



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE CIÊNCIAS**  
**DEPARTAMENTO DE FÍSICA**  
**CURSO DE LICENCIATURA EM FÍSICA**

**ELISSANDRO AQUINO MENDES**

**USANDO O ARDUINO E A LINGUAGEM PYTHON NO ENSINO DE FÍSICA**

**FORTALEZA**

**2019**

ELISSANDRO AQUINO MENDES

USANDO O ARDUINO E A LINGUAGEM PYTHON NO ENSINO DE FÍSICA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Física do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do Título de Licenciado em Física.

Orientador: Prof. Dr. Ascânio Dias

FORTALEZA

2019

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- M49u Mendes, Elissandro Aquino.  
Usando o Arduino e a Linguagem Python no ensino de física / Elissandro Aquino Mendes. – 2019.  
52 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Ciências,  
Curso de Física, Fortaleza, 2019.  
Orientação: Prof. Dr. Ascânio Dias.
1. Ensino de Física. 2. Arduino. 3. Python. 4. Open source. I. Título.

CDD 530

---

ELISSANDRO AQUINO MENDES

USANDO O ARDUINO E A LINGUAGEM PYTHON NO ENSINO DE FÍSICA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Física do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do Título de Licenciado em Física.

Aprovada em: 19/06/2019

BANCA EXAMINADORA

---

Prof. Dr. Ascânio Dias (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Michael Ferreira de Souza  
Universidade Federal do Ceará - UFC

---

Danilo da Silva Borges, Msc  
Universidade Federal do Ceará - UFC

Aos meus Pais: Benedita e Antonio (In memoriam), pelo amor e apoio de toda uma vida...

## AGRADECIMENTOS

Ao final de um ciclo, como o marcado por uma entrega de monografia, sempre teremos várias pessoas a agradecer. Por vezes é até difícil lembrar de todos e corremos o risco de sermos algo injusto. Espero não incorrer nisso nessa breve nota de agradecimento.

Primeiramente agradeço à Deus pelo dom da Vida e pela Saúde que me acompanha e me fez capaz de vencer os vários obstáculos e chegar até aqui.

A meus Pais, *Benedita de Fátima e Antonio José*, pelo apoio e amor incondicional, pelos ensinamentos de vida repassados. Agradeço a minha esposa *Monica Cibelly*, pelo companheirismo e apoio de todas as horas, pelos incentivos e até pelos puxões de orelha, obrigado Vida.

Agradeço ao professor Ascânio Dias, pela orientação dada nesse trabalho e paciência comigo durante sua execução.

Gratidão a todos os professores do departamento que contribuíram direta ou indiretamente em minha formação.

Agradeço aos meus colegas de graduação pelos excelentes momentos de descontração, estudos e troca de conhecimento, em especial aos amigos: João Evangelista, Thiago Amorim, Thiago Guimarães, Thiago Probo, Francisco Etan, Francisco Hedler, Maxwell Filho, Italo Gonçalves, Diomaique Lopes, Nicolas Carvalho, Levi Félix, Charles Xavier, Adailton Azevedo, Adaias Souza.

“Jamais considere seus estudos como uma obrigação, mas como uma oportunidade invejável para aprender a conhecer a influência libertadora da beleza do reino do espírito, para seu próprio prazer pessoal e para proveito da comunidade à qual seu futuro trabalho pertencer.”

(Albert Einstein)

## RESUMO

Desde a popularização dos computadores e o aumento de sua presença em nosso dia a dia, educadores vem empreendendo pesquisas buscando mais e melhores meios de incluí-los nos processos de ensino-aprendizagem de Física. Neste trabalho realizamos uma apresentação, à luz da bibliografia existente, das diversas formas de uso dos computadores no ensino de Física fixando nossa maior atenção na coleta de dados em tempo real e na modelagem computacional, identificando suas vantagens e desvantagens e dando exemplos de uso. Além do computador em si, faremos uso de hardware e software Open Source de apoio mostrando ser possível construirmos uma melhor experiência do aluno com a disciplina, exercitando certa interdisciplinaridade sem cairmos em custos fora da realidade de muitas de nossas escolas. Apresentaremos a plataforma Arduino mostrando sua versatilidade no uso em experimentos que envolvam o controle, coleta e análise de dados em tempo real. Também será apresentada a linguagem de programação Python utilizada tanto em conjunto com o Arduino, para visualização de dados em tempo real, quanto sozinha na implementação e visualização de soluções numéricas de equações diferenciais do movimento harmônico simples.

**Palavras-chave:** Ensino de Física. Hardware. Software. Open Source. Arduino. Python.



## ABSTRACT

With the popularity of computers and the increase of its presence in our daily lives, educators has been undertaking research seeking more and better ways to include them in the teaching-learning process of Physics. In this work, based on existing literature, we make an apresentation of many ways to use computers to the teaching physics, giving more attention on data capture in real time and computational modeling, identifying advantages and disadvantages, giving examples and suggesting approaches to use in the classroom. Besides the computer itself, we'll make use of open source hardware and software showing be possible build a better experience with discipline by the alumni without falling into expenses beyond capacity of our schools. We present the Arduino platform showing its versatility of use in experiments by involving control, capture and analisys of data in real time. Also we present the programming language Python in use alone and in conjuction to Arduino to support visualizations of images and numerical solutions of simple harmonic oscillator.

**Keywords:** Physics Teaching . Hardware. Software. Open Source. Arduino. Python.

## LISTA DE FIGURAS

Figura 1 – Primeiro IBM-PC lançado em 1981 . . . . .	14
Figura 2 – Macintosh by Apple 1984 . . . . .	14
Figura 3 – Exemplo de placas Arduino . . . . .	19
Figura 4 – Arduino com shield Ethernet usado para montar um WebServer . . . . .	19
Figura 5 – Esquema de forças no pêndulo simples . . . . .	23
Figura 6 – Solução de $x \frac{dy}{dx} = -y$ analítica e por Runge-Kutta 1ª ordem . . . . .	29
Figura 7 – Solução de $x \frac{dy}{dx} = -y$ analítica e por Runge-Kutta 2ª ordem . . . . .	30
Figura 8 – Solução do pêndulo simples por Runge-Kutta 1ª ordem . . . . .	31
Figura 9 – Solução do pêndulo simples por Runge-Kutta 2ª ordem . . . . .	31
Figura 10 – Solução do pêndulo simples por Runge-Kutta 2ª ordem - Tempo estendido . . . . .	32
Figura 11 – Visão esquemática . . . . .	33
Figura 12 – Montagem no protoboard . . . . .	34
Figura 13 – Montagem definitiva . . . . .	34
Figura 14 – Leitura serial em tempo real . . . . .	35

## **LISTA DE ABREVIATURAS E SIGLAS**

- D.D.P Diferença de potencial
- LDR Do inglês: Light Dependent Resistor - Resistor dependente da luz
- LED Do inglês: Light Emissor Diode - Diosdo emissor de luz
- RK1 Runge-Kutta de 1a. Ordem
- RK2 Runge-Kutta de 2a. Ordem

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>12</b>
<b>2</b>	<b>O COMPUTADOR NO ENSINO DE FÍSICA</b> . . . . .	<b>14</b>
<b>2.1</b>	<b>Um pouco de história</b> . . . . .	<b>14</b>
<b>2.2</b>	<b>Teorias de aprendizagem</b> . . . . .	<b>15</b>
<b>2.3</b>	<b>Algumas formas de uso</b> . . . . .	<b>16</b>
<b>2.3.1</b>	<i>Aquisição de dados por computador</i> . . . . .	<b>17</b>
<b>2.3.2</b>	<i>Modelização e simulação</i> . . . . .	<b>17</b>
<b>3</b>	<b>APRESENTANDO AS FERRAMENTAS: ARDUINO &amp; PYTHON</b> . . . . .	<b>18</b>
<b>3.1</b>	<b>Arduino: Uma plataforma de hardware/software Open Source</b> . . . . .	<b>18</b>
<b>3.2</b>	<b>Python: Uma linguagem de uso geral</b> . . . . .	<b>19</b>
<b>4</b>	<b>SIMULAÇÕES E EXPERIMENTOS</b> . . . . .	<b>22</b>
<b>4.1</b>	<b>Experimento 1 - Resolvendo numericamente o pêndulo simples</b> . . . . .	<b>22</b>
<b>4.1.1</b>	<i>Modelando o pêndulo simples</i> . . . . .	<b>22</b>
<b>4.1.2</b>	<i>Métodos numéricos</i> . . . . .	<b>25</b>
<b>4.1.3</b>	<i>Validações dos métodos numéricos</i> . . . . .	<b>28</b>
<b>4.1.4</b>	<i>Discussão e abordagens</i> . . . . .	<b>32</b>
<b>4.2</b>	<b>Experimento 2 - Visualizando oscilações amortecidas</b> . . . . .	<b>32</b>
<b>4.2.1</b>	<i>Detalhes do experimento</i> . . . . .	<b>33</b>
<b>4.2.2</b>	<i>Aplicações e Resultados</i> . . . . .	<b>35</b>
<b>4.2.3</b>	<i>Discussões e abordagens</i> . . . . .	<b>36</b>
<b>5</b>	<b>CONCLUSÕES</b> . . . . .	<b>37</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>38</b>
	<b>APÊNDICES</b> . . . . .	<b>40</b>
	<b>APÊNDICE A</b> – Métodos Runge-Kutta 1 <sup>a</sup> e 2 <sup>a</sup> ordem para solucionar $x \frac{dy}{dx} + y = 0$ . . . . .	<b>40</b>
	<b>APÊNDICE B</b> – Métodos Runge-Kutta 1 <sup>a</sup> e 2 <sup>a</sup> ordem para o pêndulo simples . . . . .	<b>44</b>
	<b>APÊNDICE C</b> – <b>Leitura serial Arduino</b> . . . . .	<b>49</b>
	<b>APÊNDICE D</b> – <b>Leitura serial</b> . . . . .	<b>50</b>

## 1 INTRODUÇÃO

Sabemos da dificuldade que muitos estudantes apresentam na assimilação de conteúdos que requerem um maior grau de abstração, como o são, os conteúdos de Física e Matemática. Para Rocha (2013), os atuais índices de qualidade da educação no Brasil bem revelam que os estudantes não estão aprendendo o mínimo exigido, refletindo a preocupante situação na qual os espaços escolares tornam-se desinteressantes, e como consequência, os alunos cada vez mais abandonam seus cursos ou passam por esses ambientes sem serem exigidos a contento, tornando-se adultos analfabetos funcionais distantes da cidadania plena.

Essa falta de atratividade tem causas das mais diversas, mas com certeza incluem como principais: o despreparo e desmotivação dos professores, a carência de recursos de infraestrutura para as escolas, a ausência de renovação das práticas pedagógicas, falta de currículos mais atualizados e a escassa elaboração de materiais educacionais consistentes com as novas formas de aprender e ensinar (BRASIL, 2010).

Focando no currículo de Física, temos nas práticas de laboratório um ponto importante na formação dos estudantes; ponto esse que segundo Séré, Coelho e Nunes (SERÉ, 2003) tem destacado papel no estímulo ao estudante em não permanecer somente no mundo dos conceitos e das "linguagens"(natural, matemática e simbólica), tendo assim, a possibilidade de relacionar esses mundos com o empírico. Tornando a experiência mais real e motivante, dando sentido ao mundo abstrato e formal das "linguagens".

Ainda segundo Seré (2003), as práticas de laboratório (em suas diversas formas de abordagem) permitem: controle do meio ambiente, a autonomia face aos objetos técnicos, ensinam as técnicas de investigação e possibilitam um olhar crítico sobre o resultado. O aluno só conseguirá questionar o mundo, manipular os modelos e desenvolver os métodos se ele mesmo entrar nessa dinâmica de decisão, de escolha, de inter-relação entre a teoria e o experimento.

É para auxiliar nesse questionamento que neste trabalho buscamos compreender e exemplificar a utilidade e a aplicabilidade do computador como meio de apoio das práticas de laboratório, buscando assim melhorar a compreensão qualitativa e quantitativa dos fenômenos físicos envolvidos. Incluiremos uma pitada de interdisciplinaridade, fazendo uso de equipamentos e linguagens de programação, ditos Open Source, e portanto sem altos custos, buscando aliviar um dos pontos de dificuldade na inclusão efetiva do computador no processo de ensino/aprendizagem em nossas escolas, que é o custo dos experimentos.

Para atingir tal intento, realizamos pesquisa bibliográfica em periódicos, livros, dissertações e teses que se refiram aos tópicos dessa monografia, sendo tais referências utilizadas na composição dos capítulos.

No capítulo 2, faremos uma pequena resenha histórica do aparecimento dos computadores a que se seguirá uma exposição resumida das suas diversas formas de uso no processo de ensino/aprendizagem relacionando suas vantagens e desvantagens, porém, nos deteremos principalmente na coleta de dados em tempo real e na modelagem computacional que servirão de base aos experimentos.

No capítulo 3 faremos uma breve apresentação dos materiais de apoio que usaremos na implementação dos experimentos, quais sejam, a placa micro-processada Arduino, que será usada na coleta em tempo real dos dados e a linguagem de programação Python, com a qual faremos a análise e visualização dos dados e também implementaremos uma solução numérica do pêndulo simples e amortecido.

No capítulo 4, teremos os experimentos; um deles será a modelagem do pêndulo simples e a resolução de sua equação diferencial por meio de métodos numéricos. Também faremos uma visualização de oscilações amortecidas em uma régua demonstrando o uso da coleta de dados em tempo real com o Arduino. No capítulo teremos seções onde faremos digressões teóricas sobre os aspectos físicos e matemáticos relevantes e também iremos expor os detalhes de implementação e execução dos experimentos, fornecendo em anexo os códigos fontes dos programas desenvolvidos.

## 2 O COMPUTADOR NO ENSINO DE FÍSICA

### 2.1 Um pouco de história

Os primeiros computadores remonta aos anos 40-50, porém somente no final da década de 70 surgiram os primeiros computadores pessoais. Eles revolucionaram e democratizaram o uso de computadores. Tendo o primeiro computador pessoal surgido em 1979, logo outros vieram. Gigantes como a IBM, pouco tempo depois em 1981, introduziam o seu hoje famoso computador pessoal, o *IBM-PC* que chegou em 1982 a ser considerada a máquina do ano pela revista *Time* (FIOLHAIS; TRINDADE, 2003).

Figura 1 – Primeiro IBM-PC lançado em 1981



Fonte: <[http://www.sabbatini.com/renato/correio/ibm\\_pc\\_81.jpg](http://www.sabbatini.com/renato/correio/ibm_pc_81.jpg)>

Como nos relata, Fiolhais e Trindade (2003), em 1984 outra gigante entra em cena, A *Apple Computers*, lançando um computador pessoal recheado de novidades e adendos visando a facilidade de uso, incorporando as interfaces gráficas e o uso do mouse, o *Macintosh*. Em seguida vieram os sistemas operacionais gráficos da *Microsoft* para IBM-PC como resposta ao lançamento da *Apple*. Assim estavam lançadas as bases para a total democratização dos computadores pessoais.

Figura 2 – Macintosh by Apple 1984



Fonte: <<http://oldcomputers.net/pics/macintosh.jpg>>

Com as facilidades de uso em crescente expansão, assim como o aumento do poder computacional, surgiram os primeiros pioneiros que vislumbraram o uso de tais máquinas no processo de ensino-aprendizagem, como nos conta Fiolhais e Trindade (2003), o exemplo do físico norte-americano Alfred Bork que numa conferência intitulada *Aprendizagem Interativa*, chegou a profetizar que o computador configurava uma revolução e que até o ano 2000 seria através do dele a principal forma de aprendizado em todos os níveis e quase todas as áreas.

Também devemos frisar o aparecimento da Internet nos anos 80 e sua popularização nos anos 90 como altamente impactante na forma como a informação passou a ser distribuída e como passou a se dar o uso dela, e dos vários artefatos de ensino que passaram a ficar disponíveis através da rede. Mas recentemente temos a febre no uso de smartphones e tablets que elevaram ainda mais a quantidade de meios onde os educadores podem disponibilizar os mais variados objetos e experimentos para auxiliar na aprendizagem como podemos ver nos exemplos em Vieira e Lara (2013) e Vieira *et al.* (2014).

## 2.2 Teorias de aprendizagem

Para balizar e orientar o uso do computador nos processos de ensino-aprendizagem tem-se duas teorias básicas: uma de base behaviorista e outra com base construtivista.

Segundo Magalhães *et al.* (2002), na visão behaviorista o professor é o controlador do ambiente. Sendo as aulas planejadas e organizadas de modo que passo a passo o aluno participe de uma lógica de estímulo-resposta-reforço para atingir um objetivo.

Com isso se busca mudar o comportamento do aluno pela via do condicionamento, com o ambiente de aprendizagem rigidamente estruturado para que o aluno possa obter as mudanças desejáveis que seria o aprendizado. Isso é fundamentado no fato de que segundo essa teoria, como nos lembra Fiolhais e Trindade (2003), nossa mente é uma "caixa preta" que dá respostas aos estímulos e esses podem ser medidos, independente dos processos mentais que ocorrem no seu interior.

Com esses pressupostos em mente, algumas formas de uso do computador, usando o enfoque de aprender com o computador, foram bastante difundidos como nos relata Magalhães *et al.* (2002):

- **Exercício e prática:** Muito usadas nos anos 70 e 80 e ainda hoje, por exemplo, no aprendizado de línguas, em cursos nas empresas e no aprendizado da computação. Baseia-se na visão behaviorista de Skinner de reforço de associações de estímulo resposta;



- **Tutorial:** Tendo com limitação que os alunos adquiriam um conhecimento pronto. Os programas tutoriais visavam responder às diferenças individuais fornecendo uma instrução de reforço quando os alunos respondiam incorretamente;
- **Tutoriais Inteligentes:** Foram desenvolvidos nos anos 80 e 90, sendo adicionadas aos tutoriais o uso de técnicas de inteligência artificial para ensinar a resolver problemas e adquirir conhecimento a partir de modelos especialistas que descreveriam a forma como um especialista resolveria o problema.

Na outra ponta, a busca do entendimento dos processos mentais que estariam na base do comportamento dos alunos e seriam indicadores dos processos que estão a se desenrolar em suas mentes, formam a base da teoria cognitiva. Desenvolvida, no final da década de 50, pelo suíço Jean Piaget, dentre outros, só no final da década de 70 começou a ter uma influência mais efetiva nas formas de ensinar (FIOLHAIS; TRINDADE, 2003).

A grande premissa da teoria ou psicologia cognitiva, base do ensino construtivista, é a de que a educação é inseparável da vida. O aluno desenvolve a sua própria imagem, criando conhecimento fundamentado em conhecimentos e habilidades anteriores. O professor passa de transmissor do conhecimento para uma papel de monitoração do crescimento cognitivo e do amadurecimento do aluno. O aprendizado passa a ser entendido como um processo ativo e colaborativo onde o aluno gerencia os recursos cognitivos disponíveis para criar um novo conhecimento extraindo informações do ambiente e também buscando cooperação com outros indivíduos para mostrar múltiplas perspectivas. Com isso em mente a ênfase passou a ser nos conteúdos e na apresentação aos alunos mediante o uso do computador. A forma individualizada com que se passou a enxergar os alunos como partícipes ativos do processo de aprendizagem e o fato que as diferenças entre eles não podem ser ignoradas conduziu a uma melhoria na utilização dos computadores. Tem-se aí um primeiro passo na direção da educação baseada no respeito a individualidade (MAGALHÃES *et al.*, 2002).

### 2.3 Algumas formas de uso

Diversas são as formas de uso do computador no processo de ensino-aprendizagem, como podemos ver em Fiolhais e Trindade (2003). Dentre as listadas damos uma breve resenha das que achamos mais relevantes para o momento atual e presente trabalho.

### 2.3.1 Aquisição de dados por computador

O laboratório tem uma relevância central na aprendizagem da Física, lá o computador encontrou um lugar ideal para participar desse processo. Dorneles *et al.* (2012), nos fala que a execução de atividades computacionais conjuntamente às atividades de laboratório, utilizando uma metodologia didática do tipo *Predizer, interagir e explicar* levou a uma melhora da aprendizagem gerando respostas menos lacônicas e aumentando o nível de compreensão dos alunos.

Com o advento de softwares e sensores dos mais diversos tipos pode-se hoje controlar e medir as mais diversas variáveis como posição, temperatura, velocidade, aceleração, força, etc. Possibilitando assim novas situações de aprendizagem permitindo medições e visualizações em tempo real que fornecem respostas imediatas a questões previamente formuladas (FIOLHAIS; TRINDADE, 2003).

### 2.3.2 Modelização e simulação

Como nos relata Elias *et al.* (2009), a modelização e a simulação são as duas formas de maior uso do computador no ensino de Física. Segundo Fiolhais e Trindade (2003), modelização está voltada as atividades de programar no computador o modelo do fenômeno físico, enquanto simulação tem uma ligação com um modelo tipo "Caixa Preta".

Na modelização o aluno é levado a criar "micromundos" e agir sobre eles modificando as equações do modelo ou parâmetros a elas relacionados. Ao usar simulações, que são baseadas numa modelo da realidade física, aos alunos é permitido alterar os valores das variáveis de entrada e a observação dos resultados (FIOLHAIS; TRINDADE, 2003).

Como exemplo de uma ambiente de modelagem muito interessantes, temos o *Modellus*<sup>1</sup>, onde o usuário pode interagir de diversas formas, inserindo o modelo matemático a ser executado e também criar um ambiente com vários objetos que responderão a esse modelo. Como exemplo de simulações computacionais podemos citar o projeto *PhET*<sup>2</sup>, que contém uma grande quantidade de simulações das mais variadas e também dá acesso a orientações pedagógicas de uso de tais simulações.

---

<sup>1</sup> <<http://modellus.fct.unl.pt/>>

<sup>2</sup> <<http://phet.colorado.edu/>>

### 3 APRESENTANDO AS FERRAMENTAS: ARDUINO & PYTHON

#### 3.1 Arduino: Uma plataforma de hardware/software Open Source

No experimento apresentado neste trabalho faremos uso da plataforma de hardware/software Open Source, *Arduino*. Como nos explica Cavalcante *et al.* (2011), a plataforma Arduino foi construída para facilitar e promover a interação física entre o ambiente e computador utilizando dispositivos eletrônicos simples e baseada em softwares e hardwares livres. Consistindo de uma placa com entradas e saídas para um microcontrolador ATMEL, um ambiente de desenvolvimento e o software de *bootloader*<sup>1</sup> que já vem gravado no microcontrolador. Apesar de existirem outras opções de plataformas construídas para microcontroladores o Arduino é um destaque mundial dada a facilidade de programação, a versatilidade e o baixo custo envolvido no seu uso (CAVALCANTE *et al.*, 2011).

No site da plataforma Arduino<sup>2</sup> encontramos uma lista com vários motivos para uso do Arduino frente as demais opções:

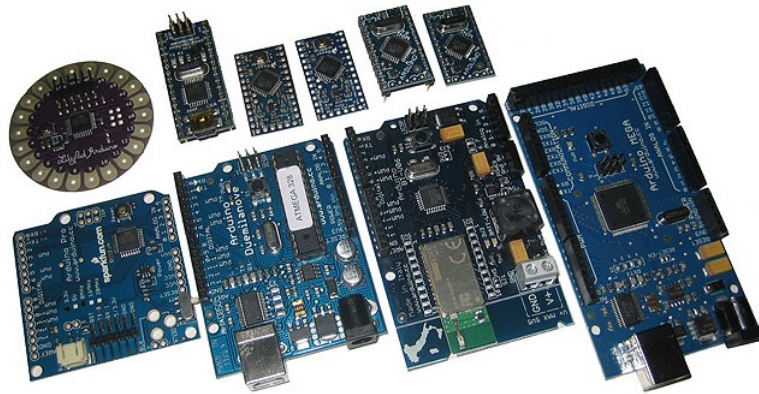
- **Custo:** Tem baixo custo, podendo ser comprado montado ou mesmo ser montado manualmente. As especificação estão disponíveis;
- **Multiplataforma:** O ambiente de desenvolvimento está disponível para várias plataformas: Windows, Mac OSX, Linux. Muitos microcontroladores são limitados ao Windows;
- **Programação simples e limpa:** A programação para Arduino é suficientemente simples para fácil acesso aos iniciantes e flexível o bastante para os mais avançados;
- **Software Open Source e extensível:** Permite a expansão das funcionalidades para usuários mais avançados mediante a inclusão de bibliotecas escritas em C++;
- **Hardware Open Source e extensível:** O Arduino é baseado na nos microcontroladores ATMEGA8 e ATMEGA168. Todas as informações necessárias para que usuários moderadamente experientes possam montar, por si próprios, o módulo estão disponíveis, possibilitando o aprimoramento do módulo;

São diversos tipos de placas visando atender o máximo de necessidades. Elas variam em número de portas analógicas/digitais disponíveis, em tamanho, forma de alimentação e programação. Abaixo vemos alguns modelos disponíveis:

<sup>1</sup> A boot loader, also called a boot manager, is a small program that places the operating system (OS) of a computer into memory. When a computer is powered-up or restarted, the basic input/output system (BIOS) performs some initial tests, and then transfers control to the master boot record (MBR) where the boot loader resides. Disponível em: <<http://goo.gl/CRa9AW>>. Acessado em: 10/05/2019

<sup>2</sup> Disponível em: <<http://goo.gl/F1ETmB>>, acessado em: 10/05/2019

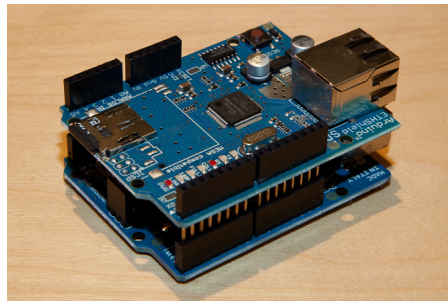
Figura 3 – Exemplo de placas Arduino



Fonte: <<http://www.robotshop.com/blog/en/arduino-robotics-projects-3666>>

Como pode ser visto temos modelos de vários tamanhos, cada um exibindo capacidades específica para atender as mais diversas necessidades. Há ainda a possibilidade de expansão com o uso dos *shields* que são placas adicionais, que podem se acoplar perfeitamente à placa principal ou ser conectada à parte para atender necessidade como: conexão WI-FI, conexão Ethernet, Bluetooth, comunicação GSM para chamadas ou envio de SMS, comunicação RF, driver para acionar dispositivos de potência, etc.

Figura 4 – Arduino com shield Ethernet usado para montar um WebServer



Fonte: <<http://www.webweavertech.com/ovidiu/weblog/archives/000484.html>>

O modelo usado no experimento é um Arduino UNO R3, um modelo simples, mas que permite a realização de vários experimentos interessantes. Há projetos e informações em abundância na Internet e também vários blogs e trabalhos científicos fazendo uso da plataforma.

### 3.2 Python: Uma linguagem de uso geral

Criada no final dos anos 80 pelo programador alemão, Guido van Rossum, que ainda hoje é um dos principais responsáveis pelas decisões envolvendo a linguagem, Python é uma linguagem de uso geral e permite o uso de múltiplos paradigmas de programação, a

saber: procedural, orientada a objeto e funcional. Ela hoje ocupa a 4ª posição no ranking Tiobe <sup>3</sup> que lista as linguagens de programação mais usada no mundo. Isso é reflexo de suas qualidades e força de seu ecossistema que é composto por várias bibliotecas com alto padrão de qualidade, fruto claro de uma comunidade muito forte e atuante, excelente documentação e ambiente de desenvolvimento criados especificamente para linguagem, permitem construir com relativa facilidade os mais variados tipos de aplicação.

Sendo uma linguagem interpretada e instalada como padrão em vários sistemas operacionais, provê uma grande facilidade de uso e hoje já é muito usada em universidades como primeira linguagem nos cursos de Ciência e Engenharia da Computação. A adoção pelas grandes empresas do setor como a Google, que até 2012 empregava o criador da linguagem, também é fator decisivo para esse boom no seu uso em praticamente todos os tipos de aplicações, de aplicativos comerciais web ou desktop, passando por jogos e softwares educacionais e chegando até a aplicações em computação de alto desempenho usando GPU's e nas várias áreas da Ciência de dados e inteligência artificial: aprendizagem de máquina, processamento de linguagem natural, visão computacional, redes neurais, etc.

Exemplo de alguns projetos usando Python como linguagem base:

- PyGame <<http://www.pygame.org>>
- kwant (Quantum transport simulations made easy) <<http://kwant-project.org/>>
- pyCUDA <<https://developer.nvidia.com/pycuda>>
- Introduction to Computer Science and Programming Using Python - MIT MOOC <<https://www.edx.org/course/introduction-computer-science-mitx-6-00-1x-0>>

No tocante ao suporte à programação existe uma vasta lista de bibliotecas disponíveis para os mais diversos fins. Como exemplos para uso científico podemos citar:

- Numpy - Computação científica <<http://www.numpy.org/>>
- matplotlib - Gráficos 2D com qualidade para publicação <<http://matplotlib.org/>>
- SymPy - Matemática simbólica <<http://www.sympy.org>>

---

<sup>3</sup> Disponível em: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>. Acessado em: 16/05/2016

Python sem dúvidas será uma linguagem cada vez mais usada pelo sua facilidade de uso e aprendizado e ecossistema disponível. É uma excelente escolha para uma primeira linguagem, projetos científicos e comerciais. Faremos uso de algumas bibliotecas citadas em nosso experimento de simulação numérica e na integração entre o computador e a plataforma Arduino será a cola a unir os dois.

## 4 SIMULAÇÕES E EXPERIMENTOS

Neste capítulo descrevemos alguns experimentos computacionais com ou sem coleta de dados que servirão de exemplo do potencial de uso de hardware e software livres no ensino de Física. Em cada experimento será dada uma breve descrição do fenômeno envolvido, do modelo matemático quando aplicável e do aparato de suporte ao experimento.

Todos os experimentos foram implementados pelo autor.

### 4.1 Experimento 1 - Resolvendo numericamente o pêndulo simples

Movimentos oscilatórios, harmônicos ou não, são muito presentes em nossas vidas. Facilmente podemos citar vários exemplos, alguns bem cotidianos: uma criança brincando num balanço no parque, os batimentos cardíacos, som de um violão, o som de um diapasão, as oscilações de um sistema massa-mola, oscilações de cargas indo e vindo em circuitos elétricos, etc.

Nosso primeiro experimento tratará da modelagem e solução numérica de um oscilador harmônico representado por um pêndulo simples idealizado.

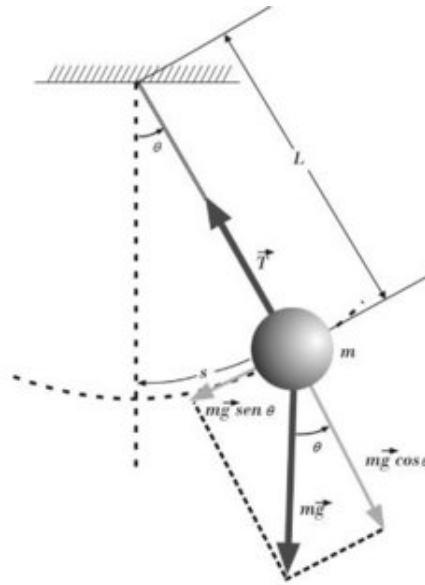
Vamos obter e resolver numericamente a equação diferencial que o modela. Implementaremos a solução usando Python e sua biblioteca gráfica, **matplotlib**.

Usaremos dois esquemas numéricos e para cada um analisaremos o comportamento temporal da energia e velocidade onde buscaremos critérios de como avaliar a resposta obtida em um experimento numérico, a saber, por meio da verificação de que se o comportamento fisicamente esperado se reflete no comportamento obtido a partir do resultado numérico.

#### 4.1.1 Modelando o pêndulo simples

O objeto central de nosso experimento serão as oscilações de um pêndulo simples idealizado. Dentre as simplificações que faremos está em considerar os comprimentos inextensíveis e desprezar quaisquer ações dissipativas sejam devidas ao atrito nas articulações ou a ação resistiva do ar.

Figura 5 – Esquema de forças no pêndulo simples



Fonte: <<http://www.scielo.br/img/revistas/rbef/v33n4/11f01.jpg>>

Na figura 5 temos um pêndulo constituído por uma haste rígida tendo uma massa soldada em uma extremidade enquanto a outra extremidade é mantida presa numa articulação que permita o conjunto mover-se livremente no plano vertical, ficando assim o movimento da massa limitado a um arco de circunferência tendo como raio o comprimento da haste. Dada as simplificações realizadas as únicas forças agindo sobre a massa serão seu peso e a tensão realizada pela haste na massa.

Para descrevermos o movimento do pêndulo, uma escolha interessante, como assinalado na figura 5, será a variação temporal do ângulo que a haste faz com a vertical.

Sendo a haste inextensível teremos um equilíbrio de forças na direção radial, em módulo:  $T = mg \cos \theta(t)$ . Na direção tangente à trajetória, a única força agindo é a componente da força peso,  $F_{tg} = mg \sin \theta(t)$ , responsável pela aceleração tangencial da massa.

Sabemos da segunda lei de Newton que:

$$F_{tg} = ma_{tg} = m \frac{dv}{dt} = -mg \sin \theta(t) \quad (4.1)$$

O sinal negativo indica a força agindo no sentido contrário ao movimento do pêndulo, uma *força de restauração*, buscando trazer o pêndulo de volta para a posição de equilíbrio estável, qual seja, o ponto mais baixo da trajetória.

Calculemos a aceleração tangencial da massa:

$$s(t) = L\theta(t) \quad (4.2)$$



$$v(t) = \frac{ds}{dt} = L \frac{d\theta}{dt} \quad (4.3)$$

$$a_{tg} = \frac{dv}{dt} = L \frac{d^2\theta}{dt^2} \quad (4.4)$$

Na equação (4.2),  $s(t)$  é o comprimento de arco percorrido pelo pêndulo, derivando essa expressão em relação ao tempo chegamos na equação (4.3) que representa a velocidade tangencial do pêndulo; por fim, derivando novamente, obtemos sua aceleração tangencial dada pela equação (4.4).

Substituindo a expressão para a aceleração tangencial, (4.4), na expressão para força resultante na direção tangencial, (4.1), e reorganizando um pouco os termos, chegamos na equação diferencial que modela o movimento do pêndulo simples:

$$\frac{d^2\theta}{dt^2} + \frac{g}{L} \sin\theta = 0 \quad (4.5)$$

Sendo uma equação diferencial de segunda ordem, por conter uma derivada segunda, precisamos de duas condições iniciais para que o problema fique em uma forma matematicamente precisa. Para tanto vamos imaginar que no instante  $t = 0$  o pêndulo esteja no seu ponto de equilíbrio estável,  $\theta(t = 0) = 0$ , e que se dê uma pancada seca na massa imprimindo nela uma velocidade  $v_0$ . Expressando matematicamente essa condição inicial, temos:

$$\theta(0) = 0 \text{ e } \frac{d\theta}{dt} \Big|_{t=0} = \frac{v_0}{L} \quad (4.6)$$

Como dito acima, a equação 4.5 é de segunda ordem, porém, não linear devido a presença do termo em  $\sin\theta$ , assim não temos como obter uma solução em termos de funções elementares. Uma possível solução analítica até pode ser encontrada, mas em termos de integrais elípticas, que tem seus valores tabelados, ver Beléndez *et al.* (2007).

Então, como não sabemos resolver a equação (4.5) analiticamente aplicaremos uma simplificação adicional e a resolveremos no regime chamado de *pequenas oscilações*.

Neste limite, consideramos a pancada inicial suficientemente fraca de forma que a amplitude máxima da oscilação, valor máximo do ângulo  $\theta$ , seja pequena.

Assim poderemos usar a aproximação  $\sin\theta \approx \theta$  na equação (4.5), ficando:

$$\frac{d^2\theta}{dt^2} \approx -\frac{g}{L}\theta \quad (4.7)$$

que tem solução analítica e considerando as condições iniciais (4.6) é dada por:

$$\theta(t) \approx \theta_{max} \sin\left(\sqrt{\frac{g}{L}}t\right) \quad (4.8)$$

onde  $\theta_{max}$ , amplitude máxima, e o período de oscilação serão dados por:

$$\theta_{max} \approx \frac{v_0}{\sqrt{gL}} \quad (4.9)$$

$$T \approx 2\pi\sqrt{\frac{L}{g}} \quad (4.10)$$

Interessante notar que o período de oscilação do pêndulo não depende da amplitude, mas apenas de seu comprimento. Essa característica poderia ser usada para estabelecer um padrão de tempo global, pois, considerando  $g \approx 9,8m/s^2$  em toda superfície terrestre, bastaria a alguém construir um pêndulo com  $L = 25cm$ , onde poderia definir um segundo como o período de oscilação de tal pêndulo dentro de uma precisão da ordem de 1%. Para obter tal precisão bastaria a amplitude máxima de oscilação ser menor que  $14^\circ$ .

Claro que para várias aplicações práticas a precisão de 1% não é suficiente e utiliza-se de outras formas para definir o padrão de tempo(OLIVEIRA, 2010).

#### 4.1.2 Métodos numéricos

Como vimos, com as devidas simplificações foi possível eliminar o termo não linear da equação (4.5), obtendo-se facilmente uma solução analítica; porém, em muitas ocasiões isso pode não ser fácil, possível ou mesmo desejável, existindo casos onde a simplificação pode eliminar características cruciais do problema descaracterizando-o completamente.

Felizmente no caso do pêndulo podemos atacar diretamente a equação não linear com o uso de métodos computacionais apropriados. É objetivo dessa seção a descrição e aplicação de dois desses métodos ao problema do pêndulo, quando também buscaremos evidenciar a correta validação da aplicabilidade ao problema.

A equação (4.5) é de segunda ordem, e mesmo sendo não linear, devido a presença de uma função trigonométrica, ela não tem termos cruzados entre a derivada e a variável independente, então, como nos ensina Klein e Godunov (2006, sec. 8.4), poderemos decompor a equação em um sistema de equações de primeira ordem acopladas.

Realizando a decomposição ficamos com o seguinte sistema:

$$\frac{d\theta}{dt} = v \quad (4.11)$$

$$\frac{dv}{dt} = -\frac{g}{L} \sin \theta \quad (4.12)$$

Devemos buscar métodos para resolver equações do tipo:

$$\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}, t) \quad (4.13)$$

Os métodos que discutiremos tem em comum basearem-se num mesmo esquema numérico conhecido como: *Métodos Runge-Kutta de ordem n*, em que para resolver uma equação do tipo 4.13 usam apenas o conhecimento dos valores de  $x_j = x(t_j)$  para calcular os valores de  $x_{j+1} = x(t_{j+1})$  (SCHERER, 2005).

Como observa Scherer (2005, cap. 2), o primeiro passo para obter uma solução de 4.13 é discretizar a variável contínua  $t$ , ou seja, troca-lá por um conjunto discreto de  $n$  valores igualmente espaçados por  $\Delta t$ , também chamado de passo, e contidos no intervalo de interesse  $[0, t_{max}]$ :

$$t_0 = 0, t_1 = t_0 + \Delta t, t_2 = t_1 + \Delta t, \dots, t_n = t_{max}$$

Como próximo passo, nos aponta Scherer (2005, cap. 2), é escrever a equação diferencial numa forma conhecida como *equação de diferenças*, usando para isso uma das possíveis expressões para a aproximação da derivada de uma função:

$$\frac{df(x)}{dx} = \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (4.14)$$

$$\frac{df(x)}{dx} = \frac{f(x) - f(x - \Delta x)}{\Delta x} \quad (4.15)$$

$$\frac{df(x)}{dx} = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} \quad (4.16)$$

As expressões no numerador das equações acima são chamadas de "diferenças finitas de primeira ordem de  $f$  em  $x$ ". Essas equações, são aproximações da derivada de  $f$  em  $x$ , respectivamente, à direita, à esquerda e centrada. Sendo a função  $f$  derivável, no limite de  $\Delta x \rightarrow 0$ , fornecerão o mesmo valor (SCHERER, 2005).

Se aproximarmos a derivada dada pela equação (4.13) fazendo uso da expressão 4.14 e da discretização do tempo, poderemos escrever a derivada como uma equação de diferenças finitas em uma forma iterativa, a partir da qual obteremos, por recorrência, todos os valores de  $x_j$  solução do nosso problema:

$$\frac{dx}{dt} = \frac{x_{j+1} - x_j}{\Delta t} \quad (4.17)$$

frisamos novamente o uso da notação  $x_j = x(t_j)$  e  $x_{j+1} = x(t_{j+1})$ .

Por fim, usando a função que define a derivada, dada pela equação (4.13), chegamos a forma computacional do método que se conhece como *método Runge-Kutta de 1ª ordem*, Runge-Kutta de 1a. Ordem (RK1), também conhecido como **Método de Euler**:

$$x_{j+1} = x_j + \Delta t f(x_j, t_j) \quad (4.18)$$

É interessante observar que para obtermos o método de Euler aproximamos a derivada usando o valor da função no início do intervalo,  $x_j$ . Uma melhoria desse método pode ser dado pela redução do passo ( $\Delta t$ ), porém, quanto menor o passo mais tempo de processamento será necessário. Em determinados problemas e/ou condições, somente reduzir o passo pode não resolver; então devemos na verdade buscar melhorar o esquema numérico usado (KLEIN; GODUNOV, 2006).

Uma melhoria no método de Euler é o **Método de Euler Modificado** ou *Runge-Kutta de 2ª ordem*, Runge-Kutta de 2a. Ordem (RK2), que consiste em calcular a derivada no ponto médio do intervalo, ficando a relação de recorrência da seguinte forma:

$$x_{j+1} = x_j + \Delta t f\left(\frac{x_j + x_{j+1}}{2}, \frac{t_j + t_{j+1}}{2}\right) \quad (4.19)$$

Para a aplicação dessa expressão existe uma dificuldade, conforme explica Scherer (2005), que é não termos o valor de  $x_{j+1}$ . Como saída Scherer (2005) nos ensina que um procedimento possível é usar o método de Euler, equação (4.18), para uma primeira aproximação de  $x_{j+1}$ , com isso em mente a nossa relação de recorrência final ficará:

$$x_{j+1} = x_j + \Delta t f(x_j + 0.5\Delta t f(x_j, t_j), t_j + 0.5\Delta t) \quad (4.20)$$

Podemos, como pode ser visto em Klein e Godunov (2006, sec. 8.6), aplicar o mesmo procedimento na busca de novas melhorias, calculando as derivadas em pontos intermediários do intervalo, aumentando assim a precisão de nosso algoritmo.

Em Klein e Godunov (2006, sec. 8.6) podemos ver os diversos passos realizados para realizar a melhoria na derivação numérica, e obtermos os esquemas numéricos que definem os Métodos de Runge-Kutta de qualquer ordem. Com base nesse procedimento, são essas as equações iterativas referentes ao método Runge-Kutta de 4ª ordem:

$$\begin{aligned}
 F_1 &= f(x_j, t_j) \\
 F_2 &= f\left(x_j + \frac{h}{2}F_1, t_j + \frac{h}{2}\right) \\
 F_3 &= f\left(x_j + \frac{h}{2}F_2, t_j + \frac{h}{2}\right) \\
 F_4 &= f(x_j + hF_3, t_j + h) \\
 x_{j+1} &= x_j + \frac{\Delta t}{6}(F_1 + 2F_2 + 2F_3 + F_4)
 \end{aligned} \tag{4.21}$$

### 4.1.3 Validações dos métodos numéricos

Sempre que buscamos uma solução numérica para um problema devemos nos preocupar em checar se os valores obtidos estão em concordância ou com alguma solução analítica ou se fazem sentido usando de algum critério. Como um tal critério, poderíamos verificar se o problema apresenta alguma característica física que possa ser usada; por exemplo, se o modelo descrever um sistema conservativo podemos usar a lei de conservação da energia e verificar se ela é respeitada durante a evolução da solução numérica.

Como exemplos, vamos aplicar os métodos RK1 e RK2 a um caso onde conhecemos a solução analítica, e assim, realizaremos a comparação dela com a solução numérica obtida, e também ao modelo do pêndulo simples onde iremos calcular e monitorar o valor da energia total do sistema em cada instante, verificando se sua evolução temporal respeita a lei da conservação da energia.

Consideremos a equação de 1ª ordem e sua solução analítica:

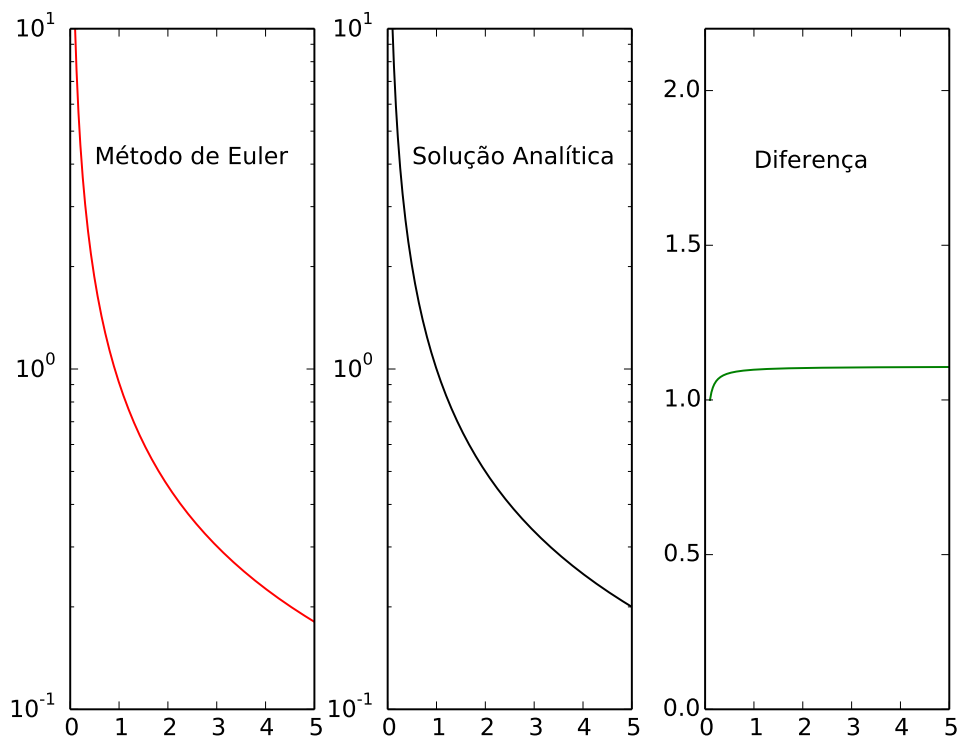
$$x \frac{dy}{dx} = -y \quad (4.22)$$

$$y = \frac{C}{x} \quad (4.23)$$

onde  $C$  é uma constante cujo valor dependerá das condições iniciais.

Nas figuras abaixo, exibimos os gráficos dos resultados numérico e analítico, além de representarmos graficamente o quociente entre os valores das duas soluções a título de comparação. Fazemos isso para os dois métodos numéricos.

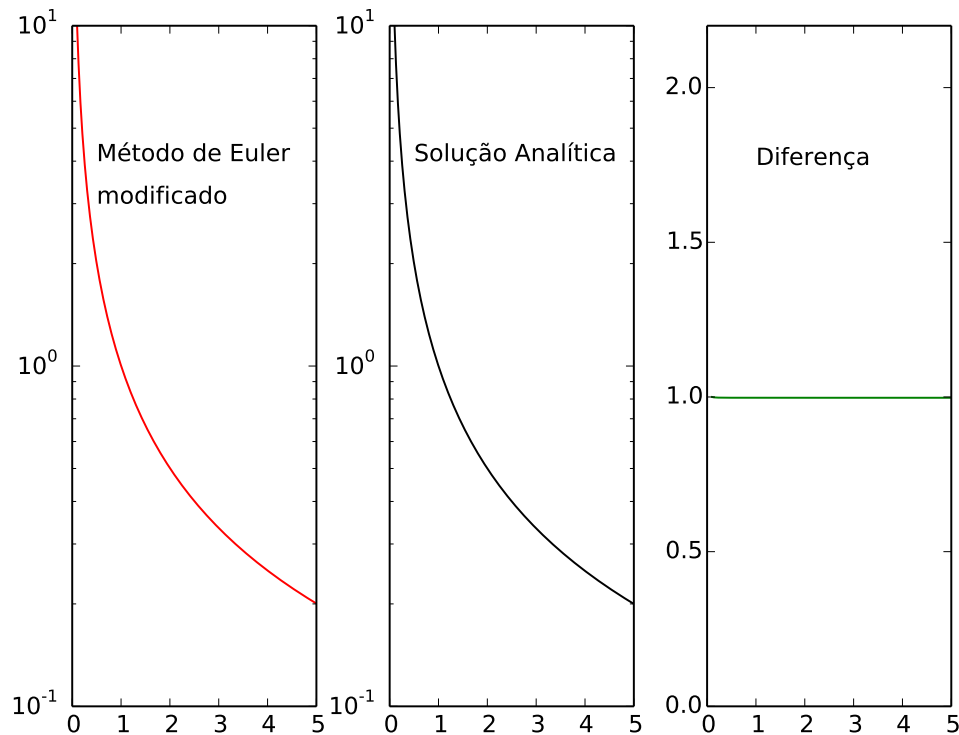
Figura 6 – Solução de  $x \frac{dy}{dx} = -y$  analítica e por Runge-Kutta 1ª ordem



Fonte: Elaborada pelo autor(2019).

Para o método RK1, a divergência com a solução analítica é razoável e sua aplicabilidade vai depender da precisão desejada. No caso do método RK2, temos uma concordância muito boa, como pode ser visto no gráfico de comparação, onde razão entre os valores das soluções se mantém igual a 1 no período de simulação. Nesse exemplo fizemos uso da solução analítica disponível para validarmos a aplicabilidade dos métodos, com isso poderemos ir realizando

Figura 7 – Solução de  $x \frac{dy}{dx} = -y$  analítica e por Runge-Kutta 2ª ordem



Fonte: Elaborada pelo autor(2019).

ajustes nos parâmetros de controle para obtermos o grau de precisão que desejamos. Porém, por vezes, para obter um aumento de precisão temos que fazer uso de um método mais elaborado.

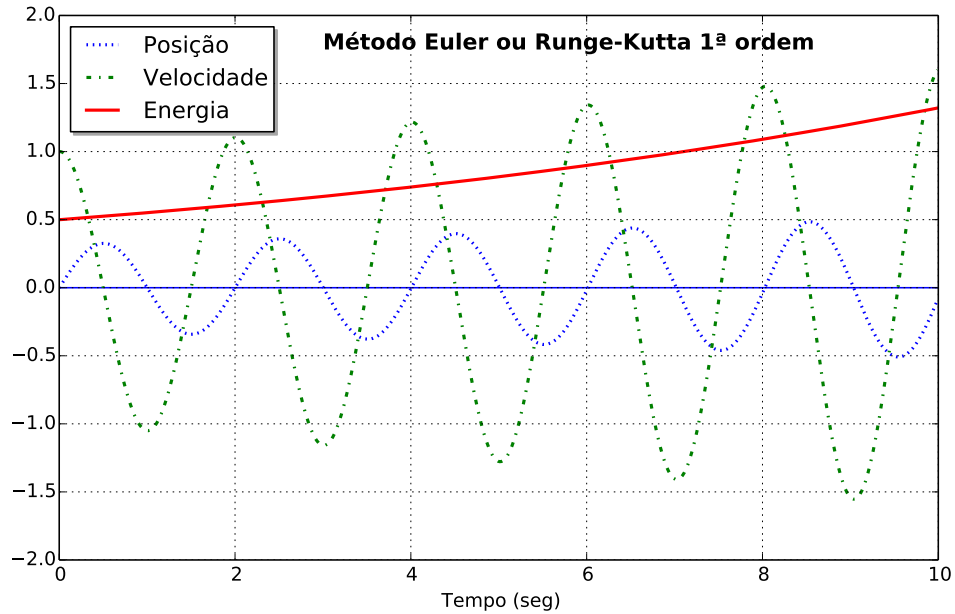
No apêndice A são fornecidos os códigos fontes escritos em Python que implementam os algoritmos e geram os gráficos apresentados acima usando a library *matplotlib*.

Agora aplicaremos os métodos RK1 e RK2 ao problema do pêndulo, fazendo uso das equações (4.11) e (4.12). Nas figura abaixo exibimos as soluções numéricas para as evoluções temporais da posição e velocidade do pêndulo e calculamos também o valor da sua energia total. Seguem os resultados:

Analisado o comportamento temporal da energia total do pêndulo, vemos que, dentro do tempo de simulação considerado, o método RK2 preserva bem o nosso vínculo que é o da conservação da energia, ou seja, o valor calculado da energia permanece constante no tempo; o mesmo não acontecendo na solução usando o método RK1 onde a energia diverge rapidamente.

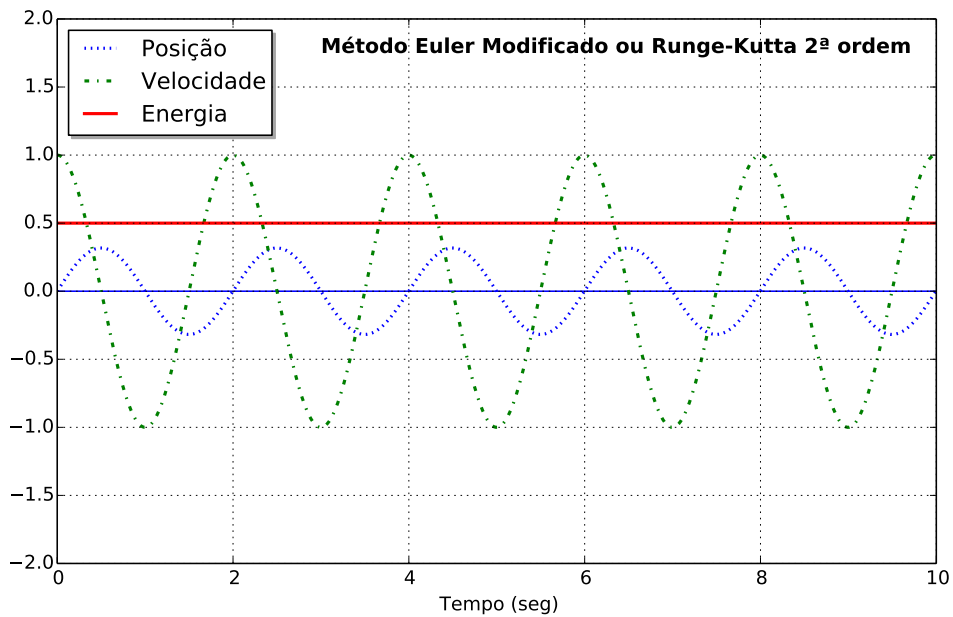
Por fim, podemos fazer a seguinte pergunta: Se aumentarmos o tempo de simulação, será que o método de Runge-Kutta 2ª ordem manterá a mesma precisão e continuará a atender nosso vínculo? Vejamos o resultado:

Figura 8 – Solução do pêndulo simples por Runge-Kutta 1ª ordem



Fonte: Elaborada pelo autor(2019).

Figura 9 – Solução do pêndulo simples por Runge-Kutta 2ª ordem

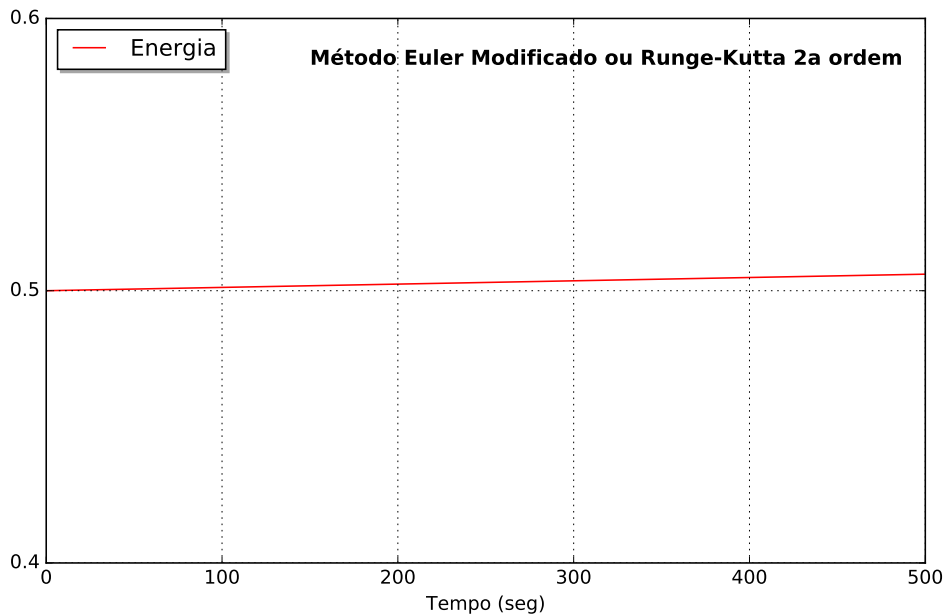


Fonte: Elaborada pelo autor(2019).

Como podemos ver na figura acima, com o aumento do tempo de simulação a energia começa a divergir mostrando que a precisão do método numérico já não é suficiente para manter nosso vínculo. Precisaremos ou mudar parâmetros ou mudar o método. Isso nos mostra o cuidado que devemos ter ao analisar a resposta obtida da simulação numérica de um problema. Os códigos fontes para geração dos resultados acima se encontram no Apêndice B.



Figura 10 – Solução do pêndulo simples por Runge-Kutta 2ª ordem - Tempo estendido



Fonte: Elaborada pelo autor(2019).

#### 4.1.4 Discussão e abordagens

Acreditamos que o material mostrado nas seções anteriores aplica-se melhor no nível de graduação; onde, após alguma introdução a linguagem Python e suas bibliotecas, os alunos poderiam realizar a implementação dos métodos numéricos discutidos e também realizarem as devidas etapas de análise dos resultados. Mas, podemos imaginar também, uma forma mais dirigida dos programas ou até mesmo seu uso de forma demonstrativa onde o foco não seria implementá-los, mas, usá-los para fomentar discussões dos fenômenos e como podemos realizar sua representação matemática e sua resolução fazendo uso do computador.

#### 4.2 Experimento 2 - Visualizando oscilações amortecidas

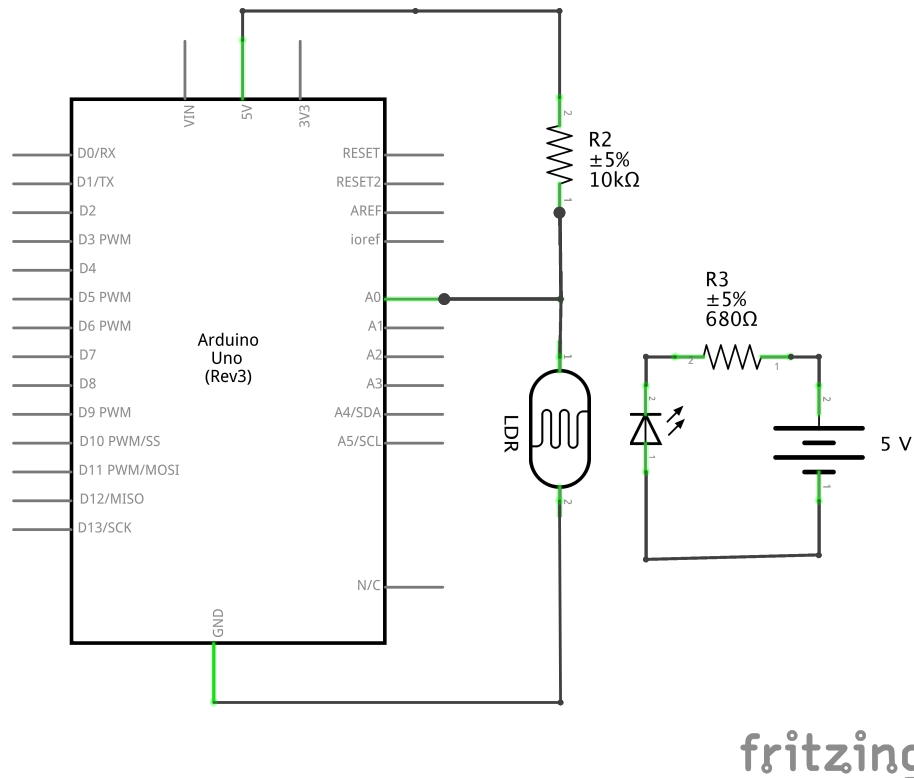
Nesse experimento vamos montar um aparato para visualização de oscilações amortecidas implementado conforme sugerido por (SOUZA *et al.*, 2011), porém, faremos uma modificação, e, ao invés da gravação dos dados em arquivo para posterior leitura, iremos realizar sua coleta e exibição em tempo real. A coleta e envio pela serial será realizada usando a placa micro-processada Arduino. Implementamos também um programa em Python que irá ler os dados da serial e realizar sua exibição.

### 4.2.1 Detalhes do experimento

Realizamos a montagem da parte eletrônica do experimento em um protoboard, eliminando o uso de soldas e permitindo realizar alterações e ajustes facilmente.

Abaixo temos uma visão esquemática do circuito:

Figura 11 – Visão esquemática



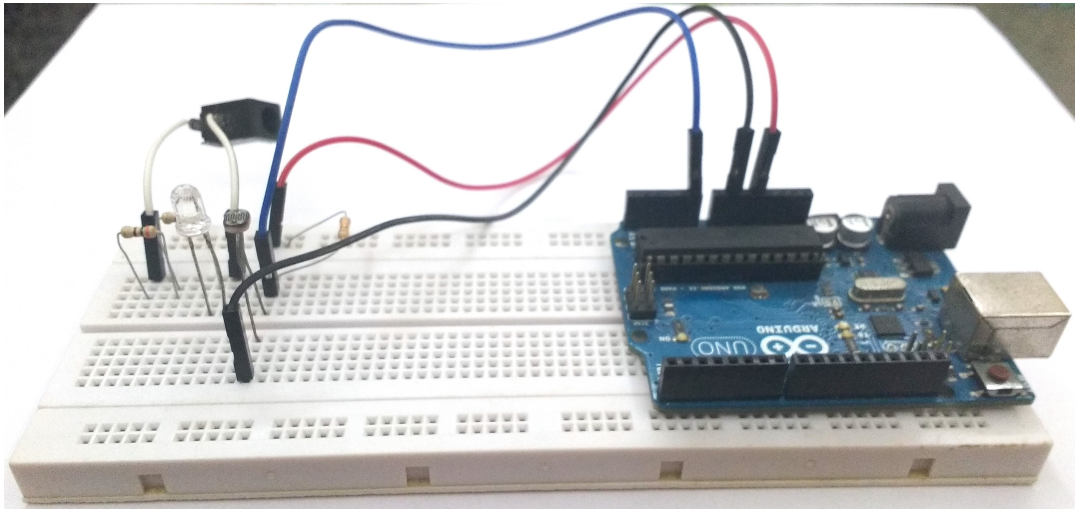
Fonte: Elaborada pelo autor(2019).

Observando o diagrama acima vemos que o Do inglês: Light Dependent Resistor - Resistor dependente da luz (LDR) e o resistor R2 formam um divisor de tensão na entrada analógica A0 do Arduino, de modo que, variando a luminosidade que atinge o LDR variamos sua resistência e daí a Diferença de potencial (D.D.P) sobre o mesmo. A D.D.P sobre o LDR é lido pelo Arduino, convertido por um conversor analógico-digital interno e enviado pela serial ao notebook. O processo de leitura da porta analógica e envio ao notebook é realizado pelo programa gravado na memória flash do Arduino, cujo código fonte pode ser visto no Apêndice C.

Abaixo podemos ver a montagem no protoboard do esquema proposto. No mesmo protoboard inserimos o LDR e o Do inglês: Light Emissor Diode - Diódo emissor de luz (LED)

de alta luminosidade. Usamos uma fonte externa para alimentar o circuito que aciona o LED, porém, cabe dizer que poderíamos ter usado o próprio Arduino para isso já que ele tem saídas nas tensões de 3.3 ou 5.0V com uma corrente máxima de 50mA, obtidas da própria USB, o que também é um facilitador do uso de tal plataforma.

Figura 12 – Montagem no protoboard



Fonte: Elaborada pelo autor(2019).

Abaixo temos a montagem definitiva: parte eletrônica e mecânica do experimento:

Figura 13 – Montagem definitiva



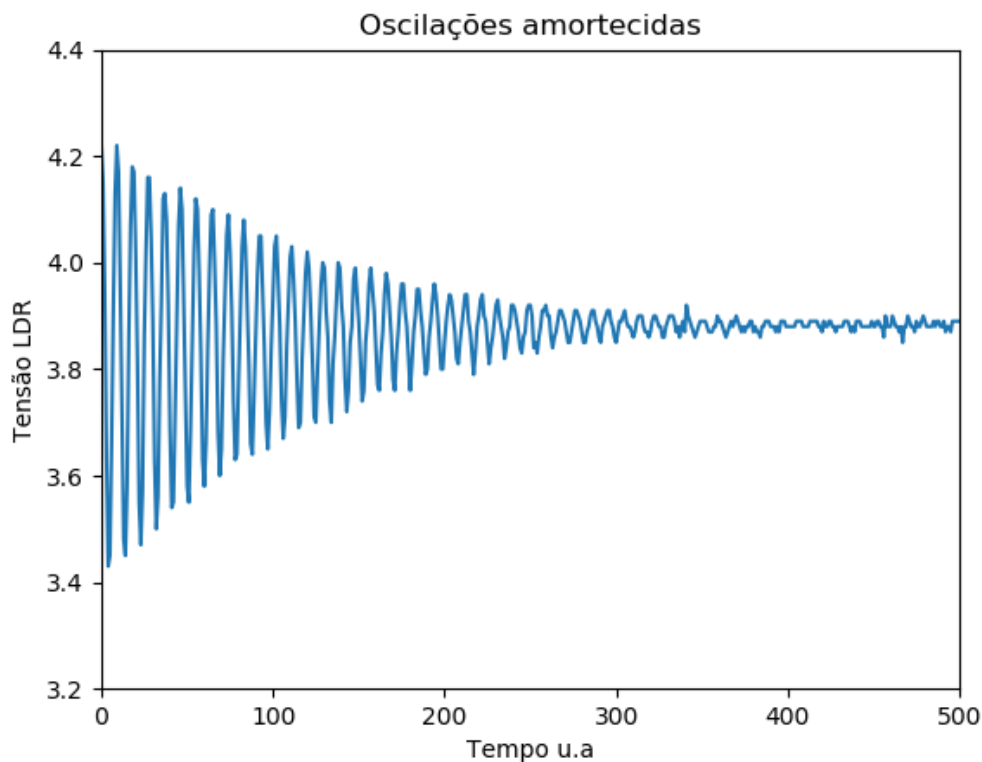
Fonte: Elaborada pelo autor(2019).

#### 4.2.2 Aplicações e Resultados

Após realizada a montagem física do experimento, conectamos o Arduino ao notebook via porta USB; então utilizando o Arduino IDE <sup>1</sup>, realizamos o upload do programa de captura e envio dos valores lidos da porta analógica. Após o upload ser concluído, o microcontrolador já estará pronto, capturando e enviando os dados para o notebook. Interessante notar que a interface com a porta serial é bem facilitada pela emulação que temos via porta USB.

Feita a preparação do software no lado do Arduino, executamos o programa Python no notebook, e, após alguns segundos necessários para completa inicialização da interface serial nos será apresentada uma janela que conterá o gráfico dos valores recebidos em tempo real. O valor estacionário em torno do qual ocorrerá a oscilação, dependerá das condições de iluminação do ambiente e posicionamento relativo do LED e do LDR; Em nossos testes, ajustamos para que ficasse por volta de 3.9V. A execução do experimento consta de movermos a régua de seu ponto de equilíbrio, quando então visualizaremos a variação na amplitude da voltagem sendo lida, refletindo a variação da luminosidade/tensão no LDR, como pode ser visto na figura:

Figura 14 – Leitura serial em tempo real



Fonte: Elaborada pelo autor(2019).

<sup>1</sup> Pode ser baixado em: <https://www.arduino.cc/en/Main/Software>

### ***4.2.3 Discussões e abordagens***

O experimento apresentado abre espaço para abordarmos vários assuntos interessantes e com viés multidisciplinar. Na montagem em si já podemos tratar dos aspectos elétricos, eletrônicos e mecânicos envolvidos, tais como: divisores de tensão, o funcionamento do LDR em si e seu papel na visualização das oscilações da régua, a forma como ocorrem as diversas transduções até obtermos a visualização dos dados. Obviamente uma discussão interessante seria a busca de outros práticos, tais como: sensores de passagem, medidores de velocidades, controle de luminosidade ambiente, etc.

A parte referente aos programas de suporte, é também rica, e contém assuntos que a depender do nível da turma podem ser aprofundados. Apesar de simples, os programas permitem trabalhar a lógica de programação, o funcionamento do Arduino, o funcionamento da interface Arduino com o notebook via o programa em Python, verificar a modificação de parâmetros, seja no programa Arduino, seja no programa Python e ver como o resultado se altera, e assim por diante.

Esse tipo de atividade onde mesclamos a implementação física dos experimentos, no estilo mão na massa, com a parte da programação, é cada vez mais bem recebida pelos alunos e integra habilidades com crescente demanda no mundo atual. Hoje ter noções de programação de dispositivos/computadores ou pelo menos saber pensar de forma algorítmica é sem dúvidas uma demanda crescente nas mais diversas áreas seja da ciência e tecnologia.

## 5 CONCLUSÕES

No decurso desse trabalho acreditamos ter posicionado o uso do computador e de dispositivos micro-processado, além das ferramentas Open Source, como extremamente úteis no apoio e melhoria do processo de ensino-aprendizagem, notadamente da Física. Teorias de aprendizagem corroboram tal utilidade quando percebem que o uso de tal ferramental apoia e reforça a adição cognitiva que o aluno, cujo processo faça uso de tal suporte, terá.

Ferramentas de uso livre, como as apresentadas: Arduino e Python, facilitam a aplicação, seja no tocante aos custos, seja na facilidade de uso, dada a vasta documentação disponível na rede. Com essas informações e ferramentas em mãos os professores podem soltar a criatividade na busca das mais diversas aplicações. A bibliografia mais recente nos dá diversos exemplos de uso envolvendo o Arduino e tecnologias relacionadas, alguns exemplos, temos: Neto *et al.* (2018), Guadagnini *et al.* (2019), Cid e Correa (2019), Aet *al.* (2019).

## REFERÊNCIAS

A, J. A. F. N. J.; BORGES, V. E. S.; NASCIMENTO, R. M. M. F. Descrição temporal de forças de colisão: um modelo didático para laboratório de física assistido por sistema embarcado. **Revista Brasileira de Ensino de Física**, scielo, v. 41, 00 2019. ISSN 1806-1117. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1806-11172019000300501&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172019000300501&nrm=iso)>.

BELÉNDEZ, A.; PASCUAL, C.; MÉNDEZ, D.; BELÉNDEZ, T.; NEIPP, C. Exact solution for the nonlinear pendulum. **Revista brasileira de ensino de física**, SciELO Brasil, v. 29, n. 4, p. 645–648, 2007.

BRASIL, M. **Parâmetros Curriculares Nacionais: Ensino Médio. Parte III. Ciências da Natureza, Matemática e suas Tecnologias**. Brasília: MEC/SEMTEC, 2010.

CAVALCANTE, M. A.; TAVOLARO, C. R. C.; MOLISANI, E. Física com arduino para iniciantes. **Revista Brasileira de Ensino de Física**, SciELO Brasil, v. 33, n. 4, p. 4503, 2011.

CID, A.; CORREA, T. Venturino: análise da variação de pressão em um tubo de Venturi utilizando Arduino e sensor de pressão. **Revista Brasileira de Ensino de Física**, scielo, v. 41, 00 2019. ISSN 1806-1117. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1806-11172019000300606&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172019000300606&nrm=iso)>.

DORNELES, P. F. T.; ARAUJO, I. S.; VEIT, E. A. Integração entre atividades computacionais e experimentais como recurso instrucional no ensino de eletromagnetismo em física geral. **Ciência e Educação (Bauru)**, scielo, v. 18, p. 99 – 122, 00 2012. ISSN 1516-7313. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1516-73132012000100007&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1516-73132012000100007&nrm=iso)>.

ELIAS, D. C. N.; ARAÚJO, M. S. T.; JR, C. F. de A.; AMARAL, L. H. Tendências das propostas de utilização das ferramentas computacionais no ensino de física no nível médio e superior proposals tendencies of use of the computation tools for medium and higher education of physics teaching. 2009.

FIOLHAIS, C.; TRINDADE, J. Física no computador: o computador como uma ferramenta no ensino e na aprendizagem das ciências físicas. **Revista Brasileira de Ensino de Física**, SciELO Brasil, v. 25, n. 3, 2003.

GUADAGNINI, P. H.; ROCHA, F. S. d.; BARLETTE, V. E. Um medidor de luminosidade com módulo sensor integrado e aquisição automática de dados com aplicações didáticas. **Revista Brasileira de Ensino de Física**, scielo, v. 41, 00 2019. ISSN 1806-1117. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1806-11172019000300605&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172019000300605&nrm=iso)>.

KLEIN, A.; GODUNOV, A. **Introductory Computational Physics**. Cambridge University Press, 2006. ISBN 9781139447522. Disponível em: <<https://books.google.ca/books?id=ISlgKyjC-sUC>>.

MAGALHÃES, M. G. M. d.; SCHIEL, D.; GUERRINI, I. M.; JR., E. M. Utilizando Tecnologia Computacional na Análise Quantitativa de Movimentos: Uma Atividade para Alunos do Ensino Médio. **Revista Brasileira de Ensino de Física**, scielo, v. 24, p. 97 – 102, 06 2002. ISSN 1806-1117. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1806-11172002000200004&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172002000200004&nrm=iso)>.

NETO, J. A. T. d. C.; APOLINÁRIO, F. R.; SOARES, A. d. A. Sistema photogate de seis canais analógicos para laboratórios didáticos de física. **Revista Brasileira de Ensino de Física**, scielo, v. 40, 00 2018. ISSN 1806-1117. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1806-11172018000100604&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172018000100604&nrm=iso)>.

OLIVEIRA, P. M. C. d. **Física em computadores**. São Paulo: Editora Livraria da Física, 2010.

ROCHA, F. Acelerômetro eletrônico e a placa arduino para ensino de física em tempo real. **Caderno Brasileiro de Ensino de Física**, v. 31, n. 1, p. 98–123, 2013. ISSN 2175-7941. Disponível em: <<https://periodicos.ufsc.br/index.php/fisica/article/view/2175-7941.2014v31n1p98>>.

SCHERER, C. **Métodos computacionais da Física**. [S.l.]: Editora Livraria da Física, 2005.

SERÉ, M.-G. O papel da experimentação no ensino da física. **Caderno Brasileiro de Ensino de Física**, v. 20, n. 1, p. 30–42, 2003. ISSN 2175-7941. Disponível em: <<https://periodicos.ufsc.br/index.php/fisica/article/view/6560>>.

SOUZA, A. R. d.; A, A. C. P.; A, D. D. U.; DIAS, M. A.; DUARTE, S.; AMORIM, H. S. d. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC. **Revista Brasileira de Ensino de Física**, scielo, v. 33, p. 01 – 05, 03 2011. ISSN 1806-1117. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1806-11172011000100026&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172011000100026&nrm=iso)>.

VIEIRA, L.; LARA, V.; AMARAL, D. Demonstração da lei do inverso do quadrado com o auxílio de um tablet/smartphone. **Revista Brasileira de Ensino de Física**, SciELO Brasil, v. 36, n. 3, p. 3505, 2014.

VIEIRA, L. P.; LARA, V. d. O. M. Macro photography with a tablet: applications on science teaching. **Revista Brasileira de Ensino de Física**, SciELO Brasil, v. 35, n. 3, p. 1–5, 2013.



**APÊNDICE A – MÉTODOS RUNGE-KUTTA 1ª E 2ª ORDEM PARA SOLUCIONAR**

$$x \frac{dy}{dx} + y = 0$$

```
1 # coding=utf-8
2 __author__ = elissandro
3
4 #
5 # Implementa o método de Euler para resolver
   numericamente
6 # a equação  $x dx/dy + y = 0$  e comparar com a solução
   analítica.
7 #
8 # Autor: Elissandro A. Mendes
9 #
10
11 import numpy as np
12 import matplotlib.pyplot as plt
13
14 x = [0.1, 5.0] # Intervalo em  $X[x_0, x_{max}]$ 
15 N = 500 # Numero de pontos a ser calculado
16
17 h = (x[1] - x[0])/N # Calculando o tamanho do passo
18
19 # Inicializando vetores
20 X = np.zeros(N)
21 X[0] = x[0]
22
23 Y = np.zeros(N)
24 Y[0] = 1/x[0]
25
26 # Loop do método de Euler
27 for i in range(1, N):
```

```
28     Y[i] = Y[i - 1] * (1 - h / X[i - 1])
29     X[i] = X[i - 1] + h
30
31 # Inicializa o , configura o e plotagem dos graficos
32 fig = plt.figure()
33 fig.subplots_adjust(wspace=0.3)
34
35 plt.xlim(0, 5)
36 plt.xlabel( X )
37
38 plt.subplot(1, 3, 1)
39 plt.semilogy(X, Y, r- )
40 plt.text(0.5, 4, u M todo de Euler )
41
42 plt.subplot(1, 3, 2)
43 plt.semilogy(X, 1/X, k- )
44 plt.text(0.5, 4, u Solu o Anal tica )
45
46 plt.subplot(1, 3, 3)
47 plt.ylim(0, 2.2)
48 plt.plot(X, 1/(Y*X), g- )
49 plt.text(1, 1.75, u Diferen a )
50 plt.savefig( ../figuras/compEulerAnalitical.pdf , dpi=None,
51             facecolor= w , edgecolor= w , orientation= portrait
52             ,
53             papertype=None, format= pdf , transparent=True,
54             bbox_inches=None, pad_inches=0.1, frameon=None)
54 plt.show()
```

```
1 # coding=utf-8
2 __author__ = elissandro
3
```

```
4 #
5 # Implementa o método de Euler modificado para
   resolver numericamente
6 # a equação  $x dx/dy + y = 0$  e comparar com a solução
   analítica.
7 #
8 # Autor: Elissandro A. Mendes
9 #
10
11 import numpy as np
12 import matplotlib.pyplot as plt
13
14 x = [0.1, 5.0] # Intervalo em  $X[x_0, x_{max}]$ 
15 N = 500 # Numero de pontos
16
17 h = (x[1] - x[0])/N # Calculando o tamanho do passo
18
19 # Inicializa o
20 X = np.zeros(N) # Vetor com valores de X no intervalo;
21 X[0] = x[0]
22
23 Y = np.zeros(N) # Vetor com valores calculados pelo
   método;
24 Y[0] = 1/x[0]
25
26 Xmid = X[0] + h/2
27 Ymid = Y[0] - (h/2)*Y[0]/X[0]
28
29 # Loop do método de Euler modificado (RK2)
30 for i in range(1, N):
31     Y[i] = Y[i - 1] - h * Ymid/Xmid
32     X[i] = X[i - 1] + h
```

```
33
34     Xmid = X[i] + h/2
35     Ymid = Y[i] - (h/2)*Y[i]/X[i]
36
37
38 # Inicializa o , configura o e plotagem gr fica
39 fig = plt.figure()
40 fig.subplots_adjust(wspace=0.3)
41
42 plt.xlim(0, 5)
43 plt.xlabel( X )
44
45 plt.subplot(1, 3, 1)
46 plt.semilogy(X, Y, r- )
47 plt.text(0.5, 4, u M todo de Euler )
48 plt.text(0.5, 3, u modificado )
49
50 plt.subplot(1, 3, 2)
51 plt.semilogy(X, 1/X, k- )
52 plt.text(0.5, 4, u Solu o Anal tica )
53
54 plt.subplot(1, 3, 3)
55 plt.ylim(0, 2.2)
56 plt.plot(X, 1/(Y*X), g- )
57 plt.text(1, 1.75, u Diferen a )
58 plt.savefig( ../figuras/compEulerModAnalitical.pdf , dpi=
59     None, facecolor= w , edgecolor= w ,
60     orientation= portrait , papertype=None, format= pdf
61     ,
62     transparent=True, bbox_inches=None, pad_inches=0.1,
63     frameon=None)
64 plt.show()
```

## APÊNDICE B – MÉTODOS RUNGE-KUTTA 1ª E 2ª ORDEM PARA O PÊNDULO SIMPLES

```
1 # coding=utf-8
2 __author__ = elissandro
3
4 #
5 # Solu o do p ndulo usando o m todo de Euler
6 #
7 # Autor: Elissandro A. Mendes
8 #
9
10 import math
11 import numpy as np
12 import matplotlib.pyplot as plt
13
14 tmax = 5. # Tempo de simula o em segundos;
15 dt = 0.01 # Incremento no tempo;
16 N = int(tmax / dt) # Calculando o n mero de itera es
    da simula o;
17
18 # Inicializando vetores
19 T = np.zeros(N) # Instante de tempo: 0, dt, 2dt, etc.
20 theta = np.zeros(N) # Deslocamento angular;
21 veloc = np.zeros(N) # Velocidade do p ndulo;
22 en = np.zeros(N) # Energia total;
23
24 T[0] = 0.
25 theta[0] = 0. # Posi o inicial. P ndulo no ponto de
    equil brio est vel;
26 veloc[0] = 1.0 # Velocidade inicial 1.0 m/s
```

```

27 en[0] = 0.5 * veloc[0] * veloc[0] # Calculamos a energia
    inicial;
28
29 # Loop do m todo de Euler
30 for i in range(0, N - 1):
31     T[i + 1] = T[i] + dt
32
33     en[i + 1] = 0.5 * veloc[i] * veloc[i] + 10 * (1 - math.
        cos(theta[i]))
34
35     theta[i + 1] = theta[i] + dt * veloc[i]
36     veloc[i + 1] = veloc[i] - 10 * dt * math.sin(theta[i])
37
38 # Inicializa o , configura o e plotagem dos gr ficos
39 plt.figure(num=1, figsize=(10, 6))
40 plt.grid()
41 plt.ylim(-2, 2)
42 plt.yticks(np.arange(-2, 2.5, 0.5))
43 plt.text(3, 1.75, u M todo Euler ou Runge-Kutta 1a. ordem
    , fontsize=14, fontweight= bold )
44 plt.axhline()
45 plt.xlabel( Tempo (seg) )
46 plt.plot(T, theta, b: , label=u Posi o , linewidth=2)
47 plt.plot(T, veloc, g-. , label= Velocidade , linewidth=2)
48 plt.plot(T, en, r- , label= Energia , linewidth=2)
49 plt.legend(loc= upper left , shadow=True)
50
51 plt.savefig( ../figuras/osceuler.pdf , dpi=None, facecolor
    = w , edgecolor= w ,
52     orientation= portrait , papertype=None, format= pdf
    ,
53     transparent=True, bbox_inches=None, pad_inches=0.1,

```

```
54         frameon=None)  
55  
56 plt.show()
```

```
1 # coding=utf-8  
2 __author__ = elissandro  
3  
4 #  
5 # Solu o do p ndulo usando o m todo de Euler  
   modificado  
6 #  
7 # Autor: Elissandro A. Mendes  
8 #  
9  
10 import math  
11 import numpy as np  
12 import matplotlib.pyplot as plt  
13  
14 # Remover coment rio para gerar figura 10 do texto.  
15 # tmax = 500.  
16 tmax = 50.  
17 dt = 0.01  
18 N = int(tmax / dt)  
19  
20 # Inicializando vetores que v o conter os valores  
   calculados  
21 T = np.zeros(N)  
22 theta = np.zeros(N)  
23 veloc = np.zeros(N)  
24 en = np.zeros(N)  
25  
26 T[0] = 0.
```

```

27 theta[0] = 0.
28 veloc[0] = 1.0
29 en[0] = 0.5 * veloc[0] * veloc[0]
30
31 # Loop do m todo de Euler modificado
32 for i in range(0, N - 1):
33     T[i + 1] = T[i] + dt
34
35     tmid = theta[i] + veloc[i] * dt / 2
36     vmid = veloc[i] - 10 * math.sin(theta[i]) * dt / 2
37
38     theta[i + 1] = theta[i] + dt * vmid
39     veloc[i + 1] = veloc[i] - 10 * dt * math.sin(tmid)
40
41     en[i + 1] = 0.5 * veloc[i + 1] * veloc[i + 1] + 10 * (1
42         - math.cos(theta[i + 1]))
43
44 # Inicializa o , configura o e plotagem dos gr ficos
45 plt.figure(num=1, figsize=(10, 6))
46 plt.grid()
47 # Remover coment rio para gerar figura 10 do texto.
48 # plt.ylim(0.4, 0.6)
49 # plt.yticks(np.arange(0.4, 0.7, 0.1))
50 plt.yticks(np.arange(-2, 2.5, 0.5))
51 plt.axhline()
52 plt.xlabel( Tempo (seg) )
53 # Remover coment rio para gerar figura 10 do texto.
54 # plt.text(150, 0.583, u M todo Euler Modificado ou Runge-
55     Kutta 2a ordem , fontsize=13, fontweight= bold )
56 plt.text(200, 1.75, u M todo Euler Modificado ou Runge-
57     Kutta 2a ordem , fontsize=13, fontweight= bold )
58 plt.plot(T, theta, b: , label=u Posi o , linewidth=2)

```



```
56 plt.plot(T, veloc, g-. , label= Velocidade , linewidth=2)
57 plt.plot(T, en, r- , label= Energia , linewidth=1)
58 plt.legend(loc= upper left , shadow=True)
59
60 plt.savefig(.../.../figuras/osceulerMod.pdf , dpi=None,
        facecolor= w , edgecolor= w ,
61         orientation= portrait , papertype=None, format= pdf
        ,
62         transparent=True, bbox_inches=None, pad_inches=0.1,
63         frameon=None)
64
65 plt.show()
```

**APÊNDICE C - LEITURA SERIAL ARDUINO**

```
1
2 const int PORTA_A0 = 0;
3 float valorTensaoLDR = 0;
4 int valorPortaA0 = 0;
5
6 void setup() {
7     Serial.begin(115200);
8 }
9
10 void loop() {
11     valorPortaA0 = analogRead(PORTA_A0);
12     valorTensaoLDR = (float) (valorPortaA0 / 1023.0) * 5.0;
13     Serial.println(valorTensaoLDR);
14     delay(20);
15 }
```

**APÊNDICE D – LEITURA SERIAL**

```
1 # coding=utf-8
2
3 import sys, serial, argparse, time
4 import numpy as np
5
6 from collections import deque
7 import matplotlib.pyplot as plt
8 import matplotlib.animation as animation
9
10 class AnalogPlot:
11     def __init__(self, strPort, maxLen):
12         self.ser = serial.Serial(strPort, 115200)
13
14         self.ax = deque([0.0]*maxLen)
15         self.maxLen = maxLen
16
17     def addToBuf(self, buf, val):
18         if len(buf) < self.maxLen:
19             buf.appendLeft(val)
20         else:
21             buf.popleft()
22             buf.append(val)
23
24     def add(self, data):
25         if (len(data) > 0):
26             self.addToBuf(self.ax, data[0])
27
28     def update(self, frameNum, a0):
29         try:
30             line = self.ser.readline()
```

```
31         data = [float(val) for val in line.split()]
32         self.add(data)
33         a0.set_data(range(self.maxLen), self.ax)
34     except:
35         pass
36
37     return a0,
38
39 def close(self):
40     self.ser.flush()
41     self.ser.close()
42
43 def main():
44     parser = argparse.ArgumentParser(description="LDR serial
45         ")
46     parser.add_argument(--port , dest= port , required=True)
47     parser.add_argument(--maxlen , dest= maxlen , required=
48         False)
49     parser.add_argument(--ymin , dest= ymin , required=False
50         )
51     parser.add_argument(--ymax , dest= ymax , required=False
52         )
53
54     args = parser.parse_args()
55
56     strPort = args.port
57
58     maxlen = 1000
59     if args.maxlen:
60         maxlen = int(args.maxlen)
61
62     ymin = float(args.ymin) if args.ymin else 3.0
```

```
59  ymax = float(args.ymax) if args.ymax else 5.0
60
61  print( Lendo dados da porta serial %s... % strPort)
62
63  analogPlot = AnalogPlot(strPort, maxlen)
64
65  # Ao abrir a serial o Arduino      resetado.
66  # Devemos aguardar ~1.5 segs antes de continuar.
67  time.sleep(1.5)
68
69  print( Plotando dados... )
70
71  fig = plt.figure()
72  ax = plt.axes(xlim=(0, maxlen), ylim=(ymin, ymax))
73  a0, = ax.plot([])
74  anim = animation.FuncAnimation(fig, analogPlot.update,
75                                fargs=(a0,), interval=1,
76                                blit=True)
77                                # frames=np.arange(1200),
78
79  plt.title(u Oscila es amortecidas )
80  plt.xlabel( Unidade arbitr rias )
81  plt.ylabel(u Tens o LDR (V) )
82  plt.show()
83
84
85  plt.close(fig)
86
87  analogPlot.close()
88
89  print( Saindo... )
90
91  if __name__ == __main__ :
```

90

`main()`