

## **Proposta de Projeto Integrador**

1. **Data:** 13/03/2025    **Grupo:** Grupo 5
2. **Nome Projeto:** Interpretador de Algoritmos
3. **Nome Usuário no GitHub:** LGAlopes
4. **Grupo de Alunos:**

RA	Nome	e-mail
0030482313003	Guilherme Vieira Silva	guilherme.silva548@fatec.sp.gov.br
0030482223019	Luís Gustavo de Assis Lopes	luis.lopes12@fatec.sp.gov.br
0030482213029	Guilherme Henrique V De Meira	guilherme.meira01@fatec.sp.gov.br

### 5. **Compreensão do Problema**

O Interpretador de Algoritmos é uma aplicação educacional desenvolvida com o objetivo de auxiliar estudantes no processo de aprendizado da lógica de programação e na resolução de problemas introdutórios por meio da utilização de pseudo-código.

Apesar de sua proposta pedagógica consolidada, o software encontra-se atualmente incompleto, apresentando limitações tanto em termos de funcionalidades quanto de interface. Recursos importantes para o pleno atendimento às atividades propostas em sala de aula, como o suporte a estruturas de dados mais complexas (por exemplo, matrizes), ainda não foram implementados.

Além disso, a interface do sistema necessita de uma reformulação para torná-la mais intuitiva, acessível e compatível com as práticas modernas de usabilidade e experiência do usuário.

O presente projeto visa a continuidade do desenvolvimento da aplicação, com foco na ampliação de suas funcionalidades e na modernização de sua interface, a fim de garantir maior eficácia no apoio ao ensino da programação.

### 6. **Proposta de Solução de Software e Viabilidade**

A proposta consiste no desenvolvimento de uma nova versão do Interpretador de Algoritmos, tomando como base conceitual e funcional a aplicação anteriormente desenvolvida. Essa nova implementação visa superar as limitações do sistema original, agregando funcionalidades adicionais que promovam um suporte mais eficaz tanto para alunos quanto para professores no contexto das aulas de lógica de programação.

Além da ampliação das capacidades da aplicação — como a inclusão de recursos que permitam o uso de estruturas mais complexas, a exemplo de matrizes —, o projeto também tem como objetivo a reformulação completa do design da interface. A nova interface será desenvolvida com foco em usabilidade, acessibilidade e experiência do usuário, tornando o sistema mais amigável, intuitivo e alinhado às demandas pedagógicas atuais.

## 7. Visão Geral dos Pré-Requisitos

O sistema deverá ser capaz de realizar a análise sintática e semântica de algoritmos escritos em pseudo-código, com suporte a estruturas fundamentais de controle de fluxo, tais como comandos condicionais (if, else) e laços de repetição (while, for).

Deverá, ainda, oferecer suporte completo à manipulação de estruturas de dados bidimensionais, como matrizes, garantindo a correta interpretação e execução dessas operações. A detecção de erros deverá ser precisa e informativa, apresentando mensagens claras e contextualizadas que auxiliem o usuário na correção do código.

A interface gráfica deverá ser totalmente reformulada, com foco em usabilidade e experiência do usuário, proporcionando uma visualização em tempo real da execução do pseudo-código, bem como feedback imediato das ações realizadas.

Além disso, o sistema deverá contemplar funcionalidades de exportação e importação de algoritmos, permitindo o armazenamento e recuperação de trabalhos. Também será prevista a integração com plataformas online para compartilhamento de códigos, visando facilitar a colaboração entre usuários.

Por fim, o sistema deverá ser multiplataforma, compatível com navegadores modernos e dispositivos móveis, assegurando acessibilidade e flexibilidade no uso da aplicação em diferentes ambientes.

## **8. Conceitos e Tecnologias Envolvidos**

O desenvolvimento do sistema abrangerá conceitos fundamentais da ciência da computação, incluindo análise sintática e semântica de pseudo-código, estruturas de controle de fluxo (como condicionais e laços de repetição), tratamento e exibição de erros, manipulação de matrizes e desenvolvimento de uma interface gráfica centrada na usabilidade.

Para a implementação, será utilizada a linguagem Java no backend, responsável pelo processamento dos algoritmos, validação semântica e sintática, além do gerenciamento das regras de negócio. No frontend, será empregado o Angular para o desenvolvimento de uma interface moderna, interativa e responsiva, garantindo uma experiência consistente em diferentes dispositivos e navegadores.

O sistema será construído como uma Progressive Web App (PWA), possibilitando que os usuários instalem a aplicação em seus dispositivos móveis e desktops, com funcionamento offline e desempenho aprimorado, aproximando a experiência à de aplicativos nativos e ampliando a acessibilidade.

Para persistência dos dados relacionados a algoritmos, usuários e configurações, será utilizado o banco de dados PostgreSQL, garantindo confiabilidade, escalabilidade e integridade das informações armazenadas.

Essa combinação tecnológica permitirá a construção de um sistema robusto, eficiente e multiplataforma, com suporte a funcionalidades como exportação e importação de algoritmos, colaboração online e visualização em tempo real da execução dos códigos.

## **9. Situação atual (estado-da-arte)**

Atualmente, além da versão anterior do **Interpretador de Algoritmos**, existem outras ferramentas utilizadas no ensino de lógica de programação por meio de pseudocódigo, como o **Portugol Studio** e o **Visualg**. Apesar de atenderem a propósitos educacionais básicos, essas soluções apresentam limitações relevantes, incluindo interfaces pouco intuitivas, funcionalidades restritas para exportação de resultados e baixa integração com plataformas online para compartilhamento e colaboração.

Com o objetivo de compreender melhor as necessidades dos usuários e orientar o desenvolvimento da nova aplicação, será realizada uma pesquisa com **alunos** por meio de formulários online. Essa pesquisa buscará identificar o nível de satisfação com as

ferramentas atualmente utilizadas, os principais desafios enfrentados e sugestões de melhorias.

Os dados coletados fornecerão subsídios importantes para o projeto, permitindo que a nova solução seja mais eficaz, acessível e alinhada às expectativas do público-alvo.

## 10. Estimativa de custo do projeto

Os custos do projeto estarão concentrados na **infraestrutura de publicação**, especificamente em **hospedagem** e **registro de domínio**.

A **hospedagem** terá um custo mensal estimado entre **R\$ 40,00 e R\$ 80,00**, considerando provedores que ofereçam boa performance e disponibilidade. O **registro de domínio** deverá custar entre **R\$ 60,00 e R\$ 90,00** por ano, dependendo da extensão escolhida.

As tecnologias utilizadas — incluindo **Java** no backend, **Angular** no frontend, **PostgreSQL** como banco de dados e **PWA** para suporte multiplataforma — são todas de uso gratuito, o que elimina a necessidade de investimentos em licenças ou ferramentas pagas.

Não há previsão de outros custos diretos no desenvolvimento da aplicação.

## 11. Glossário

- **Pseudocódigo:** Representação de algoritmos utilizando uma linguagem estruturada e de fácil compreensão, próxima ao português ou outra língua natural, sem a sintaxe rigorosa de linguagens de programação.
- **Análise Sintática:** Processo de verificar se o código segue a estrutura correta da linguagem de pseudocódigo, identificando erros de sintaxe.

- **Análise Semântica:** Avaliação do código para garantir que, além de sintaticamente correto, ele tenha sentido e comportamento esperado.
- **Estruturas de Controle:** Elementos da linguagem que permitem tomar decisões ou repetir blocos de código, como *condicionais* (if/else) e *laços de repetição* (for, while, do-while).
- **Interface Gráfica:** Parte do sistema responsável pela interação com o usuário, permitindo criar, editar e visualizar o código de forma intuitiva.
- **Tratamento de Erros:** Mecanismo para detectar e notificar o usuário sobre problemas no código, como erros de sintaxe ou semântica, com mensagens claras e precisas.
- **Exportação de Resultados:** Funcionalidade que permite salvar ou compartilhar o código e seus resultados em formatos como PDF ou texto.
- **Importação de Códigos:** Funcionalidade que permite carregar códigos de documentos existentes para edição ou execução no sistema.
- **Integração Online:** Conectar o sistema a plataformas online para facilitar o compartilhamento e acesso remoto aos projetos.
- **Compatibilidade Multiplataforma:** Capacidade do sistema de funcionar em diferentes dispositivos e navegadores, garantindo acessibilidade em diversas plataformas.