



ConBRepro

XIII CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO



IA nas Engenharias

29 nov. a 01 de dezembro 2023

Linguagem para aquisição de dados programada por fluxo de atividade

Vinicius Melchior Liborio Santos

ICET - UFAM

Carlos Alberto Oliveira De Freitas

ICET - UFAM

Emerson Marcos Pereira

ICET - UFAM

Resumo: A pesquisa desenvolvida tem como base a construção das estruturas de linguagens de programação com o objetivo de apresentar um novo modelo, com melhor compreensão de dados e de forma que exerça suas funcionalidades com acerto. Atualmente, o mundo vem se tornando cada vez mais digital, computadores e sistemas estão mais presentes em todos os setores, principalmente, nas áreas de trabalho. Com o advento tecnológico, profissionais de diversas áreas necessitam ter algum conhecimento de TI para o manuseio de programas, Sistemas Operacionais, que pertencem ao estudo da computação, mas, o intuito desta pesquisa é poder tornar o uso de linguagens de programação aplicadas a instrumentos com diversos objetivos funcionais, onde terão de ser manuseadas e utilizadas por alguém que não tem o conhecimento necessário, mas que saberá interagir com o sistema com mais facilidade e êxito na resolução de problemas. O projeto consiste em uma linguagem com semântica simplificada para facilitar sua utilização por leigos em programação, funcionará para a obtenção de diversos tipos de dados físicos, como pressão, temperatura, de forma que o esclarecimento da execução de suas atividades proporcionando um fácil entendimento do funcionamento da linguagem e, para isso, será tomado como base, linguagens existentes já aplicadas que são utilizadas com frequência.

Palavras-chave: Linguagem, Programação, Sensores, Fluxo de atividade.

Language for data acquisition programmed by activity flow

Abstract: The research undertaken is based on the construction of programming language structures with the aim of presenting a new model that enhances data understanding and ensures accurate functionality. Presently, the world is becoming increasingly digital, with computers and systems more prevalent in all sectors, particularly in the workplace. With technological advancements, professionals from various fields require some IT knowledge for managing programs and operating systems, which are part of the field of computer science. However, the purpose of this research is to make the use of programming languages applicable to instruments with various functional objectives, which must be handled and operated by individuals lacking the necessary knowledge. These individuals will be able to interact with the system more easily and successfully in problem-solving. The project entails the development of a language with simplified semantics to facilitate its use by programming novices. It will function for acquiring various types of physical data, such as pressure and temperature, while clarifying the execution of its activities to

provide an easy understanding of the language's operation. To achieve this, existing languages that are frequently applied will serve as a foundation.

Keywords: Language, Programming, Sensors, Workflow

1. Introdução

O caminho das linguagens de programação até ao que é conhecido no presente, mostra o quanto a ciência computacional evoluiu, ao ponto do ser humano se tornar altamente dependente das tecnologias para seus diversos trabalhos. No decorrer da história, é possível compreender que a criação de algoritmos previa uma necessidade de obter melhor compreensão do manuseio das máquinas, tanto a nível hardware quanto de software (SANTOS, 2006).

A programação passa pela premissa de que para existir um tipo software, ele precisa passar por um processo de programação para verificar-se, portanto, é possível afirmar que todas as máquinas possuem instruções inseridas por meio de diversas linguagens de programação, criadas ao longo do tempo (ALBUQUERQUE; et al, 2019). As linguagens principais mais usadas atualmente são a C, C+, Python, Pascal e Java, o estudo da programação não se restringe a apenas esses tipos, mas se tornam pilares para que outras línguas possam ser produzidas.

A evolução da tecnologia trouxe a possibilidade de resolver problemas complexos. Mas a tecnologia é também cada vez mais indispensável para a resolução de problemas simples do quotidiano. A cada dia, mais as pessoas estão interagindo em seus lares ou empresas, com computadores, dispositivos móveis e robôs (VICARI, et al, 2018). Atualmente segundo a pesquisa salarial brasileira realizada pelo canal código fonte é notório que o Brasil possui um déficit de profissionais na área de tecnologia da informação, devido a isso os atuantes em áreas adjacentes a TI precisam se modernizar de forma a acompanhar o desenvolvimento dos processos internos de cada organização. Para auxiliar no acompanhamento da modernização por meio da diminuição da curva de aprendizado foi desenvolvida a proposta de linguagem para auxiliar na aquisição de dados relacionados a área de atuação do profissional leigo em programação.

A compreensão do mundo digital é importante para que o profissional possa apropriar-se dos processos que nele ocorrem, podendo compreender e criticar tendências, sendo ativo nesse cenário. Para uma compreensão estruturada do mundo digital, e não apenas efêmera e permeada de tecnologias (VICARI, et al, 2018). Tendo em vista os profissionais leigos em TI a linguagem proporciona de maneira simplificada a sua comunicação com a modernização ocorrida nos processos que são de sua competência, objetivando o esclarecimento da execução de suas atividades de forma a proporcionar um fácil entendimento do funcionamento da linguagem, para isso será tomado como base linguagens existentes já aplicadas que são utilizadas com frequência.

1.1 Objetivos

- Geral:

Realizar pesquisas para embasar o desenvolvimento de uma linguagem para aquisição de dados programada por fluxo de atividade que possa ser utilizada por profissionais técnicos que não conhecem a programação de baixo e alto nível.

- Específicos:

- Pesquisar e selecionar linguagens que sejam compatíveis com a ideia central da proposta classificando-as.
- Arquitetar a estrutura da linguagem a ser desenvolvida com base nas linguagens selecionadas.

→ Desenvolver a linguagem executando testes verificativos e exploratórios juntamente com a análise dos resultados da aplicação da linguagem.

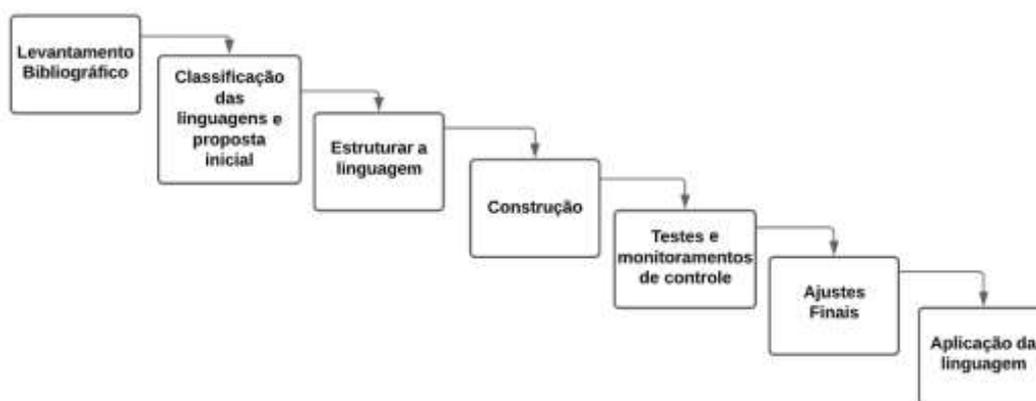
O artigo é dividido em sete seções que seguem a metodologia utilizada: Introdução, Metodologia, Levantamento bibliográfico e classificação das linguagens, Estrutura da linguagem, Construção, Aplicação da linguagem e Conclusão. Introdução: Aborda a importância da tecnologia na sociedade atual e a necessidade de uma linguagem simplificada, a Metodologia: Descreve as etapas do projeto, Levantamento bibliográfico e classificação de linguagens: Detalha como foram selecionadas as linguagens relevantes, Estrutura da linguagem: Explora a divisão entre o front-end e o back-end da linguagem, Construção: Descreve a fase prática do projeto, incluindo o desenvolvimento dos módulos front-end e back-end, Aplicação da linguagem: Destaca a aplicação prática da linguagem em um cenário de teste e por fim a Conclusão: Resume os principais resultados e sugere áreas de pesquisa futura.

w

2. Metodologia

A metodologia escolhida para desenvolvimento deste trabalho foi dividida em 7 etapas, na fig. 1, está representado por um fluxograma da metodologia geral adotada para o projeto.

Figura 1. Metodologia Geral



Fonte: O Autor(2023)

- Etapa 1- Levantamento Bibliográfico

Realização de pesquisas bibliográficas relacionadas ao tema do trabalho para formar a fundamentação teórica e fornecer uma estrutura conceitual para o desenvolvimento do projeto. Utilizando plataformas científicas como SciELO (Scientific Electronic Library Online), Google Acadêmico, Portal Capes, entre outros.

- Etapa 2 – Classificação das linguagens e proposta inicial

Utilizando-se dos resultados obtidos com o levantamento bibliográfico e da classificação de trabalhos semelhantes adquiridos foi apresentada uma proposta inicial com especificações para a construção da linguagem.

- Etapa 3 – Estruturar a linguagem

A partir das especificações obtidas na etapa anterior deu-se início a documentação e estruturação da linguagem juntamente com o levantamento de requisitos iniciais.

- Etapa 4 - Construção

Após a estruturação da linguagem foi iniciada a etapa de construção com base nas pesquisas realizadas e os requisitos iniciais levantados. Esta possui Três atividades distintas: o desenvolvimento do módulo front-end que é a interface da linguagem com o usuário, o desenvolvimento do módulo back-end que é a interface da linguagem com os equipamentos e sensores e finalmente a integração destes dois módulos integrantes da linguagem.

- Etapa 5 – Testes e monitoramentos de controle

Realização de uma bateria de testes para verificar se o projeto está conforme a especificação da documentação dos requisitos.

- Etapa 6 - Ajustes Finais

Etapa necessária para correção de possíveis erros e não atendimentos as especificações dos requisitos.

- Etapa 7 - Aplicação da linguagem

A linguagem foi implantada em alguns cenários para a checagem do funcionamento.

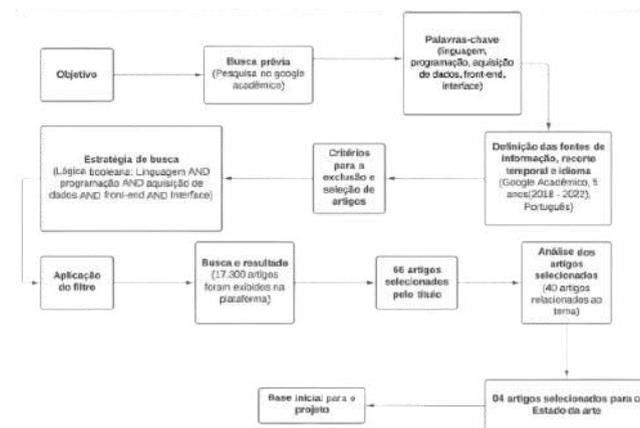
3. Levantamento bibliográfico e classificação das linguagens

Foram realizadas pesquisas bibliográficas relacionadas ao tema do trabalho com o objetivo de estabelecer uma sólida fundamentação teórica e fornecer uma estrutura conceitual para o desenvolvimento do projeto e para a coleta de informações destinadas ao desenvolvimento tanto do front-end quanto do back-end, foram conduzidas pesquisas utilizando lógica booleana (Desenvolvimento AND Linguagem) no Google Acadêmico devido sua ampla cobertura, diversidade de idiomas facilidade de uso e recursos de citação. Esse método foi adotado para verificar bibliografias que apresentassem semelhança com o tema principal e identificar potenciais artigos ou trabalhos que pudessem servir como referência para o projeto de pesquisa.

No front-end, a plataforma mostrou cerca de 17.300 resultados, observando a abrangência do termo adotado e do quantitativo de resultados, assim se fez necessário a utilização de filtro para a escolha dos artigos ideais ao projeto. os artigos retirados para realização do levantamento bibliográfico para o front-end, foram coletados no banco de dados do Google Acadêmico, no qual foi utilizado o parâmetro de pesquisa escolhido. Foram adotados os seguintes critérios: duplicidade (trabalhos duplicados nas bases), diferente idioma do requerido, estudos que não tratem do tema mesmo que pertençam aos resultados, não possui uma completude para ser utilizado como base para o projeto.

Para o Back-End foi empregado palavras-chave relevantes, como "linguagem", "aquisição de dados" e "programação", a fim de identificar trabalhos semelhantes e classificá-los por meio de uma análise criteriosa. Essa abordagem visa construir uma base sólida de conhecimento e reforço para as etapas subsequentes do projeto. Inicialmente, a pesquisa foi conduzida no Google Acadêmico, com limitações temporais e de idioma, utilizando uma estratégia de busca baseada na lógica booleana (OR, AND). Critérios foram estabelecidos para excluir resultados irrelevantes e selecionar artigos pertinentes ao projeto. Por fim, foram selecionados 04 artigos voltados para o Front-End e 05 artigos para o Back-End, todos compondo a bibliografia do projeto. O esquema abaixo na figura 2 contém os passos feitos para a obtenção de dados do front-end

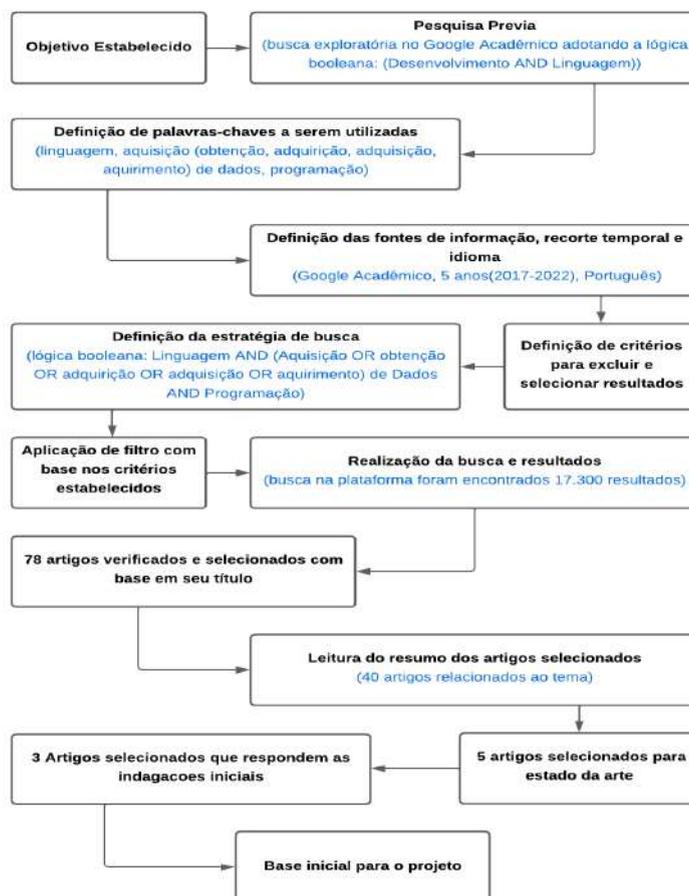
Figura 2. Fluxograma de pesquisa do Front-end



Fonte: O Autor(2023)

Na figura 3 estão registrados os processos de obtenção de dados do back-end que compõem a bibliografia.

Figura 3. Fluxograma de pesquisa do Back-end.



Fonte: O Autor(2023)

Após a realização da pesquisa bibliográfica foram selecionadas três linguagens compatíveis com o projeto, onde sua classificação foi realizada devido sua relevância para o projeto. Os trabalhos relacionados encontram-se destacados abaixo na tabela 1.

Tabela 1: Linguagens selecionadas.

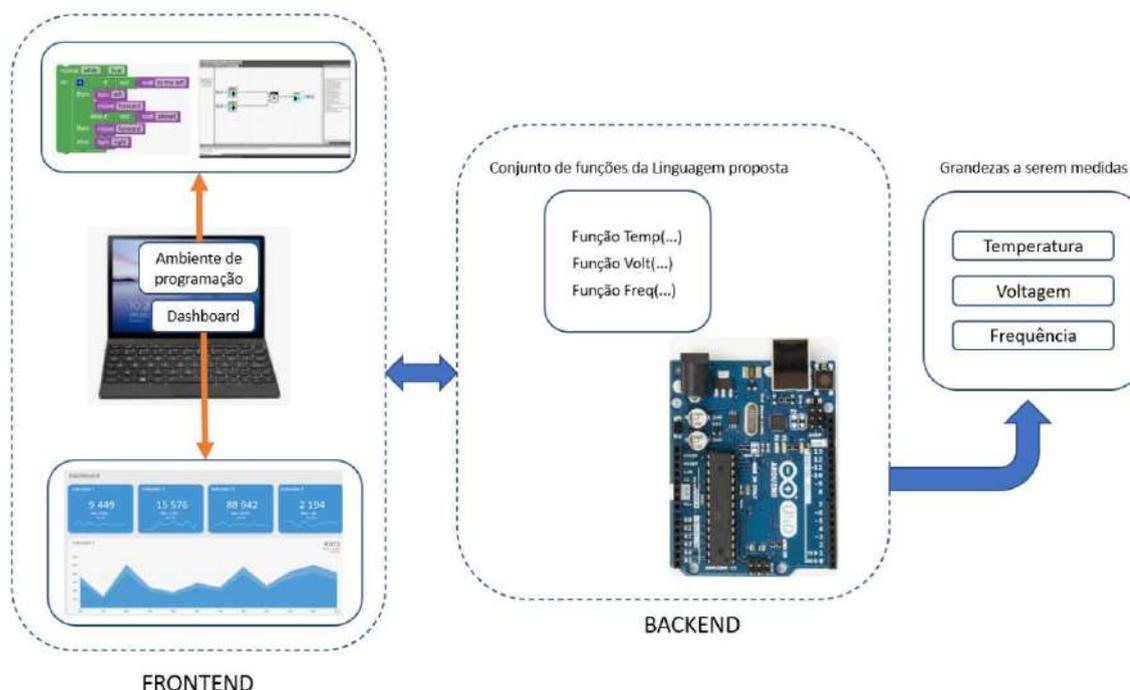
Classificação	Título	Autor
1º	Plataforma de hardware e software para aquisição e disponibilização de dados utilizando computação em nuvem	Silva, Salvador Ramos Bernardino
2º	Interpretador de Portugol.	R. Santiago CTTMar/UNIVALI, e R. L. S. Dazzi
3º	Software adaptativo: método de projeto, representação gráfica e implementação de linguagem de programação	Silva, Salvador Ramos Bernardino da

Fonte: O Autor(2023)

4. Estrutura da linguagem

A estruturação inicial da proposta foi adotado uma abordagem dinâmica e com duas vertentes, o back-end e o front-end, desta forma objetivando uma visão mais completa e especialização em cada processo, ou seja, de forma independente e eficiente em cada vertente, e facilitando a manutenção e escalabilidade do projeto. Na figura 4 está demonstrado e ilustrado a ideia geral da proposta contendo a subdivisão do front-end contendo o ambiente de programação juntamente com o dashboard para visualização dos dados e back-end contendo funções de coleta de dados pré-programadas.

Figura 4. Estrutura da linguagem.



Fonte: O Autor(2023)

Inicialmente, será descrita a arquitetura do módulo geral do back-end, o qual é parte integrante do sistema proposto. Esse módulo tem como função principal efetuar o controle e aquisição de dados provenientes de sensores. Essa tarefa é realizada por meio da integração de sensores incorporados em placas de código aberto, como Tinker 's, Arduino e Raspberry, em conjunto com os parâmetros de controle da linguagem Python.

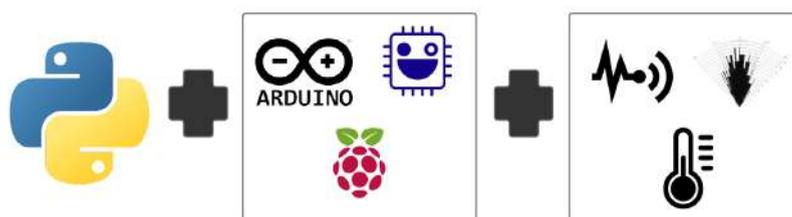
O propósito fundamental do presente módulo para o back-end é possibilitar a coleta e o armazenamento de dados provenientes de uma rede de microcontroladores. Cada dispositivo microcontrolador está equipado com sensores cujas medições são transmitidas ao módulo front-end por meio de uma comunicação estabelecida. Nesse contexto, uma rotina foi implementada para o requerimento dos dados coletados.

A linguagem Python desempenha um papel crucial ao atuar como intermediária entre o back-end e o front-end. Ela é responsável por tratar os dados coletados e carregar as funções encapsuladas para a placa controladora. Por sua vez, a placa controladora recebe os parâmetros de suas funções e se prepara para a aquisição de dados de sensores específicos, os quais estão previamente cadastrados na placa. Essa etapa permite descarregar os dados adquiridos de maneira eficiente.

A arquitetura do módulo front-end, desenvolvida em Python por meio da biblioteca tkinter, adota um modelo semelhante ao do módulo back-end, embora com algumas distinções. O módulo front-end é encarregado de apresentar, de maneira gráfica e interativa, os dados coletados pelo back-end, permitindo ao usuário visualizar e analisar esses dados. Além disso, o módulo front-end possibilita ao usuário a configuração dos parâmetros de controle do back-end, tais como a frequência de aquisição, o tipo de sensor e o endereço do microcontrolador.

Composta por três camadas, o módulo front-end se estrutura de forma organizada: a camada de interface, a camada lógica e a camada de comunicação. A camada de interface assume a responsabilidade pela interação com o usuário, a camada lógica se encarrega de processar as interações do usuário e, por fim, a camada de comunicação ou integração estabelece a comunicação com o módulo back-end. Através dessa camada de comunicação, os parâmetros de controle são enviados ao back-end, que, por sua vez, coleta os dados de acordo com os parâmetros recebidos e os transmite de volta ao front-end para visualização e análise pelo usuário. A Figura 5 abaixo representa os submódulos que compõem o back-end e front-end.

Figura 5. submódulos back-end e front-end.



Fonte: O Autor(2023)

5. Construção

Completa as estampas anteriores, iniciou-se a etapa de construção propriamente dita. Essa fase envolveu três atividades distintas: o desenvolvimento do módulo front-end, que diz respeito à interface da linguagem com o usuário; o desenvolvimento do módulo back-end, relacionado à interface da linguagem com equipamentos e sensores; e, por fim, a integração desses dois módulos onde todas as ideias e conceitos se materializam em um protótipo de linguagem funcional.

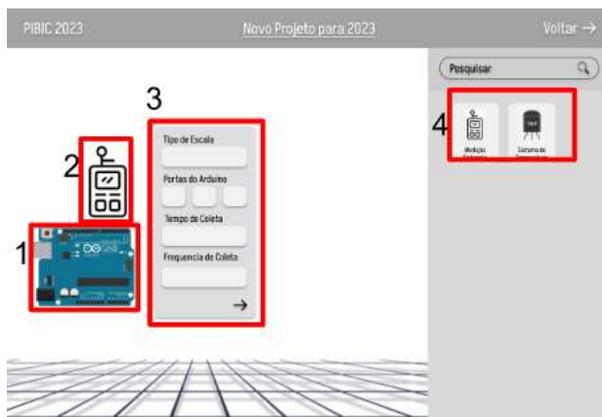
- Construção módulo front-end

Antes de iniciar a implementação do módulo front-end, realizou-se uma prototipagem da interface utilizando a ferramenta Figma. Essa etapa de prototipagem permitiu visualizar como seria a interface da linguagem, definir o layout, disposição de

elementos, cores, fontes e interações possíveis. Com oportunidade de integrar e refinar o design, levando em consideração feedbacks internos. Após a conclusão da prototipagem no Figma, deu-se início a implementação do módulo front-end utilizando a biblioteca Tkinter do Python. A implementação com Tkinter seguiu fielmente o design definido no protótipo, garantindo que a interface construída fosse coesa e atendesse às expectativas previamente estabelecidas. Abaixo estão as figuras 5 e 6 com a interface implementada.

Na figura 6 estão dispostos os elementos: 1- placa de arduino, 2 - ilustração do sensor de distância (Ultrassônico Hc-sr04), 3 - Definição de parâmetros do sensor de distância, 4 - Seletor de sensores(sensor de distância e sensor de temperatura).

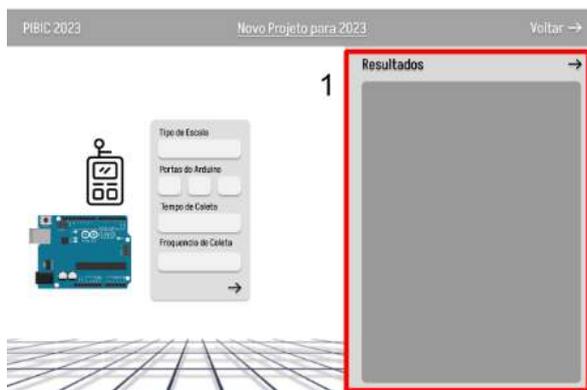
Figura 6. Definição de parâmetros do sensor.



Fonte: O Autor(2023)

Após as configurações demonstradas na figura 6 e o envio para a placa arduino é exibido uma label lateral para a exibição dos dados coletados(1), conforme a figura 7.

Figura 7. Resultados de coleta.



Fonte: O Autor(2023)

- Construção módulo back-end

A construção do back-end seguiu a construção lógica para a coleta de dados das placas de código aberto e sua interação com os sensores de coleta, integrou-se a placas de código aberto disponíveis no mercado, como Arduino, Raspberry Pi e Tinker board. O back-end estabeleceu gerenciamento e processamento dos dados coletados dos dispositivos por meio de uma estrutura para organizar as informações relevantes de forma a facilitar a manipulação e análise posterior.

- Integração dos módulos

Com a construção individual dos módulos concluída, a etapa de integração foi iniciada, com o principal objetivo de estabelecer a comunicação entre os módulos. Para

garantir uma integração bem-sucedida, foram estabelecidos protocolos de comunicação que permitissem a interação. Foi criada uma camada de comunicação que possibilitou a troca de dados e comandos entre as duas partes da linguagem.

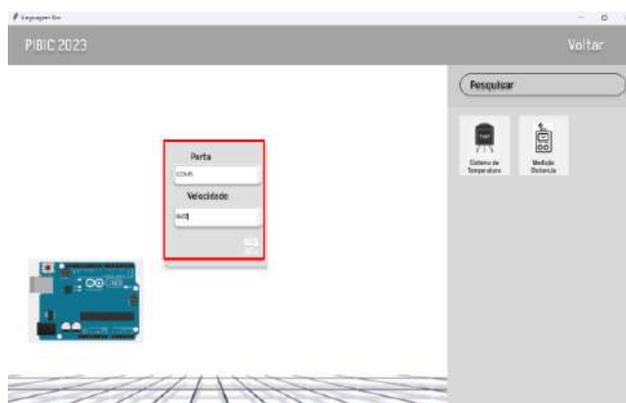
6. Aplicação da linguagem

A linguagem desenvolvida foi aplicada em cenários práticos para verificar seu funcionamento e eficácia. Nesta etapa, foi necessário avaliar a capacidade da linguagem de atender as especificações do projeto.

Para realizar a aplicação da linguagem, foram definidos cenários de teste que abrangem contextos relevantes à proposta. Esses cenários foram utilizados na etapa de testes para verificar diferentes aspectos da linguagem, como sua flexibilidade, desempenho, tratamento de exceções e facilidade de uso.

Exemplo de aplicação realizado, utilizando-se de dois sensores um para coletar a distância (Sensor ultrassônico HC-SR04) e outro para coletar a temperatura (DHT11), o experimento consistiu em coletar a variação de temperatura em relação a distância do objeto que fornecia calor. Primeiro configuramos a comunicação com a placa Arduino (COM5 e 9600) observe a figura 8.

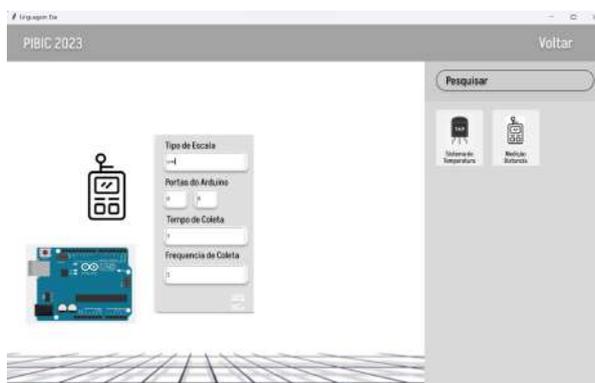
Figura 8. Configuração de conexão com Arduino



Fonte: O Autor(2023)

Após configurarmos a coleção com a placa foi configurado os sensores de distância e temperatura para realizar 5 coletas em 5 minutos, conforme mostra a figura 9 e 10, para o sensor de distância a coleta foi realizada em centímetros e conectado com a pinagem 8 e 9, para o sensor de temperatura sua coleta está em graus celsius na pinagem 6 e 7.

Figura 9. Configuração sensor de distância



Fonte: O Autor(2023)

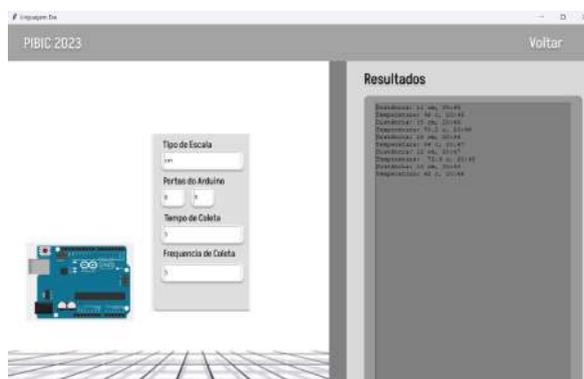
Figura 10. Configuração sensor de temperatura



Fonte: O Autor(2023)

Com tudo configurado enviamos os dados à placa e aguardamos o resultado da coleta, a figura 11 mostra o resultado da distância e da temperatura coletada do cenário montado onde o objeto que fornece o calor foi aproximado e afastado dos sensores para o teste.

Figura 11- Resultado da coleta dos sensores



Fonte: O Autor(2023)

Com a realização deste teste podemos retirar a seguinte relação de temperatura e distância da aplicação 11 cm e 96 c, 15 cm e 70.2 c, 10 cm e 94 c, 12 cm e 72.4 c, 10 cm e 92 c. Durante a aplicação, a coleta e análise dos dados em tempo real para identificar eventuais falhas, bugs e limitações, durante esta etapa relatou-se possíveis pontos de dificuldade na adoção da linguagem, devido a limitação de modelos dos sensores e placas para a aquisição de dados, porém salvo o êxito na compatibilidade com os sistemas operacionais Linux (Ubuntu e Debian) e Windows.

7. Conclusão

A crescente digitalização do mundo e a onipresença de computadores e os sistemas em nossa sociedade atual tornaram-se essenciais para compreender e utilizar linguagens de programação, mesmo para profissionais sem conhecimento técnico em programação. O desenvolvimento de uma linguagem com semântica simplificada e uma estrutura que facilite a interação com sensores e a obtenção de dados físicos tem o potencial de democratizar o acesso à tecnologia, tornando a coleta de dados mais acessível a um público mais amplo. Isso se torna especialmente relevante em um cenário onde há uma crescente demanda por profissionais de tecnologia da informação e onde a modernização dos processos se tornou uma necessidade urgente.

A aplicação prática dessa linguagem em diferentes cenários, exemplificada pelo uso de sensores de distância e temperatura, tem demonstrado sua eficácia na coleta e

processamento eficiente dos dados, estabelecendo-se como uma ferramenta útil para profissionais em diversas áreas relacionadas à tecnologia da informação, em um mundo cada vez mais orientado pela tecnologia. Assim, essa pesquisa apresentada neste trabalho possui grande importância na busca por soluções mais acessíveis e fáceis de usar, capacitando profissionais de diferentes campos a interagir com sistemas e sensores de maneira mais eficiente. No entanto, é importante ressaltar que esse trabalho também possui limitações, principalmente relacionadas à compatibilidade entre os sensores e as placas utilizadas, o que pode restringir sua aplicabilidade em determinados cenários. Além disso, a pesquisa se concentrou em um conjunto particular de tecnologias e sistemas operacionais, o que pode restringir sua aplicabilidade em situações mais abrangentes.

Para trabalhos futuros, sugere-se a expansão da compatibilidade da linguagem com uma variedade maior de sensores e placas, bem como a exploração de sua aplicação em outros sistemas operacionais. Além disso, a realização de estudos de caso mais abrangentes em diferentes ambientes e setores pode fornecer insights adicionais sobre a eficácia da linguagem em diversas situações práticas.

Referências

Livro:

ALBUQUERQUE; CARDOSO; Daisy, Sérgio. a historia da linguagem de programação. Genexatas, 2019. Disponível em: <<https://genexatas.com.br/a-historia-da-linguagem-de-programacao-parte-1/>>. Acesso em: 15, abril, 2022.

Capítulo de Livro:

VICARI, Rosa Maria; MOREIRA, Alvaro Freitas; MENEZES, Paulo Fernando Blauth. Pensamento computacional: revisão bibliográfica. 2018.

VILELA, Luis AA et al. Sistema para aquisição de dados de pressão e vazão usando microcomputador. Revista Brasileira de Agrocomputação, v. 1, n. 2, p. 25-30, 2001.

Artigo de Periódico:

BATISTA, João Paulo da Costa. Sistema de Aquisição de dados. 1990.

OLIVEIRA, Fabíola F. et al. Implementação da Linguagem Funcional Script. In: VI Congreso Argentino de Ciencias de la Computación. 2000.

ROCHA, Paulo Santana et al. Ensino e aprendizagem de programação: análise da aplicação de proposta metodológica baseada no sistema personalizado de ensino. RENOTE, v. 8, n. 3, 2010.

Monografia, Dissertação e Tese:

BETTI, Flavio. Desenvolvimento e Implantação de um Programa de Controle e Aquisição de Dados na Calibração de Instrumentos em Radiodiagnóstico. 2007. Tese de Doutorado. Tese de Doutorado. Universidade de São Paulo.

