**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** |
| 0030482223019 | Luís Gustavo de Assis Lopes | luis.lopes12@fatec.sp.gov.br |
| 0030482213029 | Lucas Edvan da Silva Mendes | lucas.mendes8@fatec.sp.gov.br |
| 0030482213029 | Guilherme Henrique V De Meira | guilherme.meira01@fatec.sp.gov.br |
| 0030482313003 | Guilherme Vieira Silva | guilherme.silva548@fatec.com.br |

1. **Compreensão do Problema**

O projeto tem como objetivo reduzir as dificuldades enfrentadas pelos alunos na disciplina de Algoritmos e Lógica de Programação, além de proporcionar mais praticidade na realização das atividades e melhorar o aprendizado da matéria.

Desde 2018, a FATEC Sorocaba já conta com o Interpretador de Algoritmos, um software desenvolvido com a finalidade de ajudar os alunos na compreensão da lógica de programação e a resolução de problemas básicos utilizando pseudocódigo.

No entanto, o Interpretador apresenta limitações em seus recursos, não oferecendo suporte a estruturas de controle mais avançadas, matrizes e até mesmo mensagens de erros mais clara.

Diante disso, um novo programa pode ser desenvolvido visando ampliar essas funcionalidades e atuar como um suporte mais eficaz aos alunos e aos docentes.

1. **Proposta de Solução de Software e Viabilidade**

A proposta é desenvolver uma aplicação multiplataforma disponibilizada para web e dispositivos móveis (PWA??), facilitando a interação entre aluno e professor dentro e fora de aula.

A versão mobile permitirá que os alunos, mesmo sem acesso à internet, possam realizar suas atividades garantindo mais flexibilidade. Já a versão web poderá ser utilizada nos laboratórios, favorecendo aqueles que não dispõem de computadores pessoais.

Além das novas features, o design será completamente reformulado, contando com uma interface mais amigável e teclas de atalho mais acessíveis.

1. **Visão Geral dos Pré-Requisitos**

A aplicação contará com funcionalidades essenciais para garantir o acesso e a usabilidade da plataforma. Para isso, será implementado um mecanismo de login com credenciais, permitindo que os usuários acessem suas contas de forma segura. Além disso, haverá a possibilidade de cadastro de novos usuários (professores e alunos??), possibilitando a criação de perfis dentro da plataforma e assegurando sua utilização. Para facilitar a recuperação de acesso. O sistema também oferecerá uma funcionalidade de recuperação de cadastro e senha, permitindo que os usuários redefinam suas credenciais sempre que necessário, garantindo a continuidade do uso da tecnologia de maneira prática e acessível.

O sistema deverá realizar análise sintática e semântica de códigos, suportando estruturas de controle como condicionais e laços de repetição. Deve permitir a manipulação de matrizes, identificar e informar erros de forma precisa, e oferecer uma interface gráficaintuitiva com visualização em tempo real. Além disso, deve possibilitar a exportação e importação de códigos, integração com plataformas online para compartilhamento, e ser compatível com navegadores e dispositivos móveis.

1. **Conceitos e Tecnologias Envolvidos**

O desenvolvimento do software envolverá conceitos fundamentais como análise sintática e semântica, estruturas de controle (condicionais e laços de repetição), tratamento de erros, manipulação de matrizes, e criação de uma interface gráfica intuitiva.

Para isso, serão utilizadas as tecnologias *Java*, para a lógica de programação e processamento dos códigos, e *JavaScript*, *HTML* e *CSS* para o desenvolvimento da interface de usuário, garantindo uma experiência interativa e responsiva. Essas tecnologias permitirão a construção de um sistema eficiente, com suporte à exportação e importação de dados, além de compatibilidade multiplataforma.

1. **Situação atual (estado-da-arte)**

Atualmente, além do Interpretador de Algoritmos, existem ferramentas populares no ensino de lógica de programação, como *Portugol Studio* e *Visualg*. Ambas auxiliam alunos na criação e execução de algoritmos em pseudocódigo, mas possuem limitações que podem ser aprimoradas na nova solução proposta.

**Portugol Studio**

* **Descrição**: Ambiente de desenvolvimento para pseudocódigo, utilizado no ensino de lógica de programação.
* **Principais Características**:
  + Interface simples e didática.
  + Suporte a variáveis, estruturas condicionais e laços de repetição.
  + Geração de mensagens de erro para auxiliar na depuração.
  + Compatível com Windows, Linux e macOS.
* **Site Oficial**: <https://github.com/Univille/Portugol-Studio>
* **Valor**: Gratuito (Open Source).

**Visualg**

* **Descrição**: Ferramenta utilizada para criar e executar algoritmos em pseudocódigo, amplamente adotada em cursos de programação.
* **Principais Características**:
  + Ambiente simples para iniciantes.
  + Suporte a comandos básicos de entrada e saída.
  + Simulação de fluxogramas para facilitar a compreensão.
  + Disponível apenas para Windows.
* **Site Oficial**: <https://visualg3.com.br/>
* **Valor**: Gratuito.

A nova solução proposta buscará aprimorar as funcionalidades dessas ferramentas, adicionando suporte a estruturas avançadas, melhor tratamento de erros, interface mais intuitiva e integração multiplataforma (web e mobile).

Para melhor compreender as necessidades dos usuários, serão feitas duas pesquisas via forms: uma com o **professor Dimas e a professora Cristiane,** avaliando a aplicabilidade do interpretador em sala de aula, e outra para **alunos do curso de Análise e Desenvolvimento de Sistemas**, verificando o nível de satisfação e possíveis melhorias.

1. **Estimativa de custo do projeto**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Função** | **Valor Hora (R$)** | **Horas Estimadas** | **Custo Total (R$)** | **Referência** |
| Desenvolvimento Frontend | 50,00 | 100 | 5.000,00 | HTML, CSS, JavaScript |
| Desenvolvimento Backend | 55,00 | 120 | 6.600,00 | Java (Lógica do Interpretador) |
| UI/UX Design | 45,00 | 60 | 2.700,00 | Interface amigável e intuitiva |
| Testes e Validação | 40,00 | 50 | 2.000,00 | Testes funcionais e usabilidade |
| Hospedagem | - | - | 360,00 a 1.200,00 | R$ 30 a R$ 100/mês |
| Registro de Domínio | - | - | 40,00 a 100,00 | Anual |
| **Total Estimado** | - | - | **16.700,00 a 17.600** | - |

A tabela de custo do projeto foi elaborada com base em valores médios praticados no mercado para desenvolvimento de software. Os valores-hora foram estimados considerando dados de plataformas de freelancers, pesquisas salariais de desenvolvedores e experiências de mercado.

Os valores de desenvolvimento *frontend*, *backend* e *UI/UX design* foram estimados a partir de referências do site *Workana* (plataforma de freelancers) e pesquisas salariais da *Glassdoor* e *vagas.com.br*. O custo de hospedagem e domínio foi baseado em serviços populares como *HostGator***,** *UOL Host*e*GoDaddy*, que oferecem planos entre R$ 30 e R$ 100/mês para hospedagem e entre R$ 40 e R$ 100/ano para domínio.

**Referências**: (verificar normas ABNT para referências)

* [Workana - Preços de freelancers](https://www.workana.com)
* [Glassdoor - Salários de Desenvolvedores no Brasil](https://www.glassdoor.com.br)
* [Vagas.com.br - Média Salarial TI](https://www.vagas.com.br)
* [HostGator - Planos de Hospedagem](https://www.hostgator.com.br)
* [GoDaddy - Registro de Domínios](https://www.godaddy.com)

1. **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.