tensão e uma corrente indicadas para seu funcionamento. Abaixo são dados alguns exemplos aproximados destes valores. Valores de Tensão e Corrente superiores poderão danificar (queimar o seu LED)

### Tensão e Corrente de Alimentação



LEDs		
Cor do LED	Tensão em Volts (V)	Corrente em Milliampères (mA)
Vermelho	1,8V – 2,0V	20 mA
Amarelo	1,8V – 2,0V	20 mA
Laranja	1,8V – 2,0V	20 mA
Verde	2,0V – 2,5V	20 mA
Azul	2,5V – 3,0V	20 mA
Branco	2,5V – 3,0V	20 mA

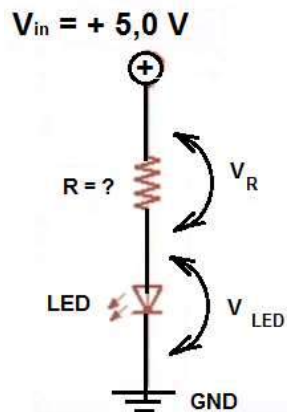
**d. Valor da Resistência R que deve ser ligada para proteger o LED**

Quando elaboramos uma projeto, devemos ter em mente que, dependendo do tipo LED que iremos utilizar e do valor da Tensão de alimentação ( $V_{in}$ ) de nosso circuito necessitaremos associar um resistor (**R**) para limitar a corrente e a tensão nos terminais do LED.

1)Seja por exemplo a seguinte situação: Vamos utilizar em nosso projeto uma fonte de tensão de entrada de 5,0 V ( $V_{in} = 5,0 \text{ V}$ ) e queremos acender um LED vermelho cuja tensão em seus terminais sabemos que deve ser ( $V_{LED} = 1,8 \text{ V}$ ) com corrente ( $I_{LEC} = 15 \text{ mA}$ ).

a)Como devemos ligar o Resistor que será utilizado para ligar este LED? Em série ou em paralelo? Monte o esquema desta ligação.

b)Qual deverá ser o valor da Tensão nos Terminais deste Resistor?



c)Qual deverá ser o valor desta Resistencia R?

2)Encontre o Valor de **R** para ligarmos um **Led Verde?** e para ligarmos um **Led Amarelo?** (considere que ambos funcionarão com uma corrente de 20 mA)

**Atividade 8:** Utilizando da fonte de alimentação de 5,0 V, (pode ser a alimentação de uma placa Arduino), uma protoborde e chaves momentâneas, monte um circuito para ligar 3 diferente LEDs (um vermelho, um amarelo e um verde). Faça o esquema e monte este circuito.



## **Anexo 2**

### **Trecho da apostila da Oficina 2**

## OFICINA II : Noções Iniciais sobre automação e Programação – Placa Arduino

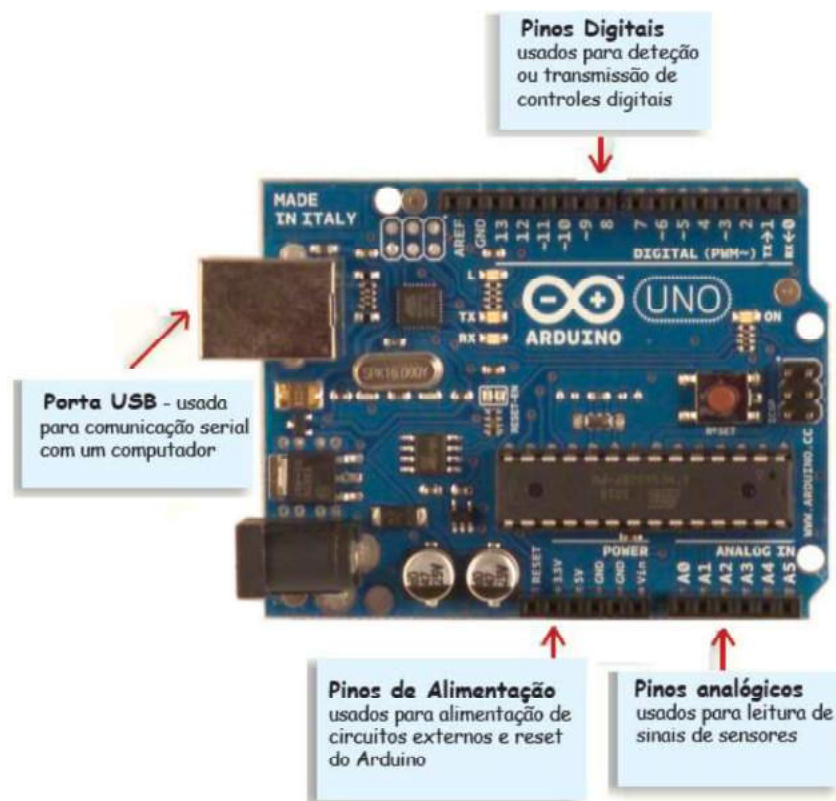
### 1. Introdução aos Circuitos de controle com Placa Arduino

#### 1.1 Introdução

Arduino é uma plataforma eletrônica criada na Itália para ser utilizada como plataforma de **prototipagem eletrônica**. Esta placa permite a automação de projetos eletrônicos e robóticos por profissionais e amadores (o projeto italiano iniciado em 2005 tinha primeiramente cunho educacional).

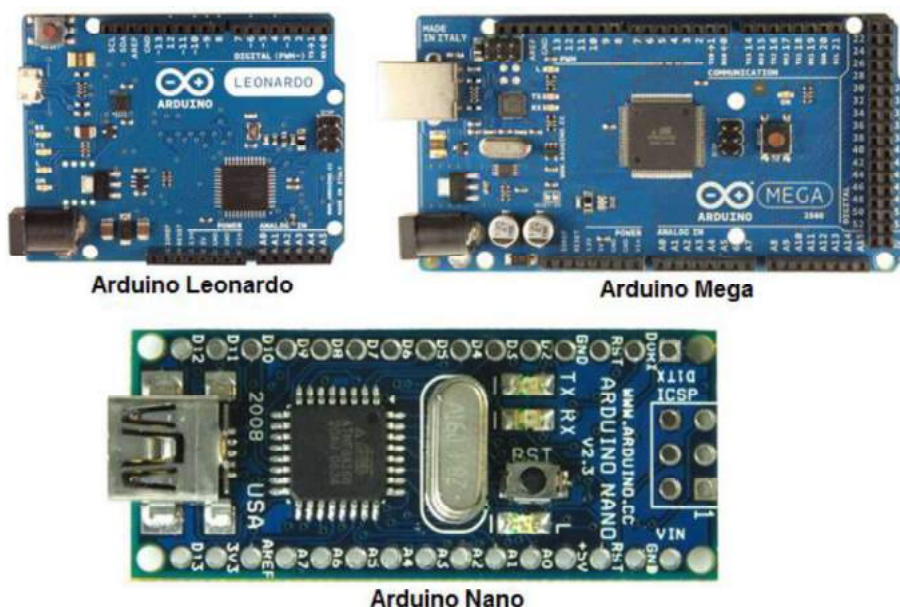
É constituído basicamente de uma placa microcontroladora, uma linguagem de programação com um ambiente de desenvolvimento e suporte para entrada e saída de dados e sinais.

Por não necessitar de um profundo conhecimento de eletrônica esta placa é bastante popular entre artistas e iniciantes, além de desenvolvedores experientes que não têm acesso a plataformas mais complexas.



Um fator que torna o Arduino atrativo é que ele é um **hardware livre**, ou seja, as pessoas podem usá-lo para criar diversos projetos sem custo algum de direitos pela utilização da plataforma. Isto traz diversos benefícios; além de serem criadas e distribuídas diversas novas bibliotecas e ferramentas para auxiliar o desenvolvimento de projetos todos os dias, conta-se com uma comunidade de milhares de pessoas que divulgam informações e detalhes sobre o que criam.

Existem diversos outros modelos de Arduino para se utilizar dependendo do que se deseja fazer, com diferentes formatos e configurações de hardware. O **Arduino Uno**, no entanto, é um dos mais utilizados. Seguem abaixo algumas imagens com alguns dos vários tipos de Arduino existentes hoje em dia.



## 1.2 Como nos comunicamos com o Arduino?

A placa Arduino que é basicamente composta de um microcontrolador juntamente com os circuitos de entrada/saída pode ser facilmente conectada a um computador e programada via **IDE** (Integrated Development Environment, ou **Ambiente de Desenvolvimento Integrado**). Este IDE deve estar previamente instalado no seu computador.

A linguagem que utilizamos para nos comunicarmos é formada por uma das mais populares linguagens utilizada para programação de computadores chamada de “linguagem C” e/ou “C++”. Como toda linguagem de programação ela é constituída de um conjunto de regras sintáticas que permite que o programador especifique precisamente sobre quais dados a plataforma irá atuar, como deverá analisar e processar estes dados e quais ações devem ser tomadas sob várias circunstâncias. O programador será na verdade como um professor que, utilizando de uma linguagem específica para ensinar a plataforma Arduino o “passo a passo” de como deverá realizar a atividade que desejamos.

A seguir apresentaremos algumas das funções e comandos que utilizaremos para programar na plataforma Arduino:

### 1.3 As principais funções da programação para plataforma Arduino.

#### a) **setup()**

Ao iniciarmos uma programação no Arduino, devermos primeiramente informar quais os pinos que utilizaremos em nosso projeto e se eles serão pinos de entrada ou saída de dados. Assim iniciamos qualquer programa do Arduino com esta função **setup()**, que em inglês significa configurar, estabelecer.

#### b) **pinMode(pino,Modo)**

É a forma que utilizamos para informarmos ao Arduino, quais serão os pinos utilizados em nosso projeto e se eles serão de entrada ou de saída. Ao utilizarmos esta função **pinMode(pino, modo)**, devemos substituir “*pino*” por um número inteiro (1, 2, 3, ...) que representará o número do pino da placa Arduino que vamos utilizar e “*modo*” deve ser substituído por “**INPUT**” se o pino for de entrada ou por “**OUTPUT**” se o pino for de saída de dados. Por exemplo, se damos um comando **pinMode(13, OUTPUT)** isso informará para o Arduino que o pino 13 será usada como pino de saída de sinal, ou seja, ele fornecerá um sinal elétrico para o circuito. (pode ser, por exemplo, um sinal para acender um led ou ligar um motor).

#### c) **loop()**

A função **loop()** é uma das funções mais importantes que iremos utilizar na construção de nossos projetos, pois é nela que iremos descrever tudo que o circuito irá fazer e como ele vai se comportar. Após a execução da função **setup()**, o programa passa a executar a função **loop()**, que deve conter todas as instruções que projetamos para ser executado e continuará a repeti-la até a execução do programa terminar.

#### d) **digitalRead(pino)** e **digitalWrite(pino, estado)**

Estas são funções utilizadas para operar com o que chamamos de “Portas Digitais”. No Arduino Uno, existem 14 portas digitais, numeradas de 0 a 13. Mais adiante voltaremos a especificar melhor os diferentes tipos de portas existentes na plataforma Arduino. As portas digitais são utilizadas para trabalhar com valores binários de tensão: 0V e 5V. Assim quando conectamos componentes a estas portas eles só poderão mandar e receber dados na forma destas duas tensões (0V e 5 V).

A função **digitalRead(pino)** lê um valor da porta pino e retorna o valor **HIGH** quando está em 5V e **LOW** quando está em 0V.

A função **digitalWrite(pino, estado)** é utilizada para dizermos ao Arduino que queremos que o pino esteja no estado ligado (**HIGH**) ou desligado (**LOW**).

Para entender melhor estes conceitos, vamos usar um exemplo simples para demonstrar o uso destas portas.

EXEMPLO 1: Vamos fazer um circuito simples no qual um LED deverá acender e apagar a cada segundo comandado pelo Arduino.

Neste exemplo vamos utilizar as funções e comandos vistos anteriormente.

A placa Arduino composta do micro controlador juntamente com os circuitos de entrada/saída pode ser facilmente conectada a um computador e programada via **IDE** (Integrated Development Environment, ou Ambiente de Desenvolvimento Integrado), sem a necessidade de equipamentos extras além de um cabo USB.

1º) Monte o circuito contendo um LED ligado a um pino do Arduino

Monte o circuito lembrando que a tensão de saída de uma porta digital do Arduino é de 5V que o nosso LED deve ser ligado a uma tensão máxima de 2,0V. Portanto será necessário utilizarmos um resistor para diminuir a tensão que será aplicada ao LED. (*Consulte os valores calculados para resistência no item 4.4 do da Oficina1*)

2º) Utilizando o cabo USB, conecte a placa Arduino ao seu computador e abra o Ambiente de desenvolvimento do Arduino.

Agora você poderá testar o funcionamento do seu circuito ligando o LED diretamente no pino de alimentação de 5 V da placa Arduino.

3º) Escreva o programa (**Sketch**) no Ambiente de Desenvolvimento do Arduino (IDE). Para isso, utilizaremos as funções e comandos necessários ao nosso projeto:

a) Primeiro de tudo devemos criar a função `setup()` que informará ao Arduino qual será o pino de saída que alimentará o nosso LED. Se escolhermos, por exemplo, o pino 13 como pino de saída, devemos digitar a função **`pinMode(13, OUTPUT)`** para informar que o pino 13 será utilizado como um pino de saída.

b) Em seguida, devemos criar a função `loop`, que é onde ficarão todos os comandos que vão ser executados repetidamente e indefinidamente pelo Arduino, como já foi dito.

→ `DigitalWrite(13, HIGH)` para ligar a tensão no pino 13;

→ `Delay(1000)` para fazer o programa esperar por 1000 ms (= 1,0s)

→ `DigitalWrite(13, LOW)` para desligar a tensão no pino 13

→ `Delay(1000)` para fazer o programa esperar por 1,0 s



```

Exemplo_1_-_acender_e_apagar_um_led

void setup() {
  pinMode(13, OUTPUT); // define o pino 13 como saída
}

void loop() {
  digitalWrite(13, HIGH); // Liga o LED
  delay(1000);             // aguarda 1,0 s
  digitalWrite(13, LOW);  // Desliga o LED
  delay(1000);             // aguarda 1,0 s
}
  
```

Carregado.

O sketch usa 1.030 bytes (3%) de espaço de armazenamento para prog  
 Variáveis globais usam 9 bytes (0%) de memória dinâmica, deixando

11 Arduino Uno en COM3

Obs. 1.: Quando estes comandos estão sendo digitados no Ambiente de Desenvolvimento do Arduino IDE, podemos colocar comentários para facilitar o entendimento do que está sendo feito. Estes comentários devem ser feitos do seguinte modo:

a) Para fazer um comentário que irá se desenvolver por mais de 1 linha, devemos usar os caracteres:

`/*` para começar um comentário de mais de 1 linha

`*/` para finalizar os comentários que foram feitos anteriormente

b) Para fazer um comentário em 1 linha apenas, podemos utilizar:

`//` para fazer um comentário de apenas 1 linha  
bs1.: Quando estes comandos estão sendo digitados na

4º) Por último temos que compilar o programa e fazer o upload para nossa placa Arduino.

Para compilar o programa devemos clicar no botão **Verificar** do Ambiente de Desenvolvimento para ver se não existe nenhum erro de código. O botão fica na barra de ferramentas.



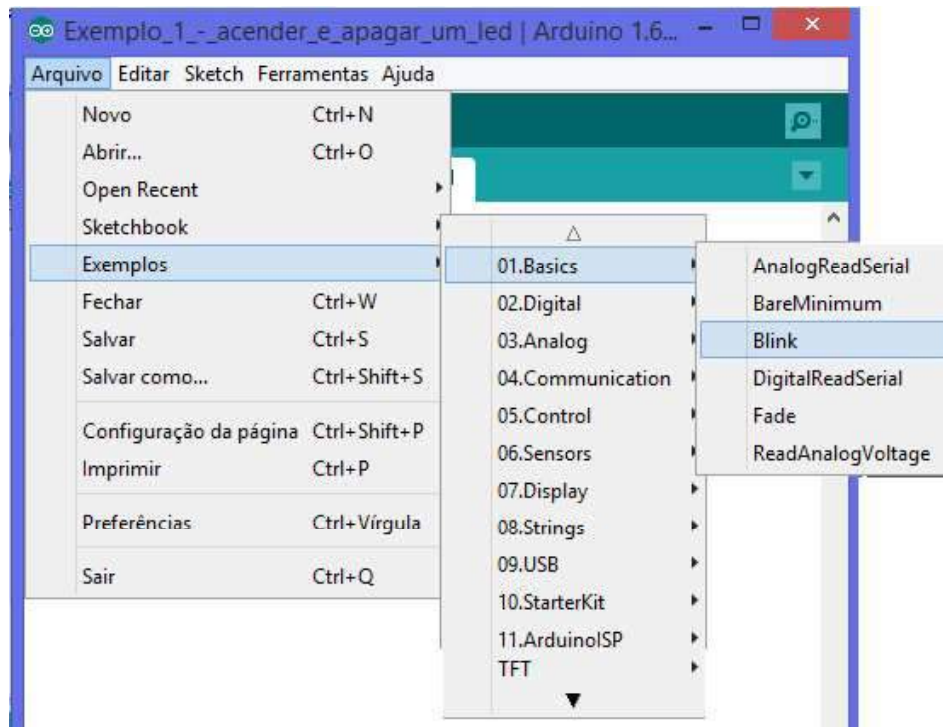
Se na barra inferior aparecer a mensagem: Done Compiling, o programa está pronto para ser enviado ao Arduino. Para tanto, basta clicar no botão **Upload** (Carregar) :



Espere então o upload ser completado e pronto. Você deverá ver o LED da placa piscando com intervalos de 1,0 segundo.

OBS 2: Este exemplo simples que chamamos BLINK (piscar) já está pronto no software de compilamento do Arduino. Para acessá-lo clique em

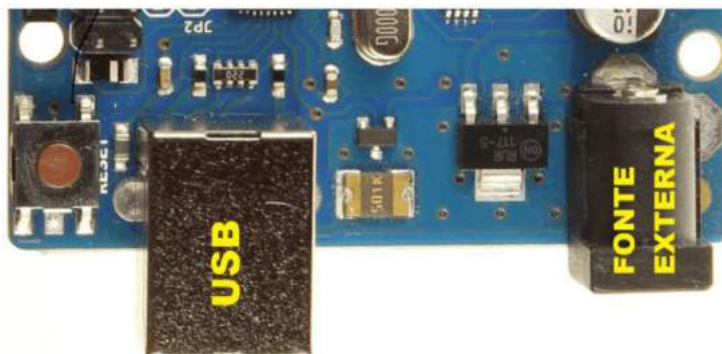
Arquivo > Exemplos > 1.BASICS > BLINK, como mostrado na figura abaixo:



Agora você já pode brincar um pouco com este exemplo. Procure modificar o tempo de acendimento do led e colocar outros para acenderem e apagarem em seqüências variadas.

#### 1.4 A alimentação da placa Arduino.

A placa Arduino pode ser alimentada diretamente pela conexão USB com o computador ou por uma fonte de alimentação externa, conforme exibido na figura abaixo:



A alimentação externa é feita através do conector Jack com positivo no centro, onde o valor de tensão da fonte externa deve estar entre os limites 6V a 20V, porém se alimentada com uma tensão abaixo de 7V, a tensão de

funcionamento da placa, que no Arduino Uno é 5V, pode ficar instável e quando alimentada com tensão acima de 12V, o regulador de tensão da placa pode sobreaquecer e danificar a placa. Dessa forma, é recomendado para tensões de fonte externa com valores de 7V a 12V.

**Atividade 1:** Agora você está pronto para construir seu primeiro projeto de acionamento: Juntamente com seus colegas, você deve elaborar um “protótipo de um sinal de trânsito” controlado por uma placa Arduino. Este protótipo de sinal deverá conter 3 LEDs: um vermelho, um amarelo e um verde.

a) complete o esquema de montagem do circuito colocando os LEDs Amarelo e Verde bem como colocando o valor do resistor que deverá ser ligado em série com o led.

b) monte este circuito em uma protoboard

c) Pesquise como deve funcionar a sequência de acendimentos dos LEDs e construa o sketch do programa no IDE.

**Diagrama Esquemático do Circuito**

