Linguagem para Aquisição de Dados Programada por Fluxo de Atividade

Vinicius M. L. Santos[1], Anacilia M. C. de A. P. Vieira[1], Carlos A. O. de Freitas[1]

[1]Instituto de Ciências Exatas e Tecnologia - Universidade Federal do Amazonas (UFAM)
Rua Nossa Senhora do Rosário, 3863 - CEP 69.103-128 - Itacoatiara - AM

{vinicius.santos, anaciliacavalcante, carlosfreitas}@ufam.edu.br

Abstract. *The research aims to develop a language to facilitate the acquisition of physical data in IoT sensor networks, crucial in a digitized world with an increasing presence of computational systems. Professionals from various fields require IT knowledge, highlighting the need for accessible languages for beginners. The study proposes a simplified semantic language to capture data such as pressure and temperature, based on existing languages. The objective is to ensure that users easily understand the language, interact efficiently with the system, and successfully solve problems.*

Resumo. *A pesquisa visa desenvolver uma linguagem para facilitar a obtenção de dados físicos em redes de sensores IoT, crucial num mundo digitalizado com crescente presença de sistemas computacionais. Profissionais de diversas áreas precisam de conhecimentos em TI, destacando a necessidade de linguagens acessíveis para novatos. O estudo propõe uma linguagem de semântica simplificada para capturar dados como pressão e temperatura, baseada em linguagens existentes. O objetivo é garantir que usuários compreendam facilmente a linguagem, interajam eficientemente com o sistema e resolvam problemas com êxito.*

1. Introdução Contextualização

De acordo com [Santos et al. 2006], o avanço da ciência computacional tem levado o ser humano a depender cada vez mais das tecnologias para suas atividades. Isso é particularmente verdadeiro no contexto da área de tecnologia da informação, onde percebe-se a crescente dificuldade para se manter atualizado com as mudanças constantes na área, com isso diversas ferramentas surgem para auxiliar a adaptação de profissionais a essas mudanças.

É fato que o uso adequado dos computadores para a solução de problemas complexos exige um profundo conhecimento teórico, bem como habilidade para uso extensivo e aplicação eficiente dos avanços da teoria. No entanto, o que se observa é que a solução destes problemas nem sempre faz uso adequado das descobertas teóricas e dos avanços tecnológicos obtidos [Neto 2000]. Uma das dificuldades que é enfrentada na atualidade é a escolha da linguagem de programação mais adequada para cada tipo de problema, devido à diversidade de linguagens, cada uma com seus prós e contras, mas nenhuma capaz de atender universalmente às necessidades práticas de forma eficiente.

Devido a isso, é importante conhecer as características e os paradigmas das

diferentes linguagens de programação para saber qual dever ser a adequada para a resolução de determinado problema.

Intitulado como Low-code, nos últimos anos, temos testemunhado a emergência de uma nova categoria de ambientes de desenvolvimento de softwares [Vincent et al. 2020]. Esses ambientes não apenas prometem um significativo aumento na produtividade do desenvolvimento de software, mas também introduzem novas abordagens para promover a convergência entre a área de Tecnologia da Informação e áreas adjacentes, ao mesmo tempo em que capacitam os usuários reduzindo significativamente a curva de aprendizado. Essas plataformas são agora reconhecidas como Plataformas Low-Code (LCP), Plataformas de Aplicativos Low-Code (LCAP) e Plataformas de Desenvolvimento Low-Code (LCDP).

2. Objetivos

A pesquisa desenvolvida tem como base a construção de uma LCDP de uma nova linguagem estruturada em blocos e interatividade denominada Skye, onde por meio da semântica simplificada, busca a facilitação da utilização por leigos em programação e que permita a obtenção de diversos tipos de dados físicos, como pressão, distância, umidade e temperatura, coletados por meio de sensores e placas de código aberto. O objetivo é que a execução das atividades proporcione um fácil entendimento do funcionamento da linguagem e permita que os usuários interajam com o sistema com mais facilidade e êxito na resolução de problemas. A nova linguagem será aplicada em uma rede de sensores para coletar e processar dados físicos para que possam ser analisados em uma interface web.

3. Metodologia adotada

O presente projeto tem a finalidade de proporcionar a expansão da linguagem para ser usada por técnicos que não possuem conhecimento profundo em programação, a continuidade do desenvolvimento deste trabalho seguiu um método dividido em sete etapas sequenciais e complementares, conforme o fluxograma da metodologia adotada, demonstrado no esquema da Figura 1 abaixo.

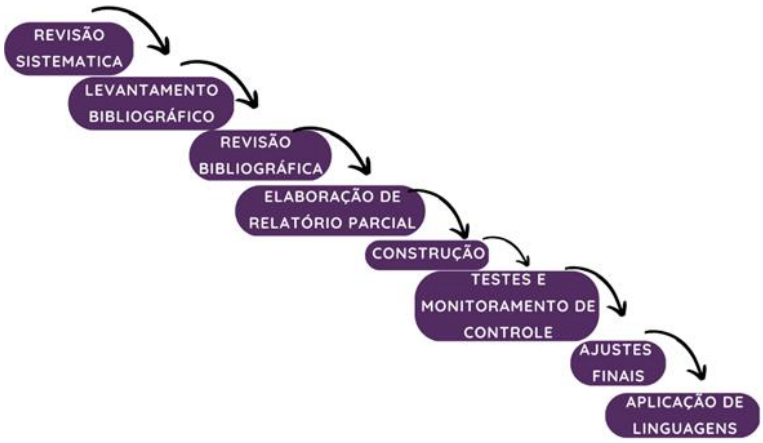

Figura 1. Metodologia

- Revisão Sistemática: Utilização de método para garantir resultados confiáveis.
- Levantamento Bibliográfico: Pesquisa em plataformas científicas para construir a base teórica.

- Revisão Bibliográfica: Análise da literatura existente para identificar lacunas de conhecimento.
- Construção: Desenvolvimento dos módulos Front End e Backend, além da integração desses módulos.
- Testes e Monitoramentos: Verificação da conformidade do projeto com as especificações.
- Ajustes Finais: Correção de erros e garantia de atendimento aos requisitos.
- Aplicação da Linguagem: Implementação em alguns cenários para verificar seu funcionamento.

4. Resultados

4.1. Procedimentos Metodológicos

Uma Revisão Sistemática de Literatura (RSL), compreendida por [Kitchenham 2004] como a possibilidade de associar e avaliar as evidências empíricas de um determinado campo de estudo a partir de análises disponíveis sobre os assuntos de interesse e através disso obtendo-se conclusões sobre as questões de pesquisa. O referido autor propõe algumas etapas para o desenvolvimento da RSL, as quais foram adotadas na presente pesquisa:

- Planejamento: Etapa que compreendeu a definição das Questões de Pesquisa (QPs) e criação de uma string para a busca de trabalhos relevantes sobre a temática em bases de dados;
- Condução: Etapa que compreendeu a aplicação da string de busca em bases de dados nacionais e internacionais e aplicação de filtros para ajudar na seleção do material, bem como critérios de inclusão e exclusão;
- Relato: Etapa que compreendeu a análise dos trabalhos selecionados e discussão acerca dos trabalhos encontrados para auxiliar a construção da linguagem.

4.2. Planejamento

A estrutura inicial de planejamento é delineada no Quadro 1, no qual as Questões de Pesquisa (QPs) e suas motivações correspondentes estão apresentadas.

Quadro 1: Questões de pesquisa e motivação

QP	Questões de pesquisa	Motivação
QP1	Como é realizada a construção de uma linguagem?	Investigar e compreender o processo envolvido na criação de uma linguagem.
QP2	Quais os métodos utilizados para sua implementação de uma linguagem?	Explorar as abordagens e técnicas empregadas no processo de implementação de uma linguagem.
QP3	Como é feita a aquisição de dados físicos em geral atualmente, quais as ferramentas utilizadas?	Explorar os métodos e ferramentas empregados no cenário contemporâneo para obter dados provenientes do mundo físico.

Após a delimitação das QPs e da motivação, foi realizada busca exploratória no Google Acadêmico devido sua abrangência de busca, adotando a lógica booleana:

(Desenvolvimento AND Linguagem) para verificar a existência de bibliografias sobre o assunto e verificar possíveis artigos/trabalhos que possam ser usados como referência para a pesquisa. Foram fornecidos pela plataforma aproximadamente 2.250.000 resultados, observando a abrangência do termo adotado e do quantitativo de resultados é necessário a aplicação de um filtro para trabalhos pertinentes ao projeto, porém a verificação de artigos existentes possibilitou a viabilidade da pesquisa.

4.3. Condução

As palavras-chave utilizadas em um levantamento foram: linguagem, aquisição (obtenção, adquirição, adquisição, aquirimento) de dados, programação. A palavra-chave "linguagem" foi escolhida pois o projeto consistirá em uma linguagem com semântica simplificada, logo a utilização do termo central é essencial. O termo "aquisição de dados" e seus sinônimos foi escolhido devido a proposta que diz respeito à obtenção de diversos tipos de dados físicos, como pressão e temperatura. Programação palavra-chave essencial ao projeto devido ao produto final do projeto se tratar de um software. A pesquisa bibliográfica foi realizada nas bases Google Acadêmico, IEEE Xplore e ScienceDirect, focando na engenharia de software. Foram considerados artigos dos últimos cinco anos, mas não se limitando a este recorte temporal, em português e inglês, para identificar lacunas e conceitos consolidados. A busca utilizou a lógica booleana: Linguagem AND (Aquisição OR obtenção OR adquirição OR adquisição OR aquirimento) de Dados AND Programação. Foram excluídos artigos duplicados, em idiomas diferentes, que não tratem do tema ou que não sejam completos, enquanto artigos fortemente relacionados foram incluídos.

4.4. Relato

A Figura 2 apresenta o resultado da quantidade de trabalhos encontrados após a aplicação da string de busca nas bases de dados. No levantamento foram encontrados 1358 artigos divididos nas três bases de dados. A partir desse total, foi realizada a discriminação através dos filtros e critérios de seleção pré-definidos, resultando em 46 artigos selecionados.

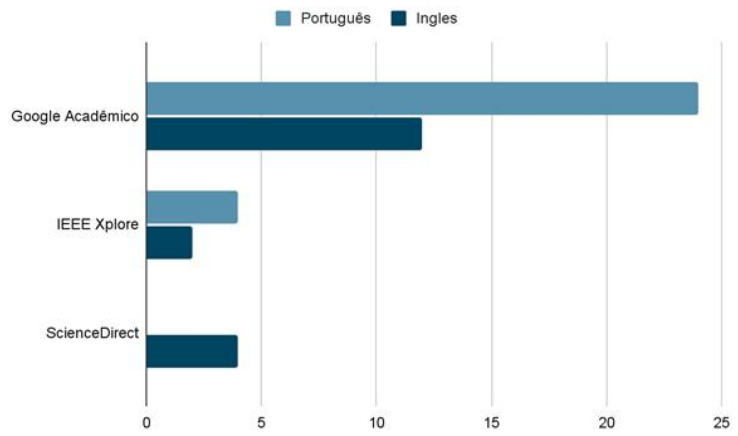

Figura 2. Quantidade de artigos selecionados por base de dados

4.6. Discussão

O desenvolvimento da Skye foi estruturado de acordo com a metodologia estabelecida, resultando no êxito da criação de uma linguagem de programação simplificada, acessível e eficiente para a coleta de dados físicos em uma rede de sensores IoT. A revisão sistemática e o levantamento bibliográfico forneceram uma base teórica para o desenvolvimento da Skye, onde foram analisados 46 artigos relevantes, dos quais 10 foram selecionados como principais referências para a construção da linguagem, devido a abordagem de temas relacionados a proposta, fornecendo a base utilizada para o desenvolvimento do projeto.

Com base nas pesquisas bibliográficas e nos requisitos iniciais, foi desenvolvida a Skye, uma linguagem estruturada em blocos com semântica simplificada, projetada para ser acessível a usuários com pouca ou nenhuma experiência em programação, principalmente pela linguagem permitir a coleta de dados físicos, como pressão e temperatura, utilizando sensores conectados a dispositivos IoT. Abordando a arquitetura da linguagem foi dividida em dois módulos principais: frontend, desenvolvido com html, css e javascript, e backend, implementado em Python utilizando o framework Django, finalizado a etapa de construção foi realizado uma série de testes básicos visando garantir que a linguagem atendesse às especificações e funcionasse conforme esperado. Os testes incluíram a interação com a ferramenta como usuário com a interface gráfica, a coleta e processamento de dados físicos por meio dos sensores, no qual também se encontra no tópico de exemplificação.

4.7. Skye Language

Neste tópico, abordaremos a ferramenta para a criação de algoritmos de baixo nível através de uma interface visual, permitindo que os usuários arrastem e soltem blocos de código em vez de escrever linhas de código manualmente.

4.7.1. Categoria dos Blocos

A Skye Language organiza seus blocos em várias categorias, facilitando a navegação e a seleção dos blocos necessários para a criação de programas.

- Bibliotecas: Blocos que permitem a importação e a utilização de diversas bibliotecas de código, expandindo as funcionalidades disponíveis na linguagem.
- Logica: Blocos que incluem operadores lógicos e estruturas condicionais, que permitem a criação de lógica de controle no programa.
- Laços de repetição: Blocos que permitem a execução repetida de um conjunto de instruções para facilitar a implementação de iterações.
- Variáveis: Blocos que permitem a criação, a modificação e a utilização de variáveis para armazenar e manipular dados no programa.
- Sensores: Blocos que permitem a leitura de dados de físicos, como temperatura, umidade e distância.
- Funções: Blocos que permitem a definição e a chamada de funções personalizadas, promovendo a reutilização de código e a organização modular do programa.
- Conexão: Blocos que facilitam a comunicação com outros dispositivos, permitindo a troca de dados, blocos essenciais para o funcionamento da rede de dispositivos IoT.

4.8. Exemplificação da linguagem

Neste tópico, exploramos a linguagem, destacando sua sintaxe e recursos através de um exemplo prático. A exemplificação visa facilitar a compreensão dos conceitos teóricos apresentados anteriormente, demonstrando como a linguagem pode ser aplicada em diferentes contextos.

4.8.1. Objetivo do experimento

O objetivo deste experimento é demonstrar a comunicação entre dois Arduino de forma serial, onde um Arduino atua como transmissor de dados de um sensor ultrassônico e o outro Arduino como receptor, exibindo os dados recebidos juntamente com informações de um sensor de temperatura e umidade (DHT11). Materiais necessários: 2 placas Arduino, Sensor ultrassônico HC-SR04, Sensor de temperatura e umidade DHT11, Buzzer, Cabos jumpers e uma Protoboard (opcional).

4.8.2. Interface inicial

Logo ao acessar a interface gráfica da linguagem, tem-se acesso a barra superior de navegação que permite navegar entre o editor de blocos, o console da linguagem, a conexão e o dashboard. No lado esquerdo da interface estão localizados os blocos divididos por categoria conforme mostrado no tópico 3.7.1. Por fim no canto inferior direito tem-se a lixeira para o descarte de blocos conforme mostra a Figura 3.

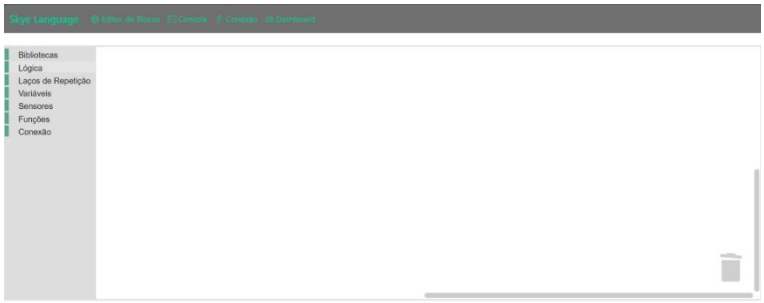

Figura 3. Interface inicial

4.8.3. Módulo receptor

Para construção do módulo receptor utilizou-se o bloco lógico de configuração e execução, importa-se a biblioteca DHT para o uso do sensor DHT11, após a importação realiza-se a configuração da conexão serial com o bloco correspondente passando como parâmetro a velocidade 115200 e por fim usa-se o bloco de leitura de serial para receber os dados enviados pelo modulo transmissor (Figura 4).

Figura 4. Construção do módulo receptor

Ao acessar o console pela barra de navegação pode-se copiar o código gerado, mas também há a possibilidade de o editar conforme demonstrado na Figura 5.

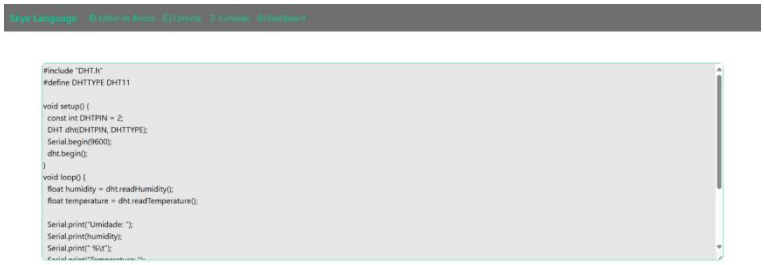

Figura 5. Algoritmo do modulo receptor

4.8.4. Módulo transmissor

Na construção do módulo de transmissão utilizou-se bloco lógico de configuração e execução, após isso é realizado a configuração da conexão serial com o bloco correspondente passando como parâmetro a velocidade 115200, ainda no bloco de configuração é criado a variável ‘duration’ que é usada como parâmetro para definir a duração de 1000 milissegundos para a coleta da distância, passando para o bloco de execução é criado um fluxo com o bloco condicional ‘if ...do’ que caso a distância coletada pelo sensor ultrassónico seja igual ou inferior a 5 centímetros é acionado um buzzer conectado no pino 6 para produzir um som, após a construção e acessar o console, tem-se o algoritmo resultante os parâmetros estabelecidos, conforme demostrado nas Figuras 6 e 7.

Figura 6. Construção do modulo transmissor

Figura 7. Algoritmo do modulo transmissor

Finalizado a etapa de construção é necessário descarregar os algoritmos gerados nos Arduinos, e aguardar que a conexão seja estabelecida e por fim acessar o dashboard pela barra de navegação e aguardar que a coleta de dados seja recebida e mostrada conforme Figura 8. Já na Figura 9 é demonstrado o esquemático do sistema montado.

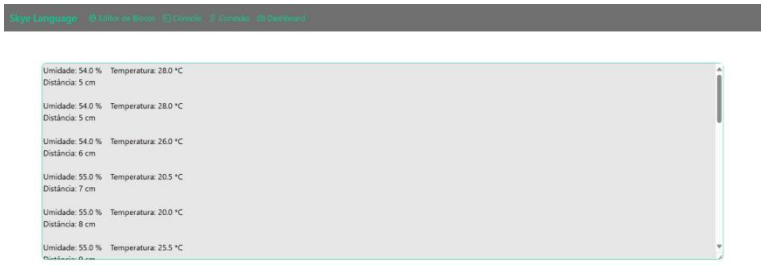

Figura 8. Tela de resultados de coleta

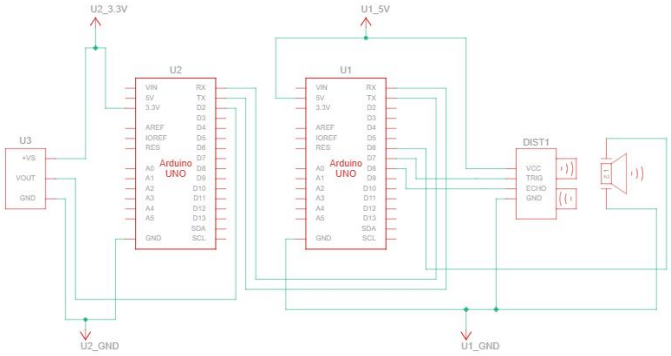

Figura 9. Esquemático do experimento

5. Considerações finais

No desenvolvimento desta pesquisa, os objetivos gerais e específicos foram essenciais para a obtenção do êxito parcial desta. A criação da Skye Language, uma linguagem para aquisição de dados programada por fluxo de atividades, contribuiu significativamente para o avanço da área e atende às necessidades de profissionais técnicos que buscam soluções eficientes e acessíveis para a coleta e análise de dados físicos em redes de sensores IoT. No entanto, o projeto também apresentou algumas limitações e desafios que podem ser superados em trabalhos futuros, como a necessidade de validar a linguagem em mais cenários e de aprimorar suas funcionalidades e recursos, incluindo suporte a mais tipos de sensores e integração com outras plataformas.

Portanto, este projeto alcançou bons resultados quando aplicado em cenários simples utilizando os parâmetros dos objetos criados para os sensores disponíveis. Contudo, reconhecemos que há margem para melhorias e expansões futuras. esperamos que este trabalho inspire novos projetos voltados à programação e coleta inteligente de dados. Com o contínuo desenvolvimento tecnológico, a importância de tornar a tecnologia mais acessível a um público mais amplo é fundamental, e acreditamos que este projeto possa contribuir para a democratização da tecnologia, facilitando o acesso e uso de ferramentas por pessoas de diferentes áreas e níveis de conhecimento.

Referências

Kitchenham, B. (2004) Procedures for performing systematic reviews. Keele, UK, *Keele University*, v. 33, s/n, p. 1-26.

Neto, J. J. (2000) “Solving complex problems efficiently with adaptive automata”. In: Conference on the Implementation and Application of Automata - CIAA 2000. Ontario, Canada: [s.n.].

Santos, S. L. (2006) Paradigmas de Programação. Campos, p. 35.

Vincent P, e Natis Y. e Iijima K. e Wong J. e Ray S. e Jain A. e Leow, A. (2020) Magic Quadrant for Enterprise Low-Code Application Platforms. September 2020 Gartner report, Gartner.