

Plataforma Interativa com Parsons Puzzles para o Ensino de Programação em C

Júlio César P. Cuencas
Sistemas de Informação – UFSM
Orientadora: Prof. Andrea S. Charão
12/05/2025

Introdução

Contexto e Problema

- Aprender programação em C é desafiador e exige uma carga cognitiva muito alta.
- Métodos tradicionais nem sempre engajam ou ajudam na estruturação de algoritmos.
- Muitos alunos se frustram ao tentar programar sem uma base sólida.

Proposta

- Utilizar a técnica **Parsons Puzzles**, que envolve reorganizar blocos de código.
- Focar no desenvolvimento da lógica antes da codificação completa.

Objetivo

Objetivo Geral

- Criar uma plataforma web interativa com Parsons Puzzles em C, voltada para iniciantes na programação.

Objetivos Específicos

- Adaptar o js-parsons à sintaxe da linguagem C.
- Desenvolver a interface com Vue.js.
- Testar a plataforma com estudantes reais e avaliar seu impacto.

Ensino de Programação

Dificuldades Comuns

- Sintaxe rígida da linguagem C dificulta o aprendizado inicial.
- Iniciantes têm dificuldades com abstrações como variáveis, estruturas de controle e lógica.
- Apesar disso, a linguagem C é fundamental para o entendimento de conceitos de programação.

Estratégias Didáticas

- Aplicação da técnica de Parsons Puzzles como abordagem baseada em problemas.
- Abordagens construtivistas, com foco na experimentação e na manipulação de blocos de código.
- Uso de ferramentas visuais que auxiliam na compreensão da lógica de programação.

Parsons Puzzles

O que são

- Exercícios onde o aluno organiza linhas de código embaralhadas para resolver um problema.

Benefícios

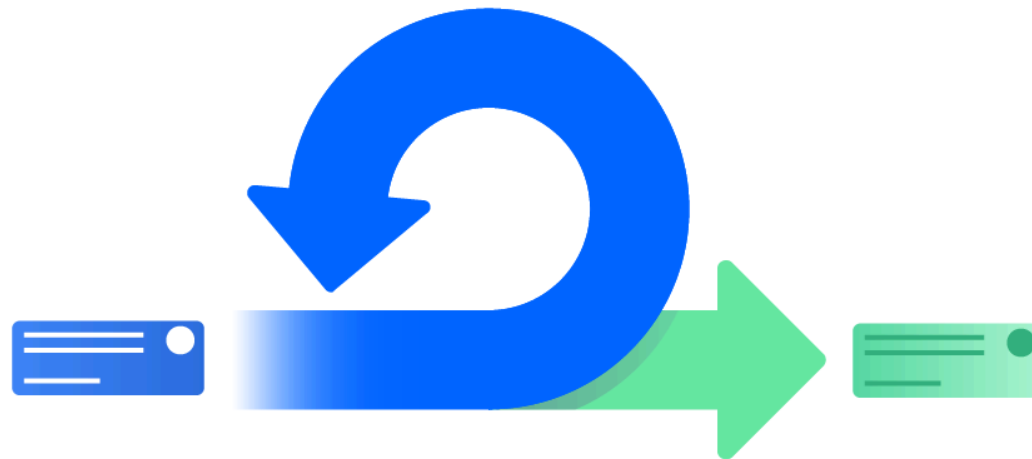
- Reforçam o raciocínio lógico sem exigir digitação ou memorização de sintaxe.
- Ajudam a desenvolver compreensão da estrutura de algoritmos.
- Reduzem a carga cognitiva para iniciantes.

Aplicações



- Usados em diversas plataformas educacionais para introdução à programação.
- Eficazes para avaliar entendimento da lógica sem distratores sintáticos.

Abordagem de Desenvolvimento

- Continuação e aprimoramento de projetos anteriores desenvolvidos por Beltrame (2023) e Izidoro (2024).
- Refatoração completa da interface utilizando **Vue.js**, com integração ao backend via **Google Apps Script**.
- Utilização de práticas ágeis, com sprints quinzenais e validação contínua com professores.



Etapas do Projeto

1. Estudo dos trabalhos anteriores e definição dos requisitos.
2. Adaptação da plataforma para suportar exercícios em linguagem .
3. Implementação da nova interface com .
4. Planejamento e aplicação de testes com alunos do CTISM.
5. Coleta e análise de dados (quantitativos e qualitativos).

Tecnologias e Ferramentas

Frontend

- Vue.js para construção modular da interface.
- Vite para build e publicação via **GitHub Pages**.
- Integração com biblioteca **js-parsons**.

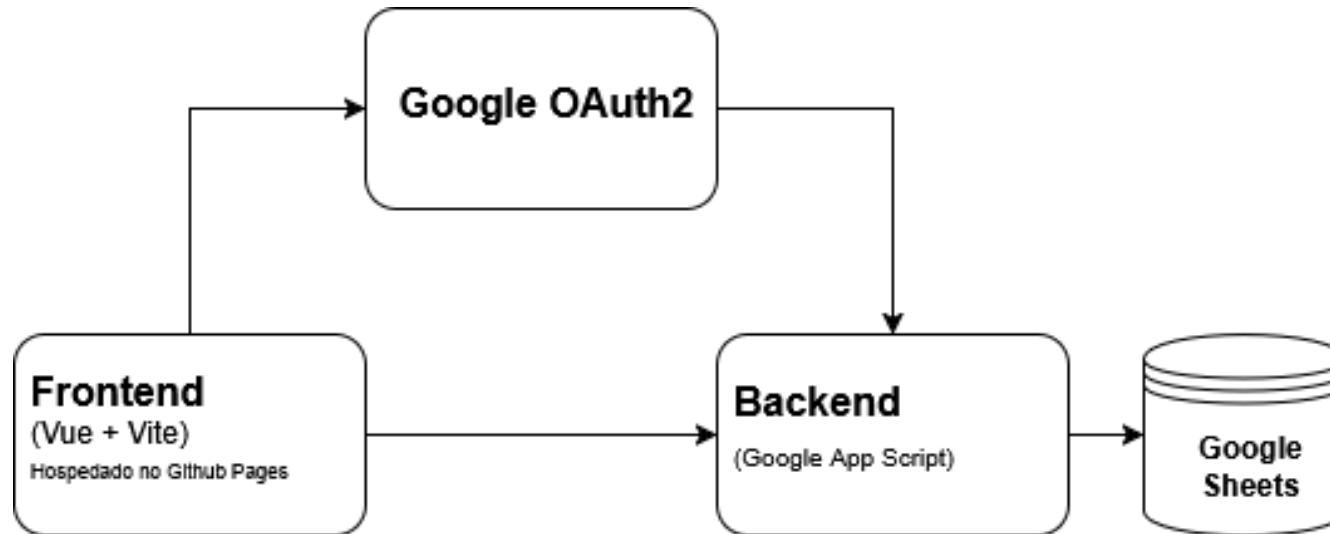
Backend

- **Google Apps Script** para API e persistência de dados no **Google Sheets**.
- Autenticação via conta Google (OAuth2).

Execução de Código C

- Integração com **OneCompiler** via iframe.
- Escolha baseada em custo zero, simplicidade e ausência de login obrigatório.

Estrutura da Plataforma



- O usuário acessa a interface desenvolvida em Vue.js, hospedada via GitHub Pages.
- A plataforma envia dados das respostas para o backend com Google Apps Script.
- Os dados são armazenados automaticamente em Google Sheets.

Estratégia de Avaliação

- Sessão de testes realizada com alunos do curso técnico em Informática (CTISM).
- Aplicação da plataforma em laboratório, com instrução e formulário de avaliação.
- Coleta de feedback:
 - **Quantitativo:** perguntas objetivas sobre uso e usabilidade.
 - **Qualitativo:** respostas abertas e observações em tempo real.

Roteiro de avaliação:

- Introdução - Apresentação do projeto (5 min).
- Teste - Usuários utilizam a plataforma (20 min).
- Feedback - Coleta de opiniões e sugestões (5 min).s

Temas Frequentes

Tabela com os temas mais frequentes mencionados pelos alunos:

Tema	Frequência
Adicionar mais exercícios	13
Indicação e explicação de erros	9
Sugestão de novas funcionalidades	8
Interface e usabilidade	5
Nada a mudar / Satisfeitos	11

As categorias com mais ocorrências ajudaram a identificar prioridades de melhoria.

Diretrizes de Melhoria

Conteúdo

- Incluir **mais exercícios** e temas variados (vetores, funções, etc).
- Adicionar níveis de **dificuldade**: iniciante, intermediário, avançado.

Feedbacks

- Oferecer mensagens de erro mais **claras e específicas**.
- Apontar **onde está o erro** e como corrigir.

Interface e Usabilidade

- Eliminação de **bugs** e melhorias na **navegação**.
- Melhorar a **interface** para torná-la mais intuitiva.

Outras Sugestões e Conclusão

Funcionalidades sugeridas

- Sistema de vidas, tempo limite e dicas graduais.
- Recursos inspirados em jogos para aumentar o engajamento.

Conclusão

- As sugestões reforçam que o projeto é bem aceito.
- Melhorias previstas:
 - Interface mais amigável;
 - Exercícios mais diversos;
 - Feedbacks mais úteis para o aprendizado.

As respostas servirão como base para a próxima versão da plataforma.

O que já foi feito







Etapas concluídas até o momento

- Estudo dos trabalhos anteriores e definição dos requisitos.
- Adaptação da plataforma para suportar exercícios em linguagem C.
- Implementação da nova interface com Vue.js.
- Planejamento e aplicação de testes com alunos do CTISM.
- Coleta e análise de dados (quantitativos e qualitativos).

Etapas futuras

- Integração do OneCompiler para execução de código C.
- Melhorias conforme feedback dos usuários.
- Implementação de um CRUD de exercícios direto na plataforma.
- Adição de lógica de adaptação de exercícios conforme o número de tentativas.
- Aprimoramento do feedback de erros, com mensagens mais detalhadas e úteis.
- Nova rodada de testes. Comparação dos resultados e validação da eficácia das mudanças.

Backlog Técnico e Pedagógico

Status	Tarefa	Prioridade
 Em andamento	Integração com OneCompiler para execução de código C	Alta
 Planejado	Melhorias na interface com base no feedback	Alta
 Planejado	Implementação de CRUD de exercícios	Alta
 Planejado	Adição de lógica de adaptação por tentativas	Média
 Planejado	Aprimoramento do feedback de erros	Média
 Planejado	Nova rodada de testes	Alta

Próximos Passos

Melhorias

- Aprimorar a **interface** com foco em usabilidade.
- Implementar **CRUD** de exercícios na própria plataforma.
- Adicionar **adaptação automática** de exercícios conforme desempenho.
- Melhorar o **feedback** de erros, com mensagens explicativas e orientativas.

Validação Futura

- Nova rodada de testes será realizada com:
 - Outro grupo de estudantes **ou**
 - As **mesmas turmas**, após as melhorias.
- O teste do **CRUD** de exercícios será aplicado com **professores**.

O objetivo é **comparar os resultados** e validar a eficácia das mudanças, consolidando a plataforma como uma ferramenta de apoio ao ensino de programação em C.

Obrigado! 🙏

Agradeço pela atenção e pela oportunidade de apresentar este trabalho.

Sigo à disposição para perguntas, sugestões e contribuições.