



## **Automação do tratamento de dados em projetos de extensão utilizando Python**

### **Automation of data handling in extension projects using Python**

**João Paulo Bachega**

[joaopaulobachega@gmail.com](mailto:joaopaulobachega@gmail.com)

Universidade Tecnológica Federal do Paraná, Londrina, Paraná, Brasil

**Roger Nabeyama Michels**

[rogernmichels@utfpr.edu.br](mailto:rogernmichels@utfpr.edu.br)

Universidade Tecnológica Federal do Paraná, Londrina, Paraná, Brasil

**Guilherme Quereza Vieira**

[guiquereza@hotmail.com](mailto:guiquereza@hotmail.com)

Universidade Tecnológica Federal do Paraná, Londrina, Paraná, Brasil

#### **RESUMO**

A automação de processos tem se tornado cada dia mais importante na busca por uma otimização do rendimento dos profissionais envolvidos nas mais diversas áreas. Este processo permite que o profissional não seja obrigado a realizar tarefas muitas vezes repetitivas e trabalhosas e possa focar em pontos de maior complexidade de seu trabalho. Com este intuito, surgiu a ideia de automatizar o processo de tratamento de organização dos dados obtidos pelo sistema de coleta baseado em Arduino desenvolvido na UTFPR. O sistema apresentou resultados satisfatórios, sendo utilizado em trabalhos de conclusão de curso, iniciações científicas e outros projetos de extensão. O resultado foi um software em linguagem Python que é capaz de ler o arquivo de texto gerado pelo equipamento de coleta de dados automatizado e realizar um tratamento básico, reorganizando as informações em uma planilha eletrônica e calculando a estatística descritiva, como média, moda e mediana, além de plotar um gráfico contendo a distribuição dos valores. O software foi testado com dados coletados no atendimento a uma disciplina na obtenção dos valores de temperatura em uma leira de compostagem, mas já demonstra ser uma adição interessante para o sistema já existente, gerando uma economia de tempo para os usuários.

**PALAVRAS-CHAVE:** Datalogger, Arduino, Linguagem de programação.

#### **ABSTRACT**

Process automation has become increasingly important in the search for an optimization of the performance of professionals involved in the most diverse areas. Such automation allows the professional not to be forced to perform tasks that are often repetitive and laborious and can focus on more complex points of their work. With this in mind, the idea of automating the process of organizing the data obtained by the Arduino-based collection system developed at UTFPR arose. The system showed satisfactory results,



SEI-SICITE 2021

Pesquisa e Extensão para um  
mundo em transformação

XI Seminário de Extensão e Inovação  
XXVI Seminário de Iniciação Científica e Tecnológica  
08 a 12 de Novembro - Guarapuava/PR



being used in course conclusion works, scientific initiations and other extension projects. The result was software in Python language that is able to read the text file generated by the automated data collection equipment and perform a basic treatment, reorganizing the information in an electronic spreadsheet and calculating descriptive statistics, such as mean, mode and median, in addition to plotting a graph containing the distribution of values. The software was tested with data collected in compliance with a discipline in obtaining temperature values in a compost windrow, but it already proves to be an interesting addition to the existing system, saving time for users.

**KEYWORDS:** Datalogger; Arduino; Programming language.



## INTRODUÇÃO

A evolução da eletrônica causou uma saturação do mercado de equipamentos para coleta de dados. No entanto, estes permanecem apresentando valores muito elevados, tornando-os inacessíveis a estudantes e pesquisadores (Azúa-Barrón et al., 2017). Neste sentido, o desenvolvimento de alternativas de baixo custo se faz cada vez mais necessário para atender a esta demanda.

Uma solução comum é o uso de plataformas e linguagens de programação open source, que apresentam custos reduzidos em relação aos artigos comerciais e permitem a obtenção de sistemas com baixo custo e condizentes com os orçamentos geralmente disponíveis para pesquisa. Dentre as muitas saídas geralmente adotadas neste sentido está o Arduino, definido como uma plataforma de prototipagem dotada de uma estrutura de software e um ambiente de desenvolvimento integrado (Dolinay et al., 2016).

Com base no Arduino e nos diversos componentes eletrônicos que podem ser utilizados anexos a ele, foi desenvolvido ao longo dos anos o projeto Sistema Open-Source e de Baixo Custo para Checagem de Dados em Tempo Real (Vieira, 2020), um sistema capaz de realizar a leitura e armazenamento de dados, além de transmiti-los via bluetooth. O sistema utiliza um conjunto de sensores de temperatura, os quais são distribuídos de forma a realizar leituras de diferentes pontos. As leituras são obtidas a cada 10 minutos, frequência recomendada pelos trabalhos de Dal Bosco para composteiras (Dal Bosco et al., 2018) e adotado pelo projeto para outras situações pelo equilíbrio entre volume de dados e amplitude de tempo coberto, e armazenadas no cartão SD e podem ser conferidas ou transferidas através de um sistema bluetooth.

Este sistema já foi utilizado no auxílio de diversos projetos dentro e fora do Campus Londrina da UTFPR, dentre os quais pode-se citar outros projetos de extensão, projetos de iniciação científica e trabalhos de conclusão de curso desenvolvidos em parceria com indústrias da região, tendo sua funcionalidade e eficiência comprovadas, como mostram os trabalhos de Vieira (2020) e Bachega (2020).

No entanto, os dados obtidos por este processo são armazenados na forma de arquivos de texto (txt), os quais necessitam passar por um tratamento para serem realocados em planilhas eletrônicas e só então serem analisados pelos pesquisadores. Tal processo representa um gasto de tempo significativo dos pesquisadores e alunos atendidos pelo projeto, que pode se tornar um empecilho para o uso do sistema.

Para atender a esta necessidade, foi proposta a criação de um software capaz de automatizar esta tarefa e permitir que os usuários possam de fato se dedicar diretamente à análise dos dados adquiridos pelo sistema. Para a criação de tal programa foi escolhida a linguagem Python, que ocupa uma parcela crescente entre as mais utilizadas no mundo e que possui uma ampla comunidade que disponibiliza grande volume de tutoriais e de bibliotecas para as mais diversas aplicações.

Este trabalho trata do desenvolvimento e dos resultados obtidos até o momento com o software, além de apresentar ideias já propostas para futuras melhorias que permitirão a ele atender de forma mais completa e eficiente os usuários do sistema.

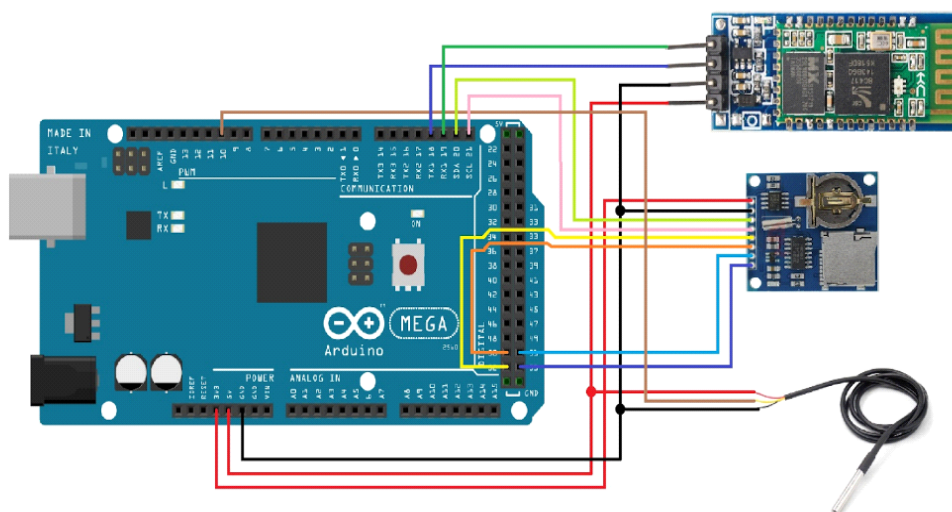
## MATERIAIS E MÉTODOS

O sistema que realiza a medição de temperaturas é apresentado na Figura 1 e utiliza como base um microcontrolador Arduino Mega 2560 ao qual foram ligados um módulo Bluetooth HC-06, 10 sensores de temperatura DS18B20 encapsulados com aço inoxidável e um datalogger RTC/SD, que é formado por um módulo RTC (Real Time Clock) do modelo RTC DS1307 acoplado a um módulo SD (Secure Digital) que permite a gravação dos dados obtidos em um cartão micro SD.



O software do sistema foi desenvolvido na própria IDE (Integrated Development Environment) do Arduino e é capaz de executar todas as tarefas relacionadas à coleta, transferência de dados via bluetooth e armazenamento de dados em cartão micro SD. Os dados utilizados para o desenvolvimento deste trabalho foram obtidos junto a um projeto que analisa a degradação de biopolímeros submetidos ao processo de compostagem de resíduos orgânicos residencial e serragem que foi atendido pelo projeto de extensão intitulado Assistência a Projetos no Processo de Coleta de Dados Utilizando Sistemas Open Source.

Figura 1 – Sistema montado com seus componentes.



Fonte: Vieira (2020).

Para desenvolvimento do software que faz o tratamento prévio dos dados coletados foi escolhida a linguagem de programação Python, a qual tem tido um destaque cada vez maior nas áreas de automação, Data Science e inteligência artificial por apresentar uma combinação de simplicidade para o uso e potencial de criação de ferramentas complexas, além da numerosa comunidade que cria tutoriais descrevendo diversas soluções com a linguagem e um grande número de bibliotecas que facilitam a criação e execução de uma infinidade de funções, como mostrado pelos trabalhos de Vicentini, que criou uma rede neural para regressões não lineares na área de química (Vicentini, 2021) e de do Couto, que utiliza uma biblioteca da linguagem para cálculo de incertezas experimentais na física (do Couto, 2021).

A versão da linguagem utilizada durante o desenvolvimento foi a Python, em sua versão 3.9, no ambiente de programação Visual Studio Code (ou apenas VS Code), versão 1.6, desenvolvido pela Microsoft e disponível gratuitamente na internet. A escolha deste ambiente ao invés da IDE do próprio Python se deu pelas ferramentas únicas apresentadas por ele que facilitam a manipulação de variáveis e de bibliotecas da linguagem.

O software desenvolvido conta com o uso de bibliotecas desenvolvidas pela comunidade, sendo elas as bibliotecas email, imaplib, math, os, re, openpyxl e matplotlib, as quais adicionam ferramentas importantes no que diz respeito à manipulação de informações de e-mails, funções matemáticas, interações com o sistema operacional e com o programa de planilha eletrônica Excel e funções para criação de gráficos.

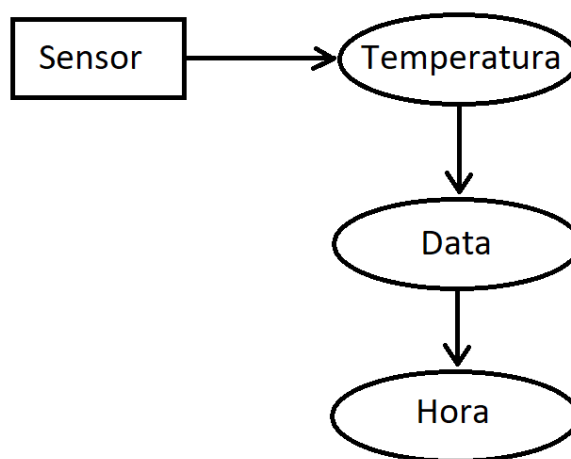
Ao ser executado, o programa traz ao usuário algumas opções (Figura 2), sendo ela a criação de uma base de dados com todos os arquivos contidos no e-mail cadastrado no programa, atualizar uma base de



dados já existente adicionando os dados de e-mails não lidos, criar uma planilha eletrônica com a base de dados já existente ou criar uma base de dados e uma planilha.

A base de dados criada se trata de um arquivo no formato txt que pode ser posteriormente lido pelo software para a criação de uma planilha eletrônica como a apresentada pela Figura 3. A planilha eletrônica criada apresenta todos os dados coletados pelos sensores, separando cada dia em uma aba diferente e separando os dados por sensor e por hora, além de realizar um tratamento estatístico descritivo.

Figura 2 – Diagrama Entidade Relacionamento da base de dados.



Fonte: Autoria Pópria.

Figura 3 – Interface do ambiente de programação ao executar o programa.

```
As ações disponíveis são:  
1 - Criar banco de dados com todos os e-mails  
2 - Atualizar banco de dados com e-mails não lidos  
3 - Criar planilha com o banco de dados existente  
4 - Criar banco de dados com todos os e-mails e criar planilha  
0 - Encerrar
```

Fonte: Autoria Própria.

Além das estatísticas por leitura, são geradas estatísticas gerais do dia para cada sensor e para os dados como um todo, como é exemplificado pela Figura 4. Outra funcionalidade importante é a criação dos gráficos com todas as leituras realizadas ao longo do dia (Figura 5), que permite uma melhor visualização do comportamento das temperaturas coletadas.



Figura 4 – Aba da planilha eletrônica apresentado os dados coletados e as estatísticas por hora e sensor.

Data	Hora	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8	Sensor 9	Sensor 10	Média	Variância	Desvio Padrão
2020/5/29	8:0:46	23	22,5	12	23,5	22	22	23,5	22,5	16,5	21	20,8	13,31	3,648287269
2020/5/29	8:10:47	23	22,5	12,5	23,5	22	22	23,5	22,5	16,5	21	20,9	11,54	3,39705755
2020/5/29	8:20:47	23	22,5	12,5	23,5	22	22	23,5	22,5	17	21	20,95	11,1225	3,335041229
2020/5/29	8:30:48	23	22	13	23,5	22	22	23,5	22,5	17	21	20,95	10,1725	3,189435687
2020/5/29	8:40:48	23	22	13,5	23,5	22	22	23,5	22,5	17	21	21	9,4	3,065941943
2020/5/29	8:50:49	22,5	22	14	23,5	22	22	23,5	22,5	17	21	21	8,5	2,915475947
2020/5/29	9:0:49	23	22	14	23,5	22	22	23	22,5	17,5	21	21,05	8,0725	2,841214529
2020/5/29	9:10:50	22,5	22	14,5	23	22	22	23	22,5	17,5	21	21	7	2,645751311
2020/5/29	9:20:50	22,5	22	15,5	23	22	22	23	22,5	17,5	21	21,1	5,79	2,406241883
2020/5/29	9:30:51	22,5	22	17	23	22	22	23	22,5	17,5	21,5	21,3	4,31	2,076053949
2020/5/29	9:40:51	22,5	22	17,5	23	22	22	23	22,5	17,5	21,5	21,35	3,9025	1,975474627
2020/5/29	9:50:52	22,5	22	18	23	22	22	23	22,5	17,5	21,5	21,4	3,54	1,881488772
2020/5/29	10:0:52	22,5	22	19,5	23	22	22	23	22,5	18	21,5	21,6	2,34	1,529705854
2020/5/29	10:10:53	22,5	22	21,5	23	22	22	23	22,5	18	21,5	21,8	1,86	1,36381817
2020/5/29	10:20:53	22,5	22	22,5	23	22	22	23	22,5	18,5	21,5	21,95	1,5225	1,233896268
2020/5/29	10:30:54	22,5	22	23	23	22	22	23	22,5	18,5	21,5	22	1,6	1,264911064
2020/5/29	10:40:54	22,5	22	22	23	22	22	23	22,5	19	21,5	21,95	1,1725	1,082820391
2020/5/29	10:50:55	22,5	22	22,5	23	22	22	23	22,5	19,5	21,5	22,05	0,9225	0,960468636
2020/5/29	11:0:55	22,5	22	22,5	23	22	22	23	22,5	20	21,5	22,1	0,69	0,830662386
2020/5/29	11:10:56	22,5	22	23	23	22	22	23	22,5	20	21,5	22,15	0,7525	0,867467579
2020/5/29	11:20:56	22,5	22	23,5	23	22	22	23	22,5	20	22	22,25	0,8125	0,901387819

Fonte: Autoria Própria.

Durante a fase de testes na qual o programa se encontra, foram utilizados arquivos de testes com dados simulados e erros intencionais para que seja possível observar o comportamento do software diante de possíveis falhas na coleta de dados e arquivos corrompidos. Além disso, foram utilizados dados de projetos atendidos anteriormente com o objetivo de comparar as planilhas obtidas pelo programa com as feitas à mão por pesquisadores e apontar possíveis melhorias.

O passo futuro é a aplicação do programa no formato atual por usuários externos. Desta forma, será possível obter um retorno em questões como facilidade de utilização, desempenho em computadores diversos e possíveis aprimoramentos na planilha resultante.

Figura 5 – Estatísticas gerais e de cada sensor ao longo de um dia.

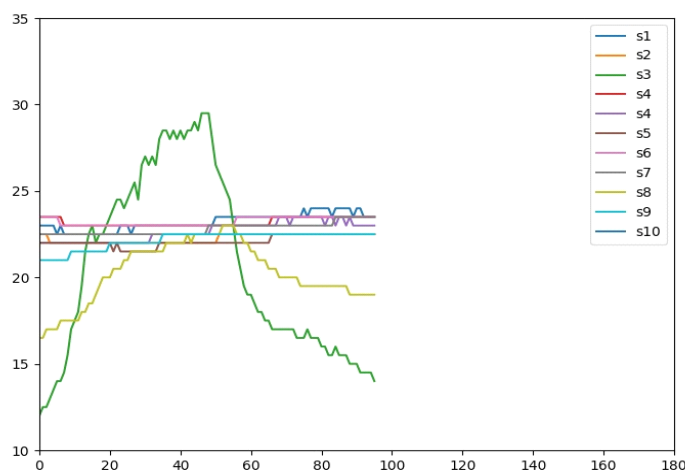
Média do dia	Variância do dia	Desvio Padrão do dia
22,2515625	4,248435059	2,061173224
Número do Sensor	Média	
1	23,21875	
2	22,25	
3	20,60416667	
4	23,19270833	
5	22,66666667	
6	22,09375	
7	23,23958333	
8	22,8125	
9	20,27083333	
10	22,16666667	

Fonte: autoria Própria.





Figura 6 – gráfico de comportamento diário dos sensores.



Fonte: Autoria Própria.

## RESULTADOS E DISCUSSÕES

De forma geral, o software foi utilizado apenas em caráter de testes até o momento, utilizando quantidades limitadas de dados em relação ao total obtido, sendo que as planilhas geradas pelo mesmo são comparadas às feitas pelos pesquisadores.

Os resultados até o momento são promissores, sendo o software capaz de filtrar os e-mails contendo os arquivos coletados pelo sistema, criar uma base de dados leve, pois tudo é compactado em um único arquivo de texto, e criar, quando solicitado, uma planilha com todos os dados organizados de forma prática e concisa, bem como realizar um tratamento estatístico inicial de forma extremamente rápida, levando cerca de 3 segundos para realizar todos os dados obtidos pelo sistema ao longo de 12 dias, poupando tempo dos pesquisadores e permitindo a eles trabalhar de forma mais focada na análise dos dados obtidos.

Duas futuras adições para o software já são discutidas, sendo elas a função de atualizar uma planilha já existente e a transformação do código em um software independente, que possa ser executado sem a necessidade de instalação do ambiente de programação básico do Python e suas bibliotecas, facilitando ainda mais sua utilização pelos projetos atendidos.

## CONCLUSÃO

Ao sair da fase de testes e passar a ser parte integrante do material oferecido aos pesquisadores que são atendidos pelo projeto, o software será uma grande adição, aumentando o nível total de automação da obtenção de dados, mantendo a premissa do baixo custo.

## AGRADECIMENTOS

Os autores agradecem a Fundação Araucária pelo suporte, por meio da bolsa de extensão concedida e a UTFPR-PROREC pelo suporte.



## REFERÊNCIAS

Azúa-Barrón, M.; Vázquez-Peña, M. A.; Arteaga-Ramírez, R.; Hernández-Saucedo, R. **Sistema de adquisición de datos de bajo costo con la plataforma arduino**. Revista mexicana de ciencias agrícolas, v.8, n.1, p.1-12. 2017.

Bachega, João Paulo, et al. Redução de custos na assistência a projetos no processo de coleta de dados. In: Seminário de Extensão e Inovação da UTFPR, 10ª Edição, 2020, Toledo, Paraná. Anais do congresso, Toledo, 2020

DAL BOSCO, Tatiane C. et al. The ideal frequency of temperature data collection in compostability experiments on domestic organic residues. **Environmental Technology**, v.39. 2018. Doi: <https://doi.org/10.1080/09593330.2018.1523233>

Dolinay, Jan; Dostálek, Petr; Vašek, Vladimír. **Arduino Debugger**, IEEE EMBEDDED SYSTEMS LETTERS, Vol.8(4), pp.85-88, DECEMBER 2016.

do Couto, Flávio M. e da Silva, Julio F. Uso do Módulo Python Uncertainties no cálculo de incertezas experimentais da diferença de potencial e corrente elétrica de um protótipo experimental. Revista Brasileira de Ensino de Física [online]. 2021, v. 43 [Acessado 14 Outubro 2021], e20200415. Disponível em: <https://doi.org/10.1590/1806-9126-RBEF-2020-0415>. Epub 01 Mar 2021. ISSN 1806-9126. <https://doi.org/10.1590/1806-9126-RBEF-2020-0415>.

Vicentini, Eduardo D. e Oliveira-Filho, Antonio G. Sampaio de INTRODUÇÃO ÀS REDES NEURAIS PARA REGRESSÕES NÃO-LINEARES: AJUSTE DE SUPERFÍCIES DE ENERGIA POTENCIAL. Química Nova [online]. 2021, v. 44, n. 2 [Acessado 14 Outubro 2021], pp. 229-234. Disponível em: <https://doi.org/10.21577/0100-4042.20170650>. Epub 07 Maio 2021. ISSN 1678-7064. <https://doi.org/10.21577/0100-4042.20170650>.

Vieira, Guilherme Quereza, et al. Sistema open-source e de baixo custo para checagem de dados em tempo real. In: Seminário de Extensão e Inovação da UTFPR, 10ª Edição, 2020, Toledo, Paraná. Anais do congresso, Toledo, 2020.

Documentação e Download de Softwares, Linguagem e Bibliotecas Usadas:

Python 3.9: <https://www.python.org>

Virtual Studio Code 1.6: <https://code.visualstudio.com>

Biblioteca email 3.6: <https://docs.python.org/pt-br/3/library/email.html>

Biblioteca Imaplib 3.9: <https://docs.python.org/3/library/imaplib.html>

Biblioteca math 3.5: <https://docs.python.org/pt-br/3/library/math.html>

Biblioteca os 3.1: <https://docs.python.org/pt-br/3/library/os.html>

Biblioteca re 3.5 <https://docs.python.org/pt-br/3/library/re.html>

Biblioteca openpyxl 3.0.9: <https://openpyxl.readthedocs.io/en/stable/>

Biblioteca matplotlib 3.4.3: <https://matplotlib.org>