

**ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO**

**VICTOR PASSOS DE PINHO**

**Assistente Pedagógico para o iVProg**

**São Paulo  
2022**

**VICTOR PASSOS DE PINHO**

**Assistente Pedagógico para o iVProg**

**Trabalho de Conclusão de Curso  
apresentado à Escola Politécnica da  
Universidade de São Paulo**

**São Paulo  
2022**

**VICTOR PASSOS DE PINHO**

**Assistente Pedagógico para o iVProg**

**Trabalho de Conclusão de Curso  
apresentado à Escola Politécnica da  
Universidade de São Paulo**

**Área de Concentração: Engenharia  
Elétrica com Ênfase em  
Computação**

**Orientadora: Professora Doutora  
Anarosa Alves Franco Brandão**

**Coorientador: Mestrando Lucas  
Mendonça de Souza**

**São Paulo  
2022**

Pinho, Victor Passos de

Assistente Pedagógico para o iVProg / V. P. Pinho -- São Paulo, 2022.

124 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Engenharia Elétrica com Ênfase em Computação 2.Engenharia de Software 3.Educação I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

## RESUMO

Os Sistemas de Gerenciamento de Curso enfrentaram um crescimento maciço no seu uso devido à ênfase no aprendizado remoto durante a pandemia de COVID-19 em decorrência da suspensão do aprendizado na escola, em muitos países a educação online se tornou o ponto focal do ensino e da aprendizagem. Durante esse período, a satisfação do aluno com esse modelo de ensino está ligada à qualidade da informação disponível e à manutenção da autoeficácia do aluno, para o sucesso a longo prazo é essencial o suporte ao aprendizado e o desenvolvimento de métodos para manter o envolvimento dos alunos. A adição de um assistente pedagógico a um sistema educacional é uma forma de atender essas duas condições, já que um assistente pedagógico tem como função colaborar com a aprendizagem do aluno. O projeto proposto tem como objetivo desenvolver um assistente pedagógico para o iVProg, um sistema educacional de código livre destinado ao ensino e à aprendizagem de algoritmos e programação para iniciantes, esse sistema dispõe de um avaliador automático de exercícios que fornece um parecer instantâneo ao aluno, informando se sua solução resolve ou não o problema proposto. O assistente pedagógico desenvolvido interage com o resultado dessa avaliação automática a fim de construir um conteúdo referente a esse resultado, para que então exiba esse conteúdo ao aluno com o intuito de ajudá-lo a encontrar a solução para o problema proposto. Neste trabalho é apresentado o desenvolvimento e a implementação desse assistente pedagógico, e são exibidos os resultados obtidos em sua execução.

**Palavras Chave:** iVProg. Sistema de gerenciamento de curso. Assistente pedagógico.

## **ABSTRACT**

Learning Management Systems have faced a massive growth in their use due to the emphasis on remote learning during the COVID-19 pandemic as a result of the suspension of in-school learning, in many countries online education has become the focal point of teaching and learning. During this period, student satisfaction with this learning model is linked to the quality of the information available and the maintenance of student self-efficacy, for long-term success it is essential to support learning and to develop methods to maintain student engagement. The addition of a pedagogical assistant to an educational system is a way to meet these two conditions, since a pedagogical assistant has the function of collaborating with the student's learning. The proposed project aims to develop a pedagogical assistant for iVProg, an open source educational system aimed at teaching and learning algorithms and programming for beginners, this system has an automatic assessment tool that provides an instant feedback to the student, informing if his solution solves the proposed problem or not. The developed pedagogical assistant interacts with the result of this automatic evaluation in order to build content related to this result, so that this content is displayed to the student in order to help him find the solution to the proposed problem. This work presents the development and implementation of this pedagogical assistant, and shows the results obtained in its execution.

**Keywords:** iVProg. Learning management system. Pedagogical assistant.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>7</b>
1.1	MOTIVAÇÃO .....	7
1.2	ORGANIZAÇÃO DO TRABALHO .....	10
<b>2</b>	<b>ASPECTOS CONCEITUAIS</b> .....	<b>11</b>
2.1	AMBIENTE VIRTUAL DE APRENDIZAGEM E SISTEMA DE GERENCIAMENTO DE CURSO .....	11
2.2	ASSISTENTE PEDAGÓGICO .....	14
2.3	SISTEMAS DE PROGRAMAÇÃO VISUAL .....	15
<b>3</b>	<b>TECNOLOGIAS UTILIZADAS</b> .....	<b>27</b>
3.1	TECNOLOGIAS PARA APOIAR O DESENVOLVIMENTO .....	27
3.2	MÉTODO .....	29
<b>4</b>	<b>ESPECIFICAÇÃO DOS REQUISITOS DO SISTEMA</b> .....	<b>31</b>
4.1	REQUISITOS FUNCIONAIS .....	31
4.2	REQUISITOS NÃO FUNCIONAIS .....	33
<b>5</b>	<b>PROJETO</b> .....	<b>34</b>
5.1	TEMAS DE EXERCÍCIOS E CASOS DE TESTE .....	35
5.2	DESEMPENHO DO ALUNO .....	39
5.3	BANCO DE DADOS .....	41
5.4	ASSISTENTE PEDAGÓGICO .....	42
5.5	ENDPOINT .....	44
<b>6</b>	<b>IMPLEMENTAÇÃO</b> .....	<b>45</b>
6.1	INTERFACE DE AUTORIA .....	45
6.2	ASSISTENTE PEDAGÓGICO .....	47
6.3	BANCO DE DADOS .....	53
6.4	ENDPOINT .....	54
<b>7</b>	<b>TESTE E AVALIAÇÃO</b> .....	<b>60</b>
7.1	TESTAGEM DO RF1 .....	60
7.2	TESTAGEM DO RF2 .....	62

7.3	TESTAGEM DO RF3 .....	67
7.4	TESTAGEM DO RF4 .....	69
<b>8</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>72</b>
8.1	CONCLUSÕES DO PROJETO DE FORMATURA .....	72
8.2	CONTRIBUIÇÕES .....	73
8.3	PERSPECTIVA DE CONTINUIDADE .....	73
	<b>REFERÊNCIAS .....</b>	<b>75</b>
	<b>APÊNDICE A - EXERCÍCIOS DE IVPROG DESENVOLVIDOS .....</b>	<b>79</b>

## 1. INTRODUÇÃO

### 1.1 MOTIVAÇÃO

A suspensão do aprendizado na escola causada pela pandemia do COVID-19 iniciou uma mudança na maneira como professores e alunos, em todos os níveis, interagem uns com os outros e com os materiais de aprendizagem. A UNESCO (United Nations Educational, Scientific and Cultural Organization, em português: Organização das Nações Unidas para a Educação, a Ciência e a Cultura) estimou em 9 de outubro de 2020 que aproximadamente 1,6 bilhão de alunos no mundo, 94% do total de alunos matriculados, foram afetados pelo fechamento de escolas devido ao COVID-19. Em muitos países, a educação online se tornou o ponto focal do ensino e da aprendizagem [1] [2] [3]. A satisfação do aluno com a educação online durante esse período está intimamente ligada à qualidade da informação e à manutenção da autoeficácia do aluno [4]. O suporte ao aprendizado e o desenvolvimento de métodos para manter o envolvimento dos alunos são essenciais para o sucesso a longo prazo [5].

O uso de um assistente pedagógico é uma forma de fornecer suporte ao aprendizado do aluno. Assistentes pedagógicos têm como função colaborar com a aprendizagem do aluno, exibindo em sua interface conteúdos que tenham como objetivo cumprir essa função, podendo exibir textos de apoio, exemplos, vídeos, dentre outros conteúdos [6].

Nos últimos anos os cursos de programação tiveram um aumento significativo no número de alunos, no entanto não houve um aumento proporcional no número de professores e instrutores. Isso dificulta um suporte adequado aos alunos, o que impacta negativamente o aprendizado. Cerca de 71% dos alunos têm emoções negativas ao aprender a programar pela primeira vez, um suporte adequado no ensino de programação permite auxiliar os alunos a lidarem com as emoções negativas e permite guiar os alunos em seu aprendizado. O uso de sistemas de avaliação automatizados é uma forma de auxiliar os professores e instrutores a fornecerem um suporte adequado. No entanto, um *feedback* negativo constante do sistema pode desencorajar o aluno, causando emoções negativas e podendo

impactar negativamente a crença do aluno em sua capacidade de completar a tarefa. A adição de um agente pedagógico, como por exemplo, um assistente pedagógico, a um sistema de avaliação automatizado permite que seja dado um suporte adicional ao aluno, de forma que o *feedback* exiba também materiais de apoio e não somente um *feedback* negativo. Essa também é uma forma de auxiliar os professores e instrutores a fornecerem um suporte adequado [7].

Muitas pessoas possuem dificuldade no aprendizado de programação, o que pode ser observado através dos índices de reprovação e desistência em disciplinas de programação em cursos formais. Dentre os fatores responsáveis por essa dificuldade, se encontram habilidades e conhecimentos que não costumam fazer parte do contexto de pessoas que não sabem programar. Exemplos desses conhecimentos são o conceito de depuração, a compreensão do funcionamento de um computador, e a elaboração de conjuntos de teste. A dificuldade em desenvolver e compreender raciocínio lógico, assim como a dificuldade na abstração de problemas, são outros fatores responsáveis pelas dificuldades que as pessoas enfrentam no aprendizado de programação [8].

As linguagens de programação visual<sup>1</sup> possuem um potencial de melhorar a compreensão de programadores iniciantes sobre os conceitos de programação, muitas pessoas possuem uma maior facilidade de compreensão em questões lógicas quando elas apresentam componentes gráficos que podem ser utilizados para resolvê-las [9]. Esta característica pode proporcionar o estímulo do pensamento computacional<sup>2</sup> e facilitar o aprendizado de conceitos de programação [8].

---

<sup>1</sup> Linguagens de programação visual são linguagens que permitem a criação de programas através da manipulação de elementos gráficos.

<sup>2</sup> Pensamento computacional é um conjunto de métodos de resolução de problemas que envolvem expressar problemas e suas soluções de maneiras que um computador também poderia executar. Maiores informações em: [https://www.researchgate.net/publication/309916650\\_Computational\\_thinking\\_tools](https://www.researchgate.net/publication/309916650_Computational_thinking_tools).

Em função disso diversos Sistemas de Programação Visual são voltados para contextos educacionais, dentre eles o iVProg. O iVProg é um sistema educacional de código livre, destinado ao ensino e à aprendizagem de algoritmos e programação para iniciantes. Ele dispõe de um avaliador automático para exercícios e pode ser facilmente integrado ao ambiente Moodle através do módulo de extensão iTarefa. Assim que o estudante finaliza o algoritmo, já consegue receber um parecer instantâneo, informando se sua solução resolve ou não o problema proposto [8] [10].

Diante disso, este trabalho pretende desenvolver um assistente pedagógico para o iVProg, de forma que colabore com a experiência de aprendizado dos alunos no iVProg.

O projeto visa desenvolver um assistente pedagógico para o sistema educacional iVProg, que deve colaborar com a aprendizagem do aluno através da exibição de conteúdo relevante ao exercício executado. Esse assistente pedagógico exibirá ao aluno informações do seu desempenho relacionadas ao resultado da avaliação automática do exercício realizada pelo iVProg. O assistente pedagógico também deverá exibir sugestões pré determinadas ao aluno baseadas no seu desempenho relacionadas ao resultado da avaliação automática do exercício.

Uma aplicação atualmente em uso do iVProg é o seu emprego em atividades que consistem em desenvolver um algoritmo que solucione um problema proposto. Nessas atividades os alunos podem realizar as avaliações automáticas do exercício quantas vezes quiserem, ficando registrada apenas a última avaliação automática realizada. A avaliação automática de exercícios do iVProg permite que o aluno receba um parecer instantâneo, informando se sua solução resolve ou não o problema proposto [10].

Nesse contexto, a adição de um assistente pedagógico que tenha como função colaborar com o processo de resolução do aluno, pode proporcionar um processo menos conturbado e pode colaborar com a linha de raciocínio do aluno na testagem do algoritmo. Pois permite que sejam exibidas informações que auxiliem o aluno nesse processo [6]. A sua colaboração em uma atividade pode diminuir a probabilidade de desistência do aluno nessa atividade, podendo contribuir com o envolvimento do aluno no curso.

A adição do assistente pedagógico ao iVProg também permite exibir ao aluno informações sobre o seu desempenho nos exercícios. Dessa forma o aluno pode visualizar as situações nas quais ele costuma cometer erros. Com essa informação em mãos, o aluno pode tomar decisões que possam contribuir com a diminuição da sua taxa de erros nessas situações. Isso pode contribuir com a manutenção da autoeficácia desse aluno.

## 1.2 ORGANIZAÇÃO DO TRABALHO

Este documento detalha toda a construção deste projeto de criação de um assistente pedagógico para o iVProg. No capítulo 2 é feita a contextualização dos aspectos conceituais dos temas abordados neste projeto, sendo apresentados com a complexidade necessária para o entendimento dos capítulos posteriores. No capítulo 3 realiza-se a contextualização das tecnologias empregados no desenvolvimento do projeto, justificando os seus usos. Além disso, é apresentada a divisão metodológica do projeto. No capítulo 4 são definidos os Requisitos Funcionais e os Requisitos Não Funcionais do projeto. Neste capítulo são apresentados os diagramas dos casos de uso definidos pelos requisitos. No capítulo 5 são definidas as modificações implementadas em componentes já existentes e são definidos os componentes criados. Neste capítulo é apresentada a nova arquitetura dos sistemas modificados pelo projeto e o ciclo de vida dos novos componentes. No capítulo 6 descreve-se como foi feita a implementação do que foi definido para cada componente no capítulo anterior. No capítulo 7 são relatados os resultados obtidos na implementação do projeto, especificando quais os procedimentos de testes aplicados. No capítulo 8 são realizadas as considerações finais sobre o projeto, analisando os resultados alcançados, as contribuições do projeto e quais são as perspectivas futuras de continuidade do projeto.

## 2. ASPECTOS CONCEITUAIS

### 2.1 AMBIENTE VIRTUAL DE APRENDIZAGEM E SISTEMA DE GERENCIAMENTO DE CURSO

Um ambiente virtual de aprendizagem (AVA) em tecnologia educacional é uma plataforma baseada na web para os aspectos digitais de cursos de estudo, geralmente dentro de instituições de ensino. Eles apresentam recursos, atividades e interações dentro de uma estrutura de curso e prevêm diferentes etapas de avaliação. Costumam ter algum nível de integração com outros sistemas institucionais [11].

AVA geralmente permitem: o gerenciamento de conteúdo, via criação, armazenamento, acesso e uso de recursos de aprendizagem; o planejamento curricular, via planejamento de aulas, de avaliações e via personalização da experiência de aprendizagem; o acesso do aluno às suas informações, recursos e progresso; a comunicação e a colaboração, via *e-mails*, avisos, *chat*, *wikis*, *blogs*. Além disso, podem apresentar outras funcionalidades que enriquecem o ambiente [11]. AVA normalmente não são projetados para um curso ou assunto específico, mas são capazes de apoiar vários cursos em toda a extensão de um programa acadêmico, oferecendo uma interface consistente dentro da instituição [11].

Um Sistema de Gerenciamento de Curso (SGC) é uma aplicação para administração, documentação, acompanhamento, automação e entrega de cursos educacionais, programas de treinamento ou programas de aprendizado e desenvolvimento. Por meio de um SGC, os professores podem criar e integrar materiais do curso, articular metas de aprendizado, alinhar conteúdo e avaliações, acompanhar o progresso de estudo, criar testes personalizados para os alunos, se comunicar com os alunos. Um SGC pode permitir que os professores criem avaliações e tarefas automatizadas para os alunos, sendo que a maioria das plataformas permite uma variedade de formatos diferentes de avaliação [12]. Além disso, SGC apresentam interoperabilidade, acessibilidade, reutilizabilidade, durabilidade, capacidade de manutenção e adaptabilidade [13].

Os SGC foram concebidos para identificar lacunas de formação e aprendizagem, utilizando dados analíticos e relatórios. Os SGC são focados no aprendizado online, mas provêm suporte a uma variedade de usos, atuando como uma plataforma para conteúdo *online*, incluindo cursos síncronos e assíncronos. Existem SGC modernos que incluem algoritmos inteligentes para fazer recomendações automatizadas de cursos com base no perfil de um usuário, podendo extrair metadados de materiais de aprendizagem para tornar essas recomendações ainda mais precisas [14].

Observa-se na literatura uma certa ambiguidade no uso das terminologias AVA e SGC, sendo muitas vezes utilizadas como sinônimos diretos. Muitos AVA e SGC possuem os mesmos recursos, mas a ênfase e a forma como estão sendo usados os diferenciam [15]. Os AVA são caracterizados por princípios pedagógicos construtivistas, usados como um local para colaborar e estender discussões em vez de apenas hospedar objetos de aprendizagem. Os SGC têm uma abordagem mais *behaviorista* (comportamental), se concentram mais no comportamento funcional do instrutor e do aluno [16] [17].

### **2.1.1 Moodle**

Moodle (*Modular Object-Oriented Dynamic Learning Environment*, em português: Ambiente Modular de Aprendizagem Dinâmica Orientada a Objetos) é um *software* de apoio à aprendizagem executado em um ambiente virtual, sendo um *software* livre e de código aberto [18]. É a plataforma de aprendizagem mais utilizada do mundo, com mais de 213 milhões de usuários. Possui uma interface simples, suporte a mais de 120 idiomas, é bem documentado, escalável, robusto, permite a integração de módulos de extensão, personalizável, seguro, atualizado constantemente e fornece privacidade, dentre outras características [18].

A filosofia do Moodle é orientada pela pedagogia construcionista social, baseada em 4 conceitos principais relacionados: Construtivismo, Construcionismo, Construcionismo Social e Conectado e Separado. Do ponto de vista Construtivista, as pessoas constroem ativamente novos conhecimentos à medida que interagem com seus ambientes. O Construcionismo afirma que o aprendizado é

particularmente eficaz ao construir algo para que outros experimentem. O Construtivismo Social estende o Construtivismo aos ambientes sociais, nos quais os grupos constroem conhecimento uns para os outros, criando de forma colaborativa uma pequena cultura de elementos compartilhados com significados compartilhados. O conceito de Conectado e Separado se aprofunda nas motivações dos indivíduos em uma discussão, o comportamento Separado ocorre quando alguém tenta permanecer objetivo e factual, e tende a defender suas próprias idéias usando a lógica para encontrar buracos nas ideias de seu oponente. Já o comportamento Conectado é uma abordagem mais empática que aceita a subjetividade, tentando ouvir e fazer perguntas em um esforço para compreender o outro ponto de vista. Há também o comportamento construído, que é aquele no qual uma pessoa é sensível a ambas as abordagens e é capaz de escolher qualquer uma delas conforme for apropriado para a situação atual [19].

Como o Moodle é um *software* livre e de código aberto, foram desenvolvidos diversos módulos de extensão para os mais variados objetivos, o que permite criar e integrar diferentes ferramentas ao Moodle [18]. O iTarefa é um destes módulos de extensão, que será descrito na próxima subseção.

#### 2.1.1.1 iTarefa

iTarefa ou iAssign (*interactive Assignment*, em português: Tarefas interativas) é um sistema educacional gratuito fornecido pelo Laboratório de Informática na Educação (LInE). Ele consiste em um módulo de extensão do Moodle que permite enriquecê-lo com mais ferramentas de aprendizagem interativas, conhecidas como iLM (*interactive Learning Modules*, em português: Módulos de Aprendizagem interativos) [20] [21].

O objetivo do iTarefa é aumentar a interatividade em atividades relacionadas a assuntos específicos (geometria, funções, programação, dentre outros) de forma flexível [20] [21]. Para melhorar a interatividade, o iTarefa utiliza iLM, que consistem em ferramentas interativas que podem ser executadas em um navegador e que usam o protocolo HTTP para se comunicar com o Moodle [20] [21].

Inicialmente, todos os iLM disponíveis foram implementados em Java (como "miniaplicativo"), mas a versão atual do iTarefa permite integrar sistemas codificados em pilha HTML5 (com CSS e JavaScript). Isso significa que qualquer sistema Web, em princípio, poderia facilmente se tornar um iLM e ser integrado ao Moodle usando o pacote iTarefa. Por exemplo: se o iLM oferece funcionalidade de avaliação automática, o iTarefa é capaz de lidar com isso [20] [21]. Por motivos de segurança, apenas o administrador tem o privilégio de integrar um novo iLM ao iTarefa. Uma vez integrado, um iLM pode ser usado por qualquer pessoa com acesso ao iTarefa. Por exemplo, todos com privilégios de professores em diante têm permissão para usar as ferramentas de autoria iTarefa para criar novas atividades para os alunos [20] [21].

Um dos principais recursos providos pelo iTarefa é a ferramenta de autoria, que permite que qualquer professor prepare facilmente as atividades para os alunos. As atividades podem ser um exercício, e caso o iLM tenha recursos de avaliação automática, o iTarefa permite a autoria de exercícios associados à avaliação automática. Tal associação permite a confecção de relatórios de avaliação com estatísticas sobre a evolução dos alunos, e permite a integração das notas das atividades com as notas gerais do Moodle [20] [21].

## 2.2 ASSISTENTE PEDAGÓGICO

Assistente pedagógico é um tipo de agente pedagógico, que tem como função colaborar com a aprendizagem do aluno, exibindo em sua interface conteúdos que tenham como objetivo cumprir essa função, podendo exibir textos de apoio, exemplos, vídeos, dentre outros conteúdos [6]. Agente pedagógico é um conceito de ciência da computação e de inteligência artificial aplicado à educação, geralmente como parte de um Sistema Tutor Inteligente (STI). Ele consiste em uma interface humana simulada entre o aluno e o conteúdo em um ambiente educacional, e busca cumprir objetivos pedagógicos, objetivos de comunicação e objetivos de tarefa [6].

Um Sistema Tutor Inteligente é um sistema computacional que visa fornecer instrução ou um parecer imediato e personalizado aos alunos, geralmente sem exigir

a intervenção de um professor humano. Têm como objetivo possibilitar o aprendizado de maneira significativa e eficaz usando uma variedade de tecnologias de computação. Normalmente visam replicar os benefícios de uma tutoria individualizada e personalizada, em contextos em que diversos alunos teriam simultaneamente acesso à instrução e orientação de um único professor (por exemplo, palestras em sala de aula) ou nenhum professor (por exemplo, atividades *online*). Os Sistemas Tutores Inteligentes são frequentemente projetados com o intuito de fornecer acesso a educação de alta qualidade a todos os alunos [22] [23].

Para que um sistema seja considerado um STI ele deve cumprir os seguintes requisitos: o conteúdo deve ser codificado de forma que permita que o sistema acesse as informações, faça inferências ou resolva problemas; o sistema deve ser capaz de avaliar a aquisição do conhecimento pelo aluno; as estratégias dos tutoriais devem ser projetadas de forma que reduzam a discrepância entre o conhecimento do aluno e o conhecimento do especialista [24].

Os STI realizam processos de diagnóstico mais adaptados ao aluno e mais detalhados, permitem que o aluno faça perguntas ao tutor, possuem um conhecimento sobre o aluno que lhes permite adaptar e direcionar o ensino, têm o conhecimento restrito e articulado claramente, e não têm a sequência de ensino pré determinada pelo projetista instrucional [25].

Os STI se baseiam em uma arquitetura composta por quatro componentes: o Modelo do Aluno, módulo que armazena e modela as características individuais do aluno; o Modelo do Tutor, módulo que possui o conhecimento sobre as estratégias de ensino que devem ser selecionadas em função das características individuais do aluno; o Modelo do Domínio, módulo que detém o conhecimento sobre a matéria; e o Modelo da Interface, módulo que através do qual é realizada a interação entre o tutor e o aluno [23].

### 2.3 SISTEMAS DE PROGRAMAÇÃO VISUAL

A programação visual permite criar programas manipulando os elementos do programa graficamente, em vez de especificar esses elementos textualmente. Para isso deve-se utilizar uma linguagem de programação visual [26]. A programação

visual usa ícones, blocos, formas e diagramas para reduzir ou até mesmo eliminar completamente o potencial de erros sintáticos, ajudando no arranjo de primitivas de programação para criar programas bem formatados [27]. Além disso, pode fornecer alguns mecanismos para exibir o significado das primitivas da linguagem, que consistem nos elementos mais simples disponíveis na linguagem de programação. Esses mecanismos podem incluir funções de ajuda que fornecem funções de documentação integradas às linguagens de programação [27].

Programação visual permite o estudo do comportamento dos programas em situações específicas. Por exemplo, permite que os usuários coloquem artefatos criados em um determinado estado para explorar como o programa reagirá a esse estado [27].

Nas linguagens textuais cada palavra é um símbolo, semelhantemente nas linguagens visuais cada objeto ou relação multidimensional é um símbolo. Assim como ícones, diagramas, esboços e ações executadas por objetos gráficos são considerados expressões visuais, coleções de símbolos nas linguagens visuais são consideradas expressões visuais [28]. Uma linguagem de programação que inclui expressões visuais em sua sintaxe é considerada uma linguagem de programação visual. Um ambiente de programação em que são usadas expressões visuais no processo de programação é considerado um ambiente de programação visual [28].

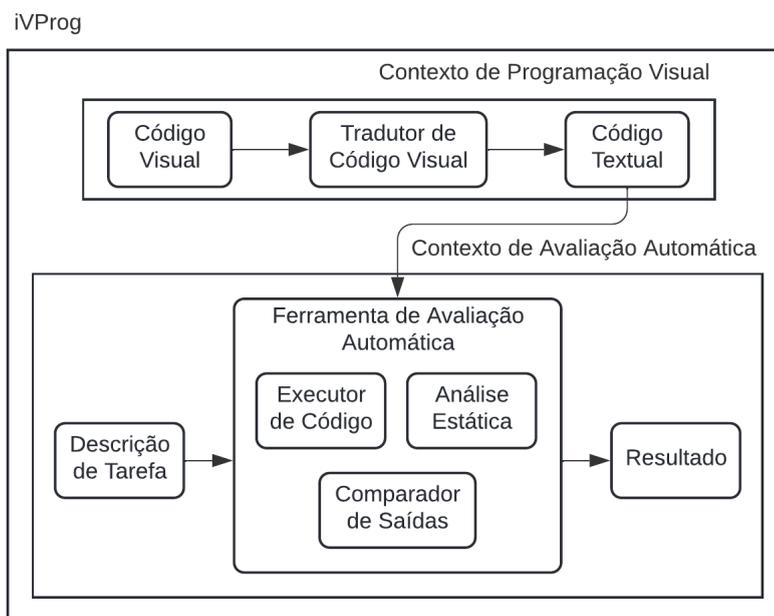
Muitas pessoas possuem uma maior facilidade de compreensão em questões lógicas quando elas apresentam componentes gráficos que podem ser utilizados para resolvê-las. Tal fato permite concluir que a programação visual possui um potencial de melhorar a compreensão de programadores iniciantes sobre os conceitos de programação [9]. Em função disso diversos Sistemas de Programação Visual são voltados para contextos educacionais, como por exemplo os sistemas Scratch, Alice e iVProg.

### 2.3.1 iVProg

iVProg (*Interactive Visual Programming on the Internet*, em português: Programação Visual Interativa na Internet) é um sistema educacional de código livre do LInE (Laboratório de Informática na Educação), destinado ao ensino e à aprendizagem de algoritmos e programação para iniciantes [8] [10]. A versão atual (Versão IV) foi implementada em HTML5, usando JavaScript. Isso possibilita sua execução em navegadores, o que permite sua incorporação a qualquer página da internet. A primeira versão (Versão I) é de 2009 e foi implementada em Java [8] [10]. O iVProg implementa o conceito de Programação Visual, que emprega ícones e simplifica a construção de códigos, reduzindo a necessidade do aluno conhecer detalhes de linguagens de programação, dando mais tempo para o aluno se dedicar ao aprendizado de algoritmos [8] [10]. Ele dispõe de um avaliador automático para exercícios que pode ser facilmente integrado ao ambiente Moodle através do pacote iTarefa. Assim que o estudante finaliza o algoritmo, já consegue receber um parecer instantâneo, informando se sua solução resolve ou não o problema proposto [8] [10]. O professor conta com uma área de autoria dentro do iVProg onde pode preparar as atividades, podendo reaproveitá-las em outros cursos. O sistema possui integração completa com o Moodle, podendo gerar um relatório com todas as atividades realizadas pelos alunos (sendo possível examinar cada solução enviada) e podendo permitir que as notas dos exercícios sejam integradas ao quadro de notas do Moodle [8] [10].

O iVProg versão IV possui sua estrutura composta por dois módulos, um relacionado ao contexto da programação visual e um relacionado ao contexto da avaliação automática. No módulo relacionado ao contexto da programação visual ocorre a tradução do código elaborado em linguagem de programação visual para linguagem de programação textual. O módulo relacionado ao contexto da avaliação automática recebe a tradução do código visual para textual e uma descrição da tarefa. Com essas informações ele realiza a execução do código, compara as saídas do programa com as saídas apresentadas na descrição da tarefa e fornece um retorno sobre a tarefa ao aluno de acordo com o resultado da execução e da comparação [8].

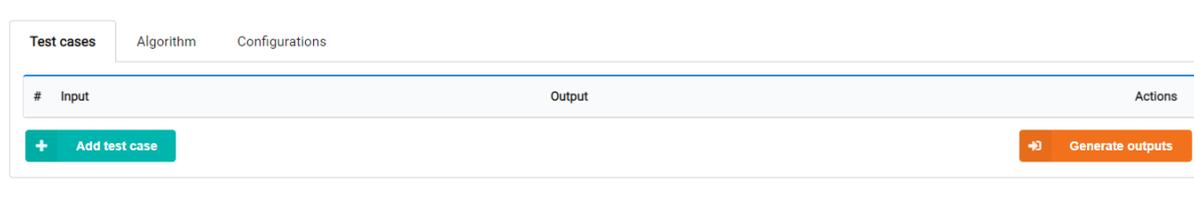
Figura 1 - Arquitetura da versão IV do iVProg



Fonte: Autoria própria, adaptação de figura fornecida por Lucas Mendonça de Souza

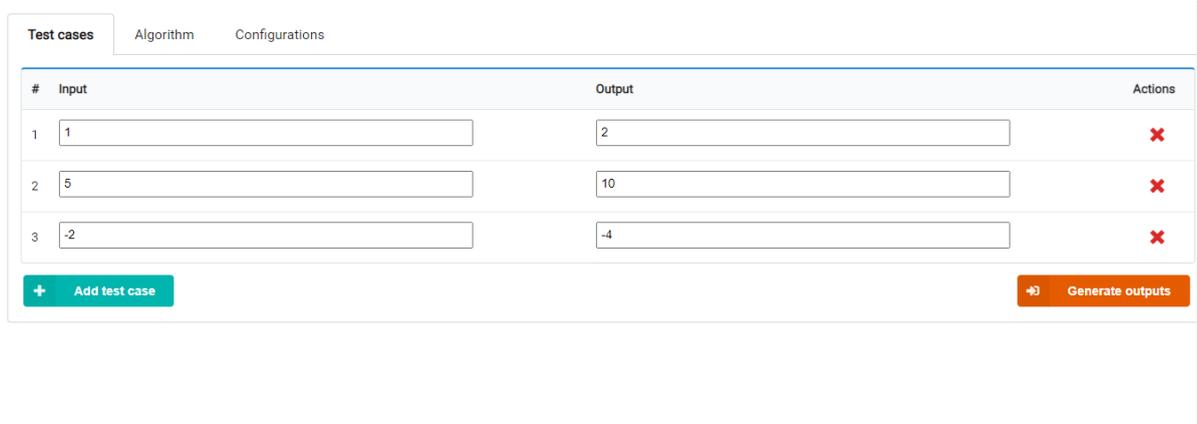
No iVProg versão IV existem duas entradas: o código visual implementado pelo usuário e a descrição de tarefa do avaliador. A descrição de tarefa do avaliador é gerada através da interface de autoria do iVProg, essa interface apresenta três abas. A primeira delas é referente aos casos de teste, contendo entradas e suas saídas esperadas, sendo possível adicionar quantos casos de teste forem necessários, removê-los e gerar as saídas esperadas para as entradas através de um algoritmo implementado na segunda aba [8].

Figura 2 - Primeira aba da interface de autoria



Fonte: Execução da versão IV do iVProg

Figura 3 - Primeira aba da interface de autoria preenchida

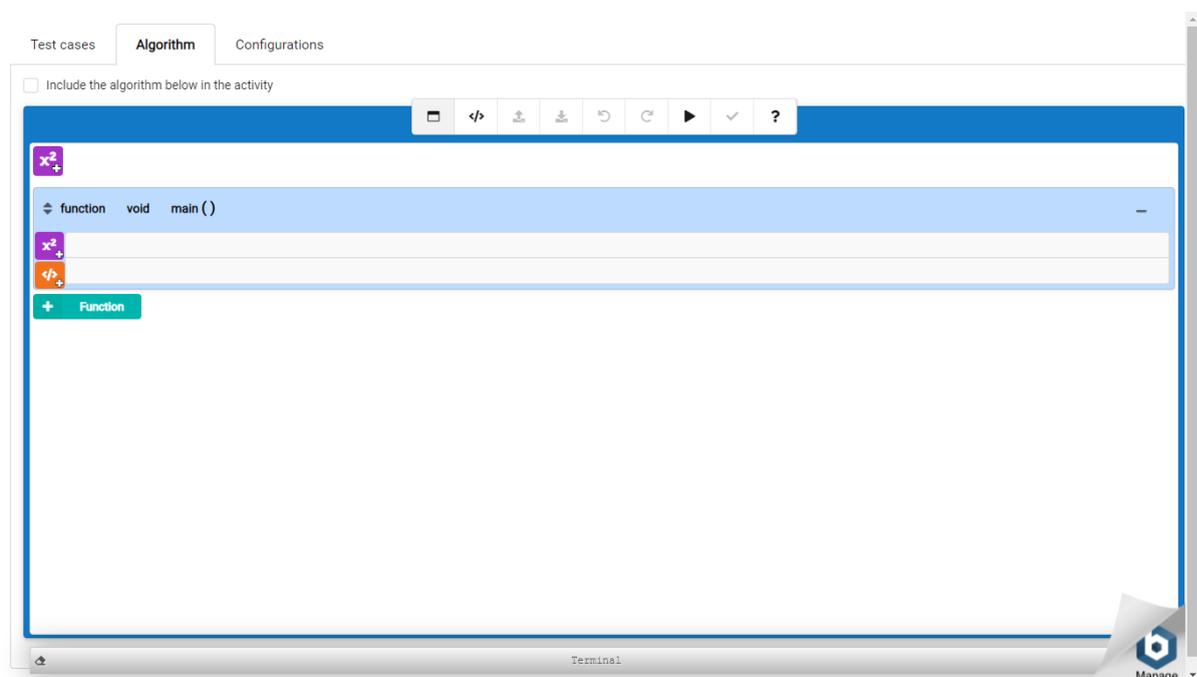


#	Input	Output	Actions
1	<input type="text" value="1"/>	<input type="text" value="2"/>	✘
2	<input type="text" value="5"/>	<input type="text" value="10"/>	✘
3	<input type="text" value="-2"/>	<input type="text" value="-4"/>	✘

Fonte: Execução da versão IV do iVProg

A segunda aba contém o iVProg, nesta aba é possível construir um algoritmo que resolva a tarefa proposta a fim de se gerar automaticamente as saídas esperadas para as entradas inseridas, é possível também construir um código com o qual a tarefa será inicializada, funcionando como uma espécie de ponto de partida para o aluno [8].

Figura 4 - Segunda aba da interface de autoria



Include the algorithm below in the activity

```
function void main ()
```

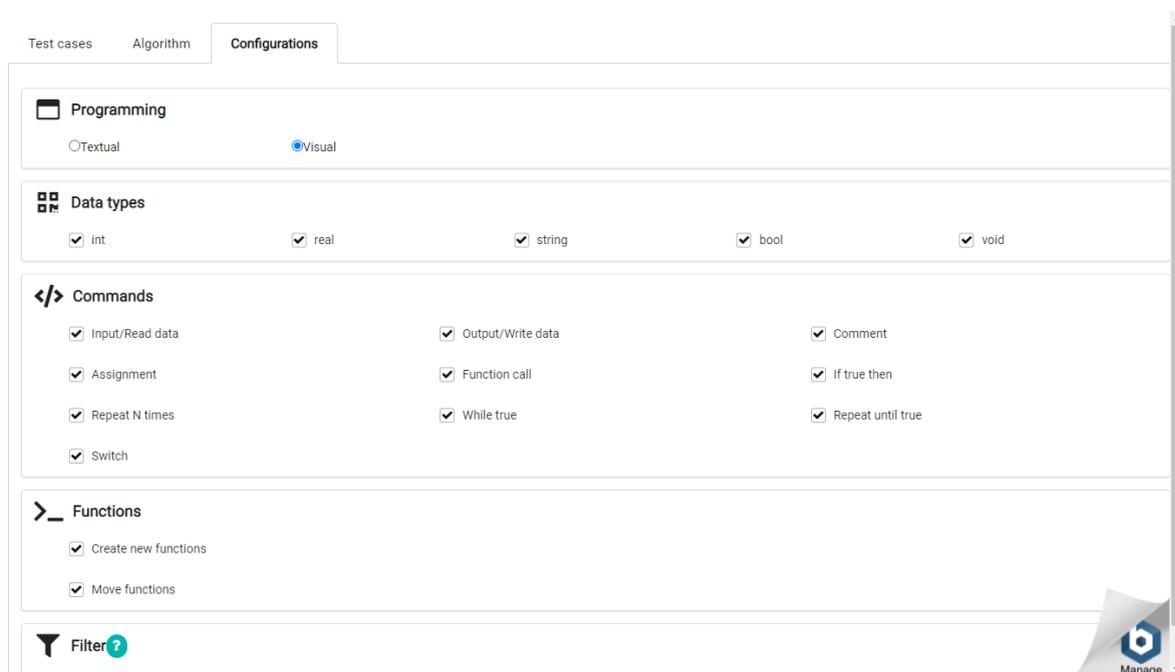
Terminal

Manage

Fonte: Execução da versão IV do iVProg

A terceira aba da interface é referente às configurações da tarefa, nela é possível selecionar quais configurações deseja-se aplicar a tarefa [8].

Figura 5 - Terceira aba da interface de autoria



Fonte: Execução da versão IV do iVProg

Ao finalizar a criação da descrição de tarefa do avaliador é gerado um arquivo .ivph que contém a descrição em formato JSON (*JavaScript Object Notation*, em português: Notação de Objeto JavaScript). A descrição é composta pelos casos de teste do exercício, que são conjuntos formados por entradas e por saídas esperadas para aquelas entradas, pelo código inicial do exercício e pelas configurações do exercício [8].

Figura 6 - Arquivo .ivph que contém a descrição de tarefa do avaliador

```

{
  "testcases" : [
    {
      "input": ["1"],
      "output": ["2"]
    },
    {
      "input": ["5"],
      "output": ["10"]
    },
    {
      "input": ["-2"],
      "output": ["-4"]
    }
  ],
  "settings_programming_type":
  [{"name": "programming_type", "value": "visual"}],
  "settings_data_types":
  [{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type", "value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "boolean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}],
  "settings_commands":
  [{"name": "commands_read", "value": "on"}, {"name": "commands_write", "value": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "commands_attribution", "value": "on"}, {"name": "commands_functioncall", "value": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_repeattimes", "value": "on"}, {"name": "commands_while", "value": "on"}, {"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch", "value": "on"}],
  "settings_functions":
  [{"name": "functions_creation", "value": "on"}, {"name": "functions_move", "value": "on"}],
  "settings_filter":
  []
}

```

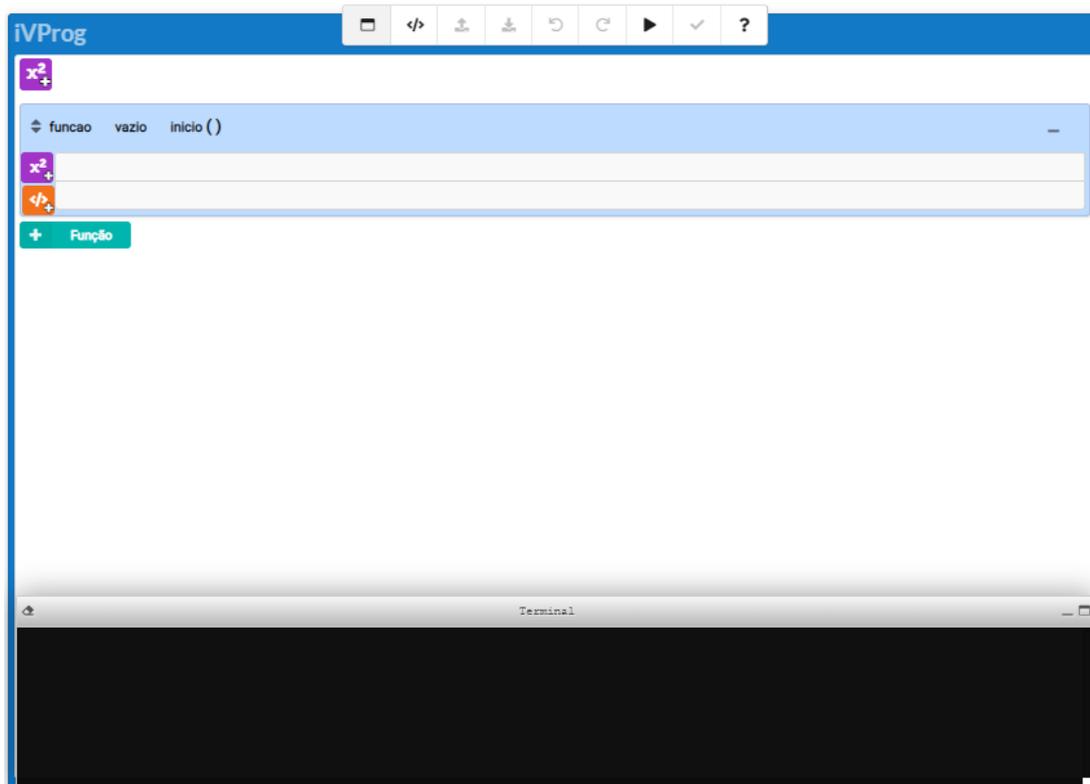
Fonte: Execução da versão IV do iVProg

A interface do iVProg versão IV possui duas janelas diferentes, a janela do código e a janela do terminal [8].

A janela do código é fixa na tela, possui duas abas (código visual e código textual), possui um botão para executar o código, um para executar a avaliação automática, um para desfazer a última ação, um para refazer a última ação desfeita, um para baixar o código desenvolvido, um para carregar um código no iVProg e um para pedir ajuda. Nessa janela é possível construir um código visual, observar a sua tradução para código textual e apertar os botões mencionados [8].

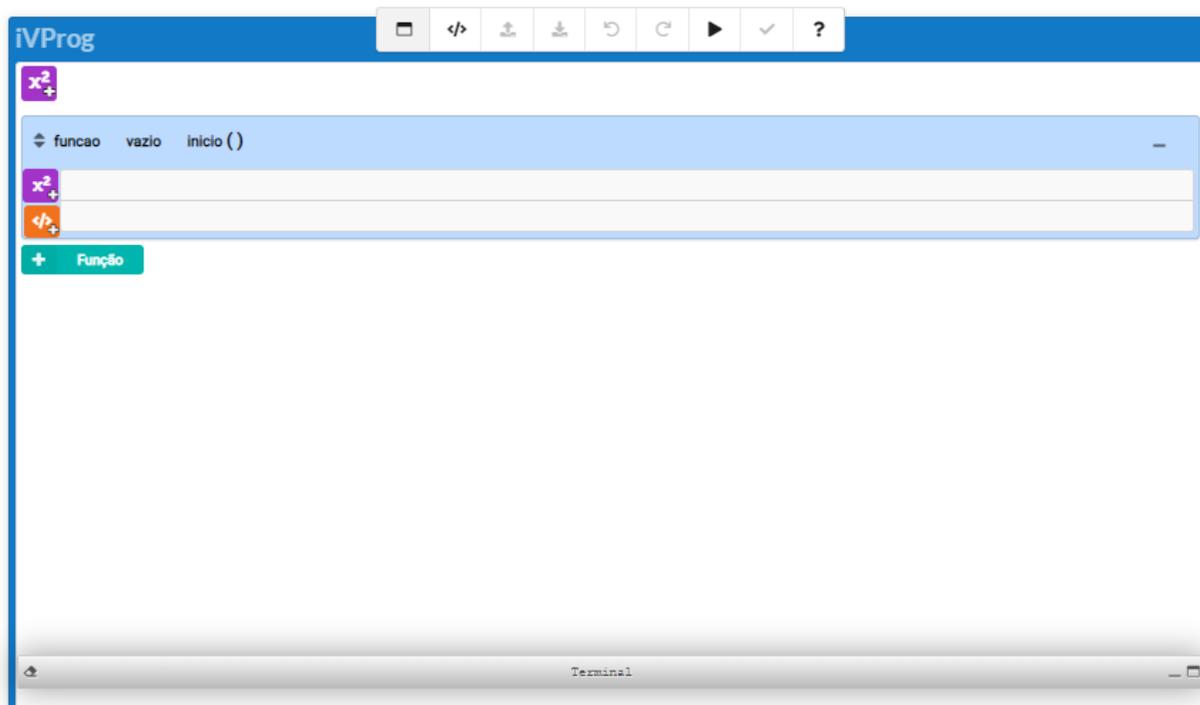
A janela do terminal pode ser movimentada pela tela, além disso ela pode ser minimizada e expandida. Nela são exibidos o resultado da execução do código e o resultado da avaliação automática [8].

Figura 7 - iVProg com o Terminal expandido



Fonte: Execução da versão IV do iVProg

Figura 8 - iVProg com o Terminal minimizado



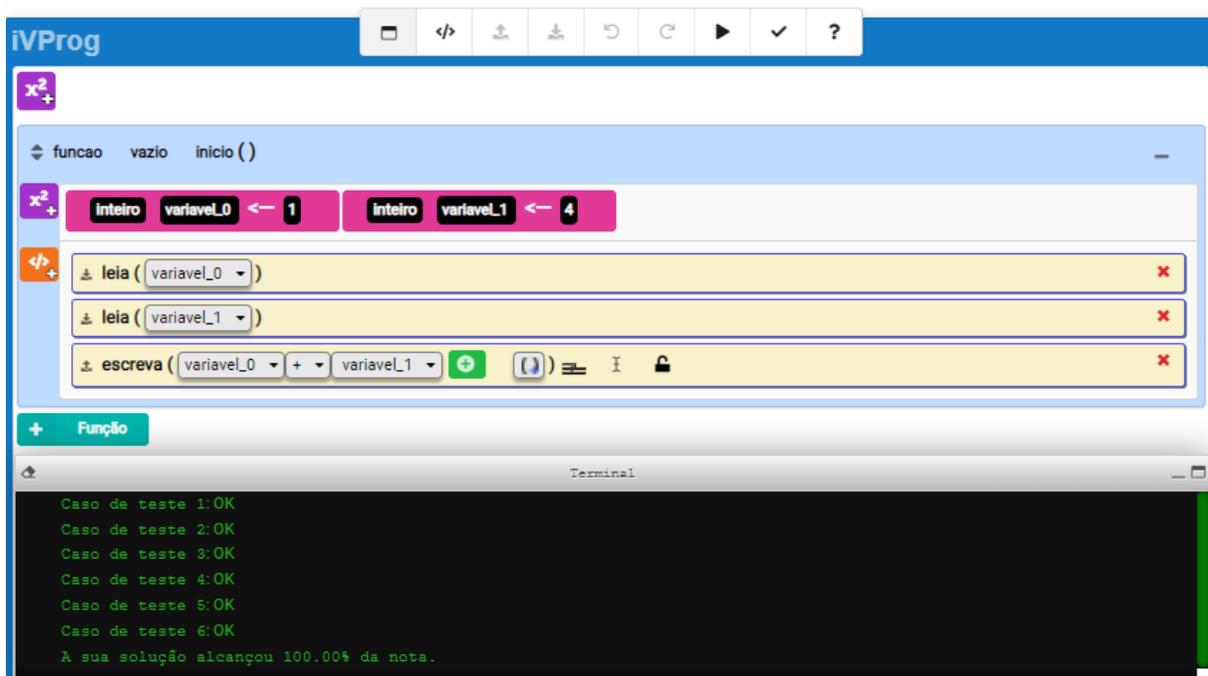
Fonte: Execução da versão IV do iVProg

A versão IV do iVProg possui uma saída composta por duas partes, uma parte consiste no resultado da execução do código, e a outra parte consiste no resultado da avaliação automática, sendo ambas exibidas na janela do terminal do iVProg [8].

O resultado da execução do código fornece informações sobre a execução do código, informando sobre a ocorrência de erros e exibindo o resultado da execução do código [8].

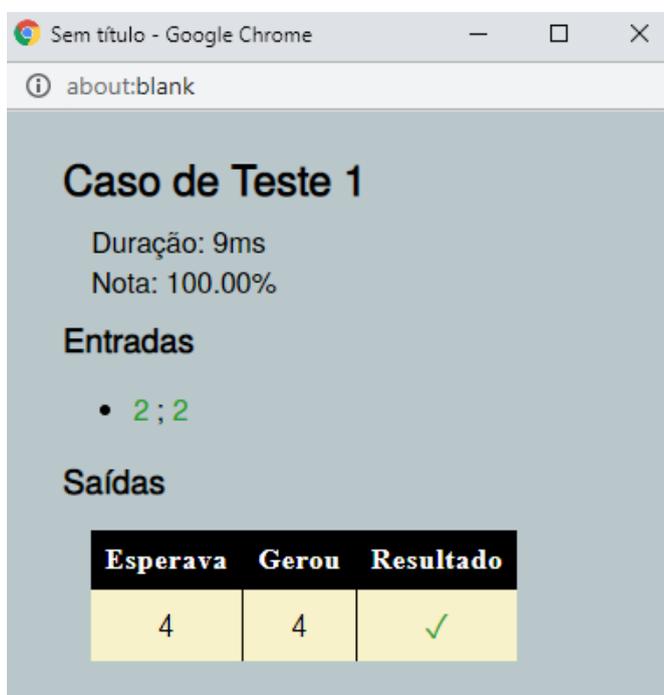
O resultado da avaliação automática exibe informações sobre a comparação dos casos de teste com o resultado da execução do código, informando o resultado de cada caso de teste e a nota total obtida. Os detalhes do resultado de cada caso de teste podem ser visualizados ao se clicar em cada caso de teste, o que resulta na abertura de uma nova janela do navegador contendo os detalhes do resultado do caso de teste clicado [8].

Figura 9 - Execução da avaliação automática de um exercício de soma de dois valores



Fonte: Exemplo 3 da página de exemplos do site do iVProg

Figura 10 - Nova janela do navegador exibida após clicar no texto “Caso de teste 1” presente no terminal da figura 9



Fonte: Exemplo 3 da página de exemplos do site do iVProg

### 2.3.2 Scratch

Scratch é um sistema de programação visual educacional gratuita baseado em blocos de alto nível que foi desenvolvido pelo Lifelong Kindergarten Group no Massachusetts Institute of Technology (MIT). Possui como público alvo crianças e adolescentes entre 8 e 16 anos, foi traduzido para mais de 70 idiomas, apresenta mais de 84 milhões de usuários registrados e mais de 95 milhões de projetos compartilhados. A versão atual, 3.0, pode ser acessada via editor *online*, e seus projetos podem ser exportados para HTML5, JavaScript, aplicativos Android e arquivos EXE, com o auxílio de ferramentas externas [29].

O Scratch foi desenvolvido para ser divertido, educativo e fácil de aprender. Possui ferramentas para criar histórias interativas, jogos, obras de arte, simulações e mais, usando programação baseada em blocos. Também possui seu próprio editor de pintura e editor de som embutido. Os usuários programam no Scratch arrastando blocos da paleta de blocos e anexando-os a outros blocos como um quebra-cabeça.

As estruturas de vários blocos são chamadas de scripts. O Scratch é amplamente utilizado em instituições de ensino como um meio de introduzir programação básica de computadores para crianças e adultos, de forma que possam adquirir uma compreensão dos fundamentos de programação com o Scratch, e em muitos casos migrar para outras linguagens de programação [29].

Durante o uso do Scratch as pessoas podem criar e colaborar com outras pessoas em projetos. Os projetos que os usuários criam podem ser compartilhados no site do Scratch e visualizados por outras pessoas. A flexibilidade do Scratch permite que o criador seja criativo e realmente faça o projeto desejado [29].

Ao projetar a linguagem, a principal prioridade dos criadores foi tornar a linguagem e o ambiente de desenvolvimento simples, intuitivo e que pudessem ser aprendidos facilmente por crianças que não tinham experiência anterior em programação. Há um forte contraste entre as poderosas funções multimídia e o estilo de programação *multi-thread* e o escopo bastante limitado da linguagem de programação Scratch [29].

A interface do Scratch é dividida em três seções principais: uma área de palco, uma área da paleta de blocos e uma área de codificação para colocar e organizar os blocos em *scripts* que podem ser executados pressionando a bandeira verde ou clicando no próprio código. Os usuários também podem criar seus próprios blocos de código e eles aparecerão em "Meus Blocos". A área do palco apresenta os resultados e todas as miniaturas de personagens listadas na área inferior. O palco usa as coordenadas x e y, com o centro do palco sendo a coordenada 0,0. Com um personagem selecionado na parte inferior da área de teste, blocos de comandos podem ser aplicados a ele arrastando-os da paleta de blocos para a área de codificação. A guia Fantasias permite que os usuários alterem a aparência do personagem para criar vários efeitos, incluindo animação. A guia Sons permite anexar sons e músicas ao personagem [30].

### 2.3.3 Alice

Alice é um sistema de programação educacional que implementa uma linguagem de programação baseada em objetos integrada a um ambiente de desenvolvimento. O sistema foi desenvolvido inicialmente na Universidade da Virgínia em 1994, e depois na Carnegie Mellon a partir de 1997, por um grupo de pesquisa liderado por Randy Pausch. Seu ambiente de programação visual facilita a criação de uma animação para contar uma história, jogar um jogo interativo ou um vídeo para compartilhar na web. Alice é uma ferramenta de ensino disponível gratuitamente projetada para ser a primeira exposição de um aluno à programação orientada a objetos. Ela permite que os alunos aprendam conceitos fundamentais de programação no contexto da criação de filmes animados e videogames simples. Em Alice, objetos tridimensionais povoam um mundo virtual e os alunos criam um programa para animar esses objetos [31].

Alice motiva o aprendizado por meio da exploração criativa. Em sua interface interativa, os alunos arrastam e soltam blocos gráficos para criar um programa, onde as instruções correspondem às instruções padrão em uma linguagem de programação orientada à produção, como Java, C++ e C#. Alice permite que os alunos vejam imediatamente como seus programas de animação são executados, permitindo que eles entendam facilmente a relação entre as instruções de programação e o comportamento dos objetos em sua animação. Ao manipular os objetos em seu mundo virtual, os alunos ganham experiência com todas as construções de programação normalmente ensinadas em um curso introdutório de programação [31].

Suporta vários recursos excelentes para auxiliar em uma transição completa para a linguagem de programação Java, incluindo a visualização do código Java gerado em uma janela lado a lado e até mesmo a exportação de seu mundo para o NetBeans para poder estender a funcionalidade codificando mundos Alice diretamente em Java [32].

Alice é usada por professores em diversos graus educacionais, desde de crianças a adultos, em assuntos que vão desde artes visuais e artes da linguagem até os fundamentos de programação e introdução para cursos de Java [33].

### 3. TECNOLOGIAS UTILIZADAS

#### 3.1 TECNOLOGIAS PARA APOIAR O DESENVOLVIMENTO

##### 3.1.1 JavaScript

Como mencionado na seção 2.3.1, a versão atual do iVProg (versão IV) está implementada em HTML5 com o uso de JavaScript. Com base nisso e na escolha de implementar o assistente pedagógico como um agente dentro do iVProg, foi definido que a linguagem utilizada no desenvolvimento do sistema do assistente pedagógico seria JavaScript [8] [10]. Desta forma se pode manter uma coesão no código do iVProg, além de evitar possíveis dificuldades e possíveis falhas na interação entre códigos de linguagens diferentes.

JavaScript é uma linguagem de programação interpretada estruturada, de script em alto nível que permite a construção de páginas interativas, o que o torna parte essencial dos aplicativos da web [34] [35]. A maioria dos *sites* utiliza o JavaScript. Em função disso todos os principais navegadores têm um mecanismo JavaScript dedicado para executá-lo. Portanto o seu uso permite projetar aplicações que executam em qualquer dispositivo com acesso a navegadores, permitindo uma maior facilidade de acesso a essa aplicação [36].

##### 3.1.2 PHP

Como mencionado na seção 2.3.1, a versão atual do iVProg (versão IV) possui integração completa com o ambiente Moodle através do módulo de extensão iTarefa, ou seja, a comunicação com o Moodle deve ser feita via iTarefa [8] [10]. A versão atual do iTarefa está implementada em PHP, portanto para que a comunicação com o Moodle seja feita da forma desejada é necessário utilizar também a linguagem PHP no desenvolvimento do projeto [20] [21].

PHP (*PHP: Hypertext Preprocessor*, em português: Pré-processador de Hipertexto, originalmente: *Personal Home Page*, em português: Página Inicial Pessoal) é uma linguagem interpretada, de código livre, orientada a objetos [37] [38].

Originalmente era utilizada apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, mas atualmente oferece funcionalidades em linha de comando e adquiriu características adicionais que possibilitam usos adicionais não relacionados à páginas web [39] [40].

### 3.1.3 Node.js

O projeto do iVProg utiliza o gerenciador de pacotes `npm` do `Node.js` como ferramenta de construção do programa a partir do código fonte. Em função disso, e da escolha de implementar o assistente pedagógico como um agente dentro do iVProg, é necessário utilizar o `Node.js` no desenvolvimento do sistema do assistente pedagógico [41].

`Node.js`<sup>3</sup> é um software de código aberto, multiplataforma, de *back-end* JavaScript baseado no interpretador V8 do Google que possibilita executar código JavaScript fora de um navegador. Permite que desenvolvedores usem JavaScript para desenvolver ferramentas de linha de comando e *scripts* do lado do servidor, executando-os do lado do servidor para produzir conteúdo dinâmico de página da web antes que a página seja enviada ao navegador do usuário.

Em decorrência disso, o `Node.js` representa um paradigma "JavaScript em todos os lugares", unificando o desenvolvimento de aplicativos da web em torno de uma única linguagem de programação, em vez de diferentes linguagens para scripts do lado do servidor e do lado do cliente.

É usado principalmente para a criação de servidores web e ferramentas de rede usando JavaScript e uma coleção de módulos que lidam com várias funcionalidades básicas. Os módulos são fornecidos para entrada e saída do sistema de arquivos, rede, dados binários, funções de criptografia, fluxos de dados, dentre outras funções.

---

<sup>3</sup> Disponível em: <https://nodejs.org/en/about/>.

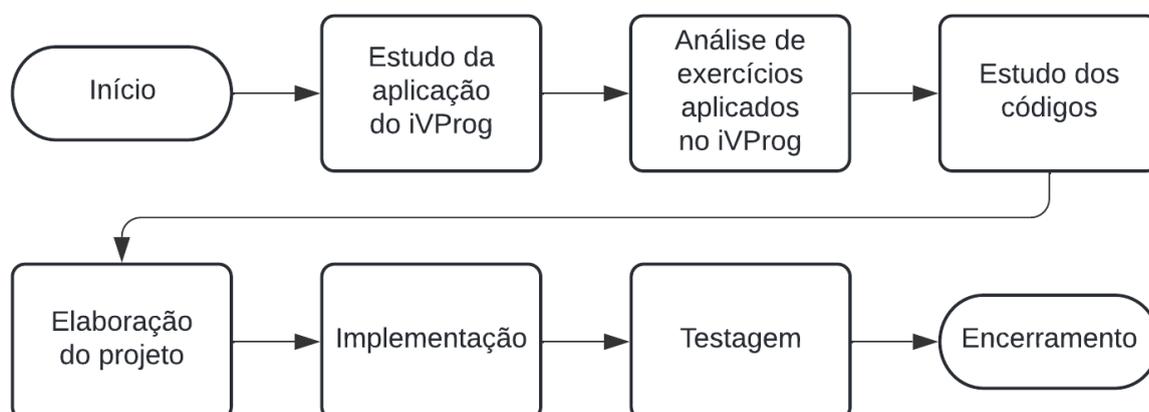
### 3.1.4 Banco de dados do Moodle

O assistente pedagógico necessita armazenar informações em um banco de dados. Em função do ambiente Moodle já apresentar um banco de dados próprio, foi decidido que as informações do assistente pedagógico devem ser armazenadas no banco de dados do Moodle. O acesso e o armazenamento nesse banco de dados é viabilizado pelo iTarefa [20] [21]. O Moodle apresenta seu próprio banco de dados, com esquema<sup>4</sup>, definição<sup>5</sup> e manipulação<sup>6</sup> próprias, o que resulta em funções próprias<sup>7</sup> para o gerenciamento e manipulação de dados.

### 3.2 MÉTODO

A divisão metodológica adotada neste projeto apresenta as seguintes etapas: estudo da aplicação do iVProg, análise de exercícios aplicados no iVProg, estudo dos códigos, elaboração do projeto, implementação e testagem.

Figura 11 - Divisão metodológica do projeto



Fonte: Autoria própria

<sup>4</sup> Disponível em: [https://docs.moodle.org/dev/Database\\_schema\\_introduction](https://docs.moodle.org/dev/Database_schema_introduction).

<sup>5</sup> Disponível em: [https://docs.moodle.org/dev/Data\\_definition\\_API](https://docs.moodle.org/dev/Data_definition_API).

<sup>6</sup> Mais informações disponíveis em: [https://docs.moodle.org/dev/Data\\_manipulation\\_API](https://docs.moodle.org/dev/Data_manipulation_API).

<sup>7</sup> Disponíveis em: [https://wimski.org/api/3.8/d3/d6e/classmoodle\\_database.html](https://wimski.org/api/3.8/d3/d6e/classmoodle_database.html).

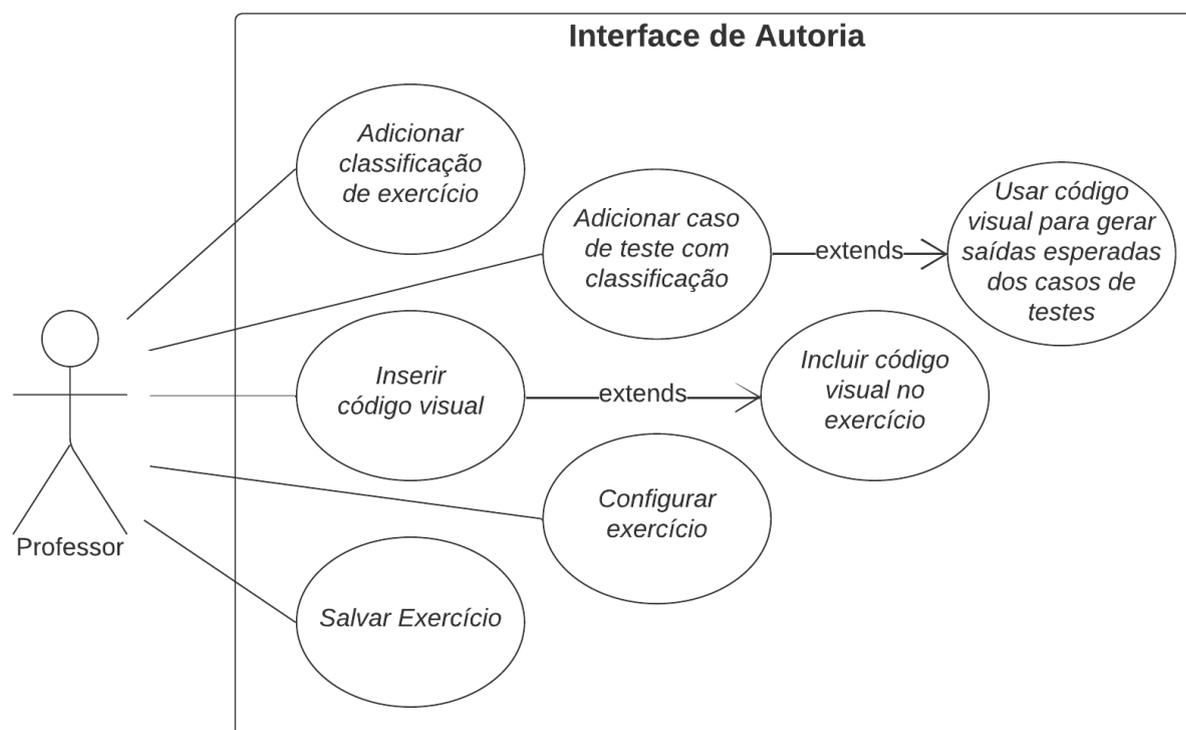
O estudo da aplicação do iVProg consiste em utilizar cada funcionalidade do iVProg, e tem como objetivo a familiarização com o iVProg, a fim de se entender como ele é utilizado, como se dá o seu funcionamento e de quais outras formas poderia ser utilizado. A análise de exercícios aplicados no iVProg ocorre para se entender quais são os tipos de exercícios aplicados, quais são os tipos de erros cometidos e se é possível categorizar os casos de testes utilizados. O estudo dos códigos é a etapa em que os códigos do iVProg, iTarefa e do banco de dados do Moodle são estudados, a fim de se entender como se dá a implementação do iVProg e como pode ser feita a comunicação com o iTarefa para viabilizar o acesso ao banco de dados do Moodle. Na elaboração do projeto são definidas as especificações de requisitos do sistema e como se planeja cumprir essas especificações. Na implementação é colocado em prática o que foi definido na elaboração do projeto, construindo os códigos necessários para cumprir os requisitos definidos. Na testagem são realizados testes elaborados para verificar se os requisitos de sistema definidos estão sendo cumpridos, ao longo da testagem realiza-se correções na implementação a fim de cumprir os requisitos definidos.

## 4. ESPECIFICAÇÃO DE REQUISITOS DO SISTEMA

### 4.1 REQUISITOS FUNCIONAIS

O primeiro Requisito Funcional (RF1) é que o sistema deve permitir a classificação de exercícios na interface de autoria, de forma que essa classificação possa ser utilizada pelo assistente pedagógico para calcular o desempenho dos alunos no exercício realizado através do resultado da avaliação automática do exercício.

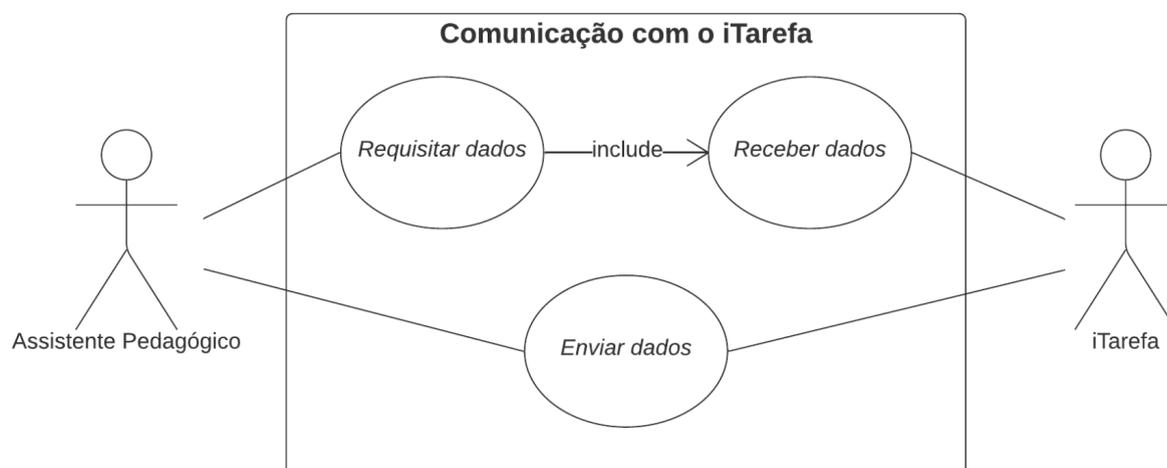
Figura 12 - Diagrama de casos de uso da interface de autoria



Fonte: Autoria própria

O segundo Requisito Funcional (RF2) é que o sistema deve ser capaz de se comunicar com o iTarefa para viabilizar o acesso e armazenamento de dados no banco de dados do Moodle.

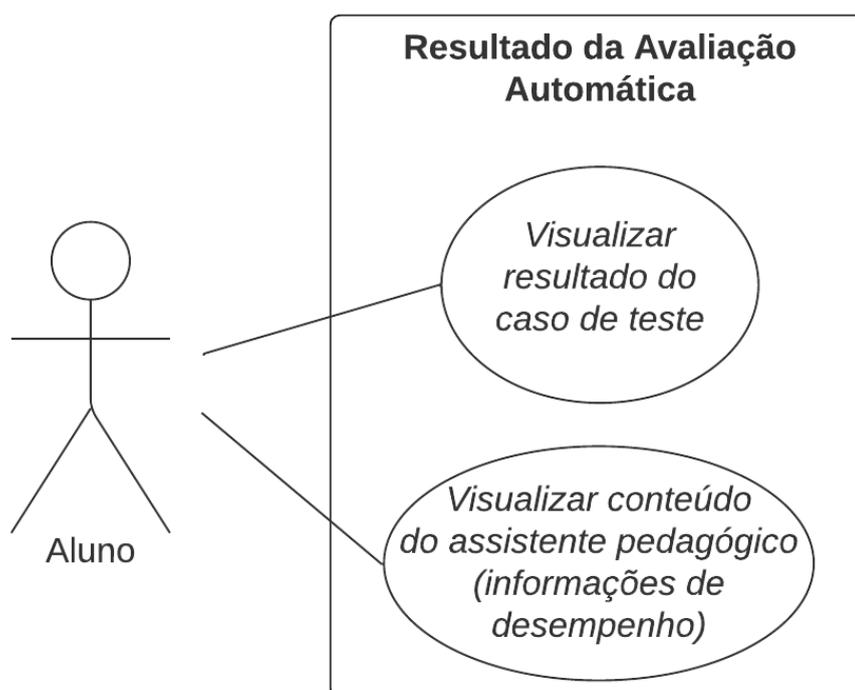
Figura 13 - Diagrama de casos de uso da comunicação com o iTarefa



Fonte: Autoria própria

O terceiro Requisito Funcional (RF3) é que o sistema deve permitir que os alunos possam visualizar informações do seu desempenho relacionadas ao resultado da avaliação automática do exercício.

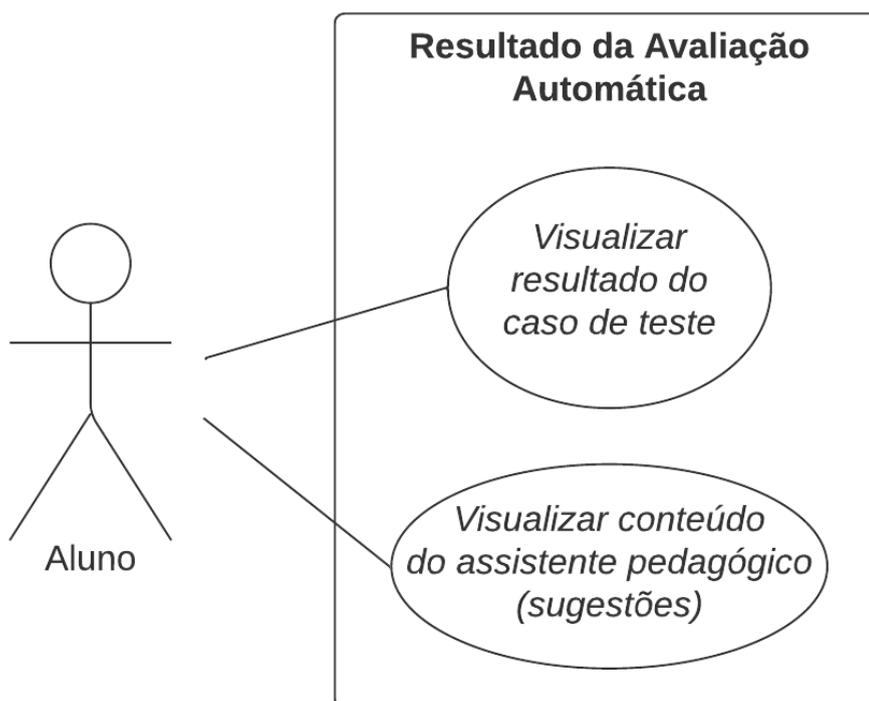
Figura 14 - Diagrama de casos de uso do resultado da avaliação automática



Fonte: Autoria própria

O quarto Requisito Funcional (RF4) é que o sistema deve permitir que os alunos possam visualizar sugestões pré determinadas baseadas no seu desempenho relacionadas ao resultado da avaliação automática do exercício.

Figura 15 - Diagrama de casos de uso do resultado da avaliação automática



Fonte: Autoria própria

#### 4.2 REQUISITOS NÃO FUNCIONAIS

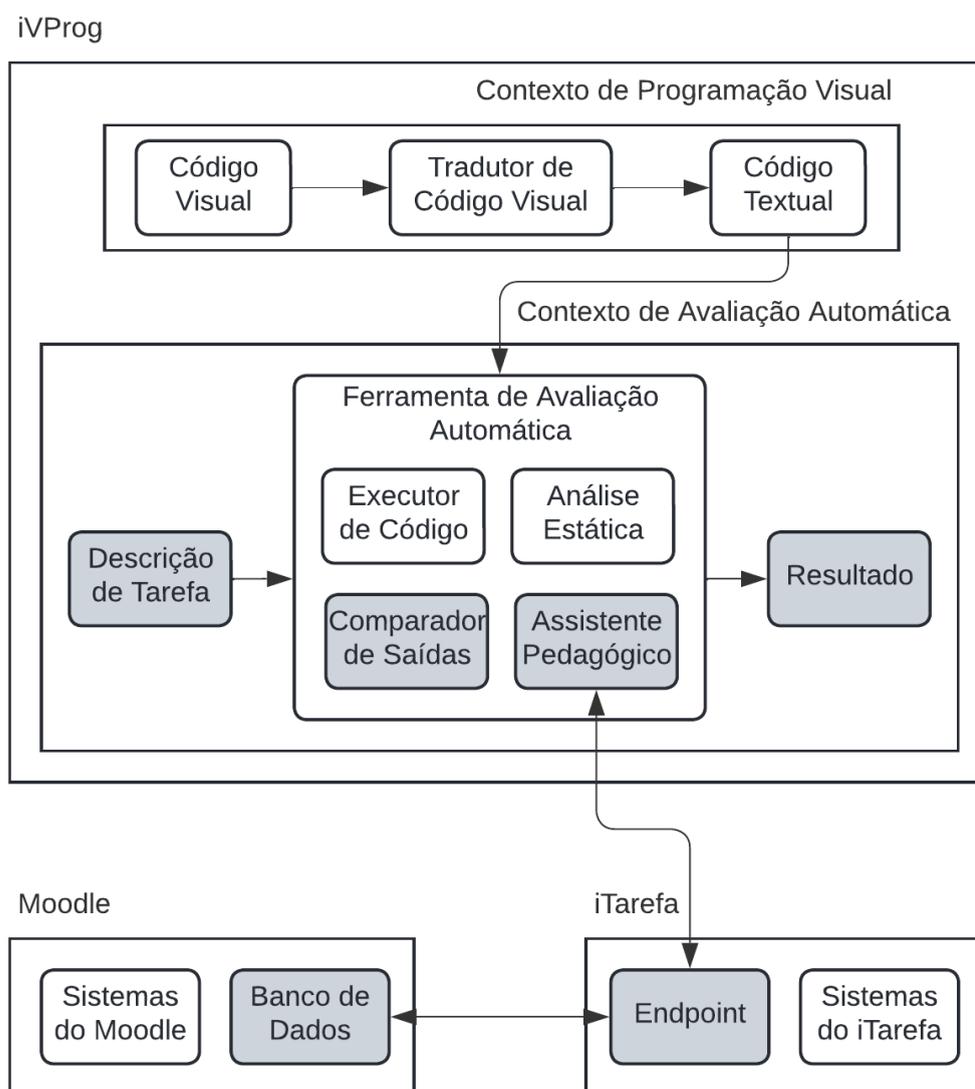
O sistema também apresenta Requisitos Não Funcionais, sendo o primeiro deles que o sistema deve ser uma extensão conservativa, ou seja, o sistema deve adicionar novas funcionalidades ao iVProg mantendo as funcionalidades já existentes.

O outro Requisito Não Funcional presente é que o sistema deve seguir o mesmo estilo e modelo de interface que é usado na versão atual do iVProg (versão IV), a fim de se evitar problemas de usabilidade por parte tanto dos usuários atuais do IVProg quanto de novos usuários.

## 5. PROJETO

Para cumprir o Requisito Funcional RF1 necessita-se alterar a interface de autoria do iVProg, a fim de se inserir as classificações de exercício e de caso de teste. O Requisito Funcional RF2 necessita da criação de um canal de comunicação (endpoint) entre o assistente pedagógico e o iTarefa e da adição de uma tabela referente ao assistente pedagógico no banco de dados do Moodle. Os Requisitos Funcionais RF3 e RF4 estão contidos nas funcionalidades do assistente pedagógico.

Figura 16 - Arquitetura dos sistemas modificados por este projeto (os componentes criados ou modificados por este projeto estão destacados em cinza)



## 5.1 TEMAS DE EXERCÍCIOS E CASOS DE TESTE

A partir de uma análise dos exercícios de iVProg presentes no curso “2021: Introdução à Programação (Diurno)” do SAW, foram classificados os exercícios por tema e os casos de teste por tipo. Nessa análise foram levantados quais os possíveis temas principais de cada exercício, quais os erros cometidos pelos alunos e quais os possíveis tipos de casos de teste aplicáveis ao exercício.

Na tabela abaixo encontra-se o resultado da análise: a primeira coluna contém o nome do exercício no curso; na segunda coluna se encontram os temas principais do exercício; na terceira coluna estão os erros cometidos pelos alunos com a quantidade de alunos que cometeu o erro e os tipos de casos de teste afetados pelo erro; na quarta coluna se encontra uma classificação dos casos de teste aplicáveis no exercício.

Tabela 1 - Resultado da análise dos exercícios de iVProg presentes no curso “2021: Introdução à Programação (Diurno)” do SAW

<b>Exercício</b>	<b>Temas</b>	<b>Erros dos alunos</b>	<b>Casos Teste</b>
EF-3.1.a Introdução - Olá mundo!	Leitura e Escrita	1 - Errou um caractere (Todos) 1 - Número errado de entradas (Todos)	Caso Geral
EF-3.1.b Introdução - Ler um inteiro e imprimí-lo	Leitura e Escrita	-----	Caso Geral Caso Nulo Caso Negativo
EF-3.2.a Consumo médio	Matemática	1 - Mais de uma função e todas vazias (Todos)	Caso Geral Caso Nulo Tipo de Entrada
EF-3.2.b Troca de valores	Manipulação de Entrada	1 - Número errado de entradas (Todos)	Caso Geral Caso Nulo Caso Negativo
EF-3.2.c Antecessor e sucessor	Lógica Matemática	1 - Número errado de entradas e mais de uma função (Todos)	Caso Geral Caso Nulo Caso Negativo
EF-3.2.d Imposto em Marcianópolis	Matemática	1 - Número errado de entradas (Todos)	Caso Geral Caso Nulo Tipo de Entrada

EF-3.2.e Folha de pagamento	Lógica Matemática	1 - Errou uma das saídas/Não prestou atenção no enunciado (Todos) 1 - Número errado de saídas e Lógica/Falta de atenção (Todos)	Caso Geral Caso Nulo Tipo de Entrada
EF-3.2.f Total de dias	Matemática	-----	Caso Geral Caso Nulo Caso Negativo
EF-3.2.g Área do círculo	Matemática	1 - Tipo de variável errado (Tipo de Entrada) 1 - Uso errôneo da função "raiz_quadrada" do iVProg (Todos)	Caso Geral Caso Nulo Tipo de Entrada
Divisão da conta do restaurante	Lógica Matemática	1 - Lógica (Todos) 1 - Número errado de entradas (Todos)	Caso Geral Caso Limite Caso Nulo Tipo de Entrada
Inverter número inteiro de 3 dígitos	Manipulação de Entrada Lógica Matemática	3 - Número de entradas errado (Todos) 3 - Uso errôneo da função "inverter_valor" do iVProg (Todos)	Caso Limite Caso Nulo Caso Negativo Igualdade
Item 4 - Média ponderada	Matemática	1 - Ordem de leitura errada (Todos) 1 - Desistiu (Todos)	Caso Geral Caso Nulo Caso Negativo
EF-5.1.a Identificar se 2 números são iguais ou diferente	Condição	-----	Caso Geral Caso Limite Caso Nulo Caso Negativo
EF-5.1.b Imprimir o maior número dentre dois inteiros	Condição	-----	Caso Limite Caso Nulo Caso Negativo Igualdade
EF-5.1.c Verificar se um número é divisível por outro	Condição	1 - Condição (Caso Limite e Igualdade)	Caso Limite Caso Nulo Igualdade Tipo de Entrada
EF-5.1.d Soma ou subtração	Condição	2 - Sinal (Caso Limite)	Caso Limite Caso Nulo Caso Negativo Igualdade
EF-5.1.e Par ou ímpar	Condição	-----	Caso Geral Caso Negativo

Soma de inteiros representando ângulos internos de um triângulo	Condição	-----	Caso Geral Caso Limite Caso Nulo Caso Negativo Igualdade
6.1 Laço - Somatório dos primeiros naturais até um limite	Laço	2 - Lógica/Não entendeu o enunciado (Caso Geral) 1 - Inicialização de variável e Atribuição da variável pós laço (Caso Limite)	Caso Geral Caso Limite
6.2 Laço - Soma dos números ímpares (até digitar 0)	Laço Condição	2 - Atribuição de variável pré laço (Caso Limite e Caso Nulo) 1 - Condição de parada (Todos) 1 - Lógica/Não entendeu o enunciado (Todos)	Caso Geral Caso Limite Caso Nulo Caso Negativo
8.1 Vetor - Soma das posições x e y	Vetor Laço	1 - Lógica/Não entendeu como funciona atribuição dentro de laços (Todos)	Caso Geral Caso Limite Caso Negativo Igualdade

Como pode-se observar, a tabela não possui muitos erros dos alunos. Isso ocorreu porque o iVProg armazena apenas a última execução do exercício do aluno, o que limitou o acesso aos erros anteriores à última execução.

Na análise apresentada na tabela foram definidos sete possíveis temas de exercício:

T1 - Leitura e Escrita: exercícios relacionados apenas a leitura e escrita de dados.

T2 - Manipulação de Entrada: exercícios que tem como objetivo a manipulação da entrada recebida.

T3 - Lógica: exercícios nos quais a lógica desempenha um papel importante.

T4 - Matemática: exercícios onde se aplicam problemas matemáticos.

T5 - Condição: exercícios que necessitam do uso de pelo menos uma condição para que possam ser resolvidos.

T6 - Laço: exercícios em que a solução desejada necessita do uso de pelo menos um laço para que possam ser resolvidos.

T7 - Vetor: exercícios em que a solução desejada necessita do uso de vetores para que possam ser resolvidos.

Foram definidos também seis possíveis tipos de caso de teste:

C1 - Caso Geral: testam o código executado com valores gerais, que não tem comportamento diferenciado, ou seja, os que não se encaixam nos outros tipos de caso de teste.

C2 - Caso Limite: testam o código executado com valores que se encontram no limite da mudança de comportamento do exercício.

C3 - Caso Nulo: testam o código com valores nulos.

C4 - Caso Negativo: testam o código com valores negativos.

C5 - Igualdade: testam o código com valores de entrada iguais.

C6 - Tipo de Entrada: testam se o código executado tem o tipo certo de variável como entrada.

Os temas e os casos de teste definidos são aplicáveis a problemas simples de programação, que é o caso de uso atual do iVProg. Para problemas mais complexos é necessário um estudo mais aprofundado para se definir temas e tipos de caso de teste apropriados.

Tabela 2 - Temas de exercício e tipos de casos de teste

<b>Temas de Exercício</b>	<b>Tipos de Caso de Teste</b>
T1 - Leitura e Escrita	C1 - Caso Geral
T2 - Manipulação de Entrada	C2 - Caso Limite
T3 - Lógica	C3 - Caso Nulo
T4 - Matemática	C4 - Caso Negativo
T5 - Condição	C5 - Igualdade
T6 - Laço	C6 - Tipo de Entrada
T7 - Vetor	

## 5.2 DESEMPENHO DO ALUNO

O desempenho do aluno deve conter informações sobre o desempenho completo do aluno e informações sobre o desempenho recente do aluno, pois permite que o aluno tenha um panorama geral e um panorama recente de seu desempenho. A partir disso e do resultado da análise realizada na seção anterior, foi definido o formato do desempenho do aluno. Ele é composto pelo desempenho relativo aos temas de exercício, pelo desempenho relativo aos tipos de caso de teste, pelo registro dos últimos X erros relativos aos temas de exercício cometidos e pelo registro dos últimos X erros relativos aos tipos de caso de teste cometidos. O valor definido para X inicialmente é 50, esse valor pode ser alterado futuramente caso a aplicação do assistente pedagógico em um ambiente real gere essa necessidade.

O desempenho do aluno é armazenado em um objeto para que possa ser transmitido e utilizado por diferentes linguagens de programação, esse objeto é chamado de “performance” e possui quatro propriedades: “type”, “tag”, “lastTypeErrors” e “lastTagErrors”.

A propriedade “type” é um vetor de objetos “type” que tem como função armazenar informações de desempenho relativas aos temas de exercício. A propriedade “tag” é um vetor de objetos “tag” que tem como função armazenar informações de desempenho relativas aos tipos de casos de teste.

As propriedades “lastTypeErrors” e “lastTagErrors” são vetores que tem como função armazenar os últimos X erros cometidos relativos aos temas de exercício e os últimos X erros cometidos relativos aos tipos de casos de teste respectivamente.

Figura 17 - Objeto desempenho do aluno

performance
<p>type (vetor) = lista de objetos type (desempenho de tema de exercício)</p> <p>tag (vetor) = lista de objetos tag (desempenho de tipo de caso de teste)</p> <p>lastTypeErrors (vetor) = fila que contém os temas de exercício dos últimos X exercícios em que ocorreu pelo menos 1 erro</p> <p>lastTagErrors (vetor) = fila que contém os tipos de caso de teste dos últimos X tipos de casos de teste avaliados em que ocorreu um erro</p>

Fonte: Autoria própria

Um objeto “type” possui três propriedades: “type”, “total” e “errors”, a primeira é um texto que tem como função identificar o objeto pelo nome do tema ao qual esse objeto é relacionado, a segunda é um número inteiro que armazena o total de exercícios desse tema realizados, e a terceira é um número inteiro que armazena em quantos desses exercício foram cometidos erros.

Figura 18 - Objeto types

type
<p>type (texto) = tema do exercício</p> <p>total (inteiro) = quantidade de exercícios avaliados desse tema</p> <p>errors (inteiro) = quantidade de exercícios avaliados desse tema em que ocorreu pelo menos 1 erro</p>

Fonte: Autoria própria

Um objetos “tag” possui três propriedades: “tag”, “total” e “errors”, a primeira é um texto que tem como função identificar o objeto pelo nome do tipo de caso de teste ao qual esse objeto é relacionado, a segunda é um número inteiro que armazena o total de exercícios desse tipo de caso de teste realizados, e a terceira é um número inteiro que armazena em quantos desses exercício foram cometidos erros.

Figura 19 - Objeto tag

tag
tag (texto) = tipo de caso de teste
total (inteiro) = quantidade de avaliações desse tipo de caso de teste realizadas
errors (inteiro) = quantidade de avaliações desse tipo de caso de teste em que ocorreu um erro

Fonte: Autoria própria

### 5.3 BANCO DE DADOS

A tabela do banco de dados do Moodle referente ao assistente pedagógico deve armazenar o desempenho do aluno em exercícios de iVProg de um curso. Para que o aluno possa ter seu desempenho dentro do curso armazenado no banco de dados é necessário que haja um campo relativo à identificação do aluno e um campo relativo à identificação do curso no banco de dados.

O desempenho do aluno é armazenado em formato JSON. Assim é possível armazenar o objeto que contém todo o desempenho em um único campo. Isso facilita o gerenciamento do banco de dados, pois o desempenho inteiro é acessado de uma vez só. Caso cada componente do desempenho fosse armazenado em um campo individual, a complexidade do gerenciamento do banco de dados aumentaria de forma considerável, pois em cada execução do assistente pedagógico são alteradas diferentes combinações de componentes do desempenho.

Tabela 3 - Exemplo de registro na tabela do banco de dados do Moodle referente ao assistente pedagógico.

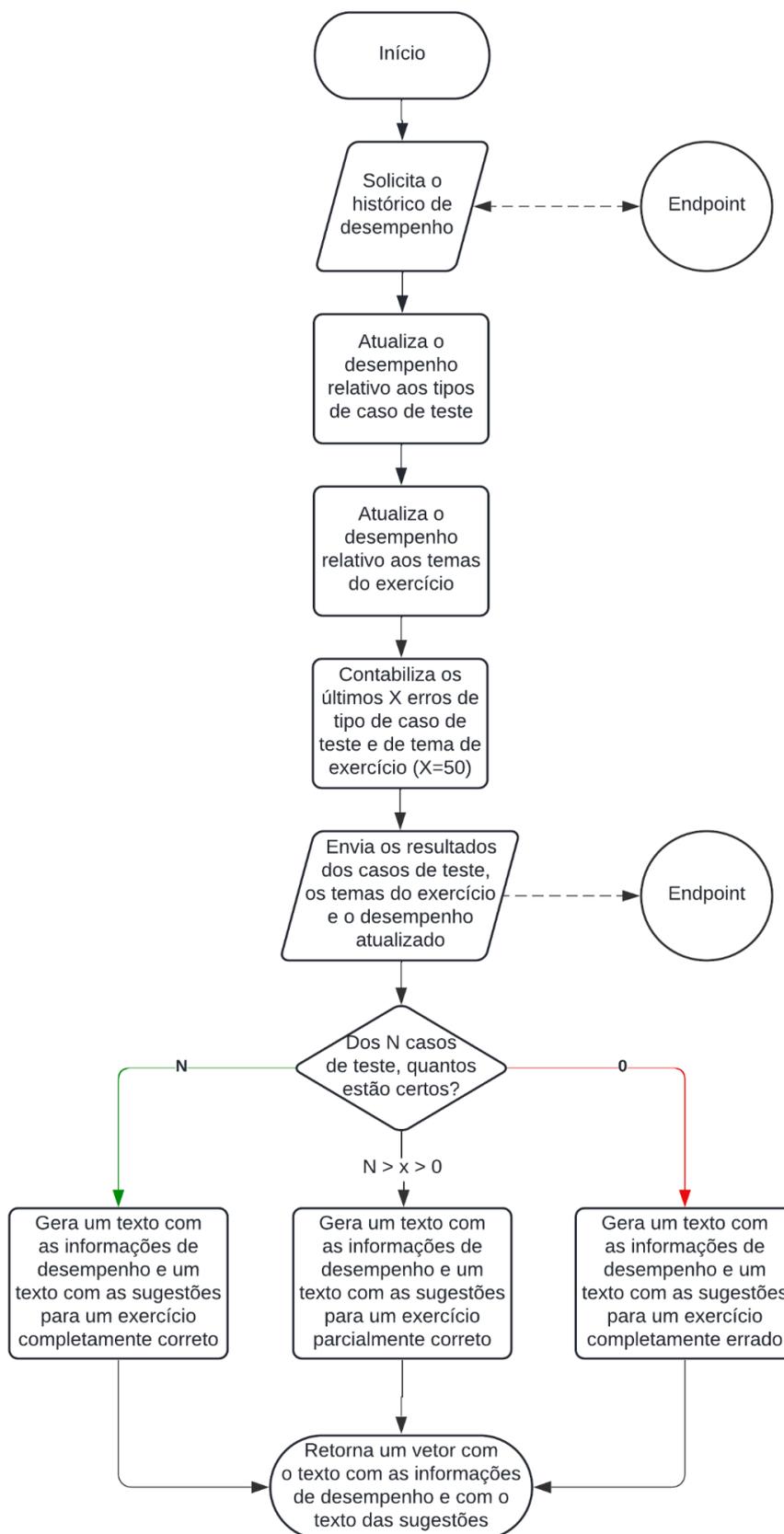
ID	ID do Aluno	ID do Curso	Histórico de Desempenho
[Número]	[Número]	[Número]	[Json]
1	4	132	<pre>{   "type": [ {"type": "Leitura e Escrita", "total" :3, "errors": 0}, {"type": ...} ], "tag": [ {"tag": "Caso Geral", "total": 19, "errors": 0}, {"tag": ...} ], "lastTypeErrors": ["Laço", ...], "lastTagErrors": ["Tipo de Entrada", ...] }</pre>

#### 5.4 ASSISTENTE PEDAGÓGICO

O assistente pedagógico é responsável por atualizar o desempenho de acordo com o resultado do exercício, por gerar um texto com informações do desempenho no exercício e por gerar um texto com sugestões ao usuário.

Em sua comunicação com o endpoint, ele recebe o histórico de desempenho, e envia o desempenho atualizado e os dados utilizados para realizar a atualização.

Figura 20 - Ciclo de vida do assistente pedagógico

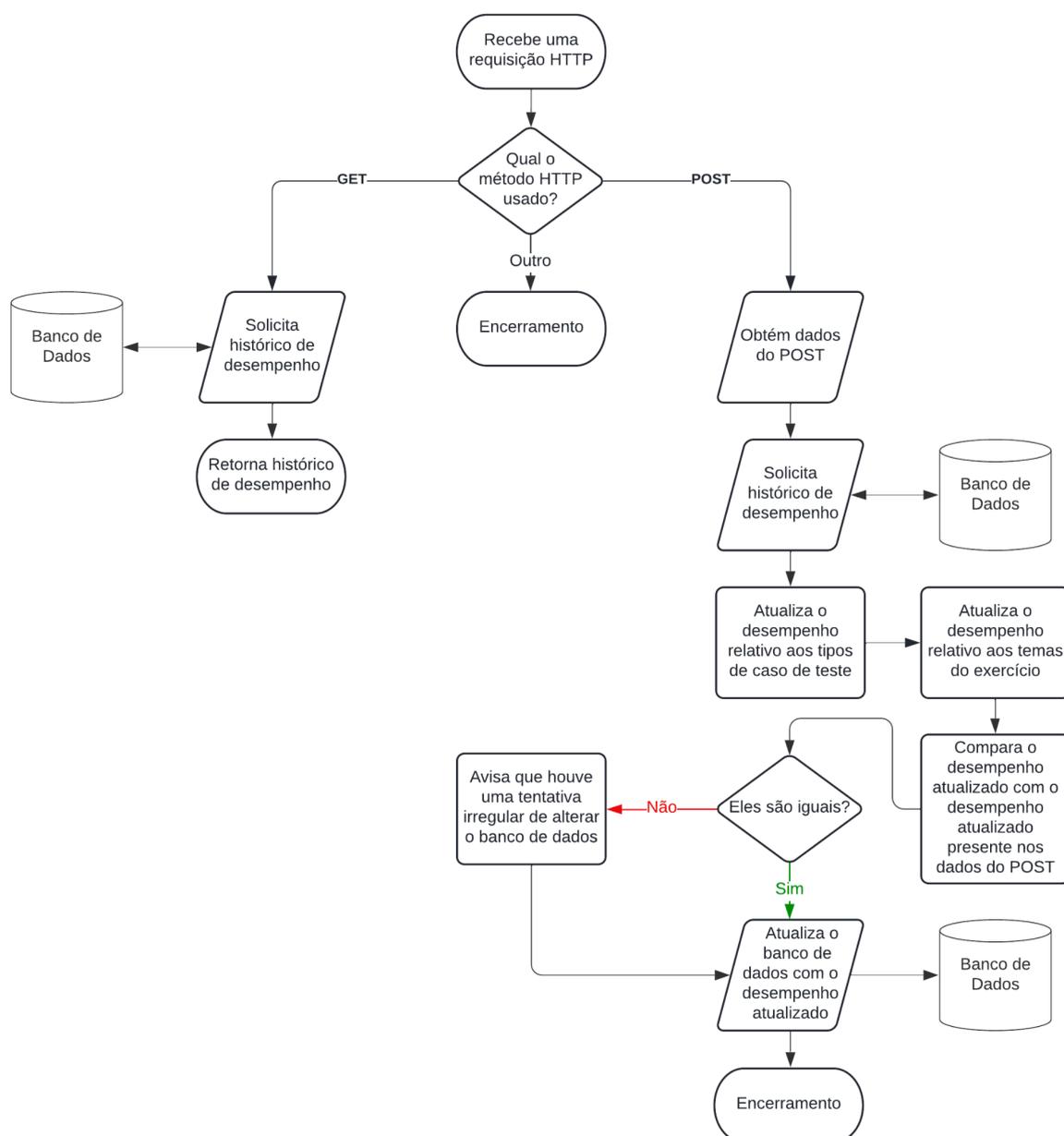


Fonte: Autoria própria

## 5.5 ENDPOINT

O endpoint realiza a comunicação do iTarefa com o assistente pedagógico, viabilizando o acesso e armazenamento de dados no banco de dados do Moodle. Assim como o assistente pedagógico, o endpoint também realiza a atualização do desempenho. Isso é feito para evitar ataques que alterem o banco de dados via alteração do desempenho atualizado enviado pelo assistente pedagógico.

Figura 21 - Ciclo de vida do endpoint



Fonte: Autoria própria

## 6. IMPLEMENTAÇÃO

### 6.1 INTERFACE DE AUTORIA

Para implementar os novos casos de uso definidos para a interface de autoria é necessário modificar o arquivo “iassign-integration-functions.js” presente na pasta “js” do código fonte do iVProg. Essa pasta contém os arquivos .js que serão utilizados para gerar a aplicação do iVProg. O arquivo “iassign-integration-functions.js” contém as funções que realizam a integração da aplicação do iVProg com o iTarefa. As modificações necessárias consistem na adição da classificação aos casos de teste e na adição da classificação de exercício.

#### 6.1.1 Adição da classificação de caso de teste

Para adicionar a classificação aos casos de teste foi necessário alterar a estrutura dos casos de teste nas seguintes funções: “prepareTestCases()”, “prepareTableTestCases(div\_el)” e “addTestCase(test\_case)”. Além disso, foi necessário adicionar as possíveis classificações na função “addTestCase(test\_case)”.

#### 6.1.2 Adição da classificação de exercício

Para adicionar a classificação de exercício foi necessário criar funções semelhantes às funções relativas aos casos de teste já existentes, portanto foram criadas as seguintes funções: “prepareExerciseType()”, “prepareTableExerciseType(div\_el)”, “addType(exercise\_type)” e “updateTypeCounter()”.

Além disso, foi necessário adicionar essas novas funções criadas em outras funções que contém funções relativas aos casos de teste já existentes. As novas funções foram baseadas nestas últimas. Portanto foi adicionado o uso dessas novas

funções nas seguintes funções: `prepareActivityToEdit(ilm_cont)`, `prepareActivityCreation()` e `getAnswer()`.

### 6.1.3 Propagação das classificações adicionadas

Como a classificação dos casos de teste foi adicionada aos casos de teste que já existiam no iVProg, não foi necessário realizar alterações para propagar a sua classificação. A classificação de exercício por outro lado, precisa ser propagada ao longo do código fonte do iVProg, portanto é necessário realizar algumas modificações em arquivos do código fonte.

Para adicionar a classificação de exercício à avaliação automática foi necessário adicioná-la como atributo ao construtor das classes exportáveis `Assessment(ast_code, exerciseType, testCases, domConsole)` e `IVProgAssessment(ast_code, exerciseType, testCases, domConsole)`, presentes nos arquivos `assessment.js` e `ivprogAssessment.js` respectivamente, estando ambos presentes na pasta `assessment` da pasta `js`. A pasta `assessment` contém os arquivos relativos ao avaliador automático que são utilizados para gerar a aplicação do iVProg. O arquivo `assessment.js` define a classe exportável `Assessment(ast_code, exerciseType, testCases, domConsole)`, relativa a tarefa que passa pela avaliação automática. O arquivo `ivprogAssessment.js` define a classe exportável `IVProgAssessment(ast_code, exerciseType, testCases, domConsole)`, filha da classe `Assessment(ast_code, exerciseType, testCases, domConsole)`, relativa ao resultado da avaliação automática do iVProg.

Em função da adição da classificação de exercício, como atributo no construtor da classe `IVProgAssessment(ast_code, exerciseType, testCases, domConsole)`, foi necessário adicionar esse atributo no uso dessa classe no arquivo `functions.js` presente na pasta `visualUI` da pasta `js` e no arquivo `iassignHelpers.js` presente na pasta `util` da pasta `js`. A pasta `visualUI` contém os arquivos relativos à interface de programação visual que serão utilizados para gerar a aplicação do iVProg. O arquivo `functions.js`

exporta a função de avaliação automática e exporta a função de preparar a atividade para o estudante. A pasta “util” contém os arquivos que possuem funções que serão reaproveitadas e utilizadas em várias partes dos códigos. O arquivo “iassignHelpers.js” exporta a função de avaliação automática e exporta a função de preparar o exercício para o estudante. Para que se possa realizar a determinação e a aquisição dos temas do exercício foram criadas respectivamente as funções exportáveis “setExerciseType(exerciseType)” e “getExerciseType()” no arquivo “functions.js”. Para que seja possível realizar a determinação dos temas do exercício na preparação do exercício para o estudante, foi adicionada a importação e o uso da função “setExerciseType(exerciseType)” no arquivo “iassignHelpers.js”. Para que seja possível realizar a aquisição dos temas do exercício foi adicionada a importação e a exportação da função “getExerciseType()” no arquivo “main.js”. Com isso é possível utilizar essa função na função “prepareActivityToEdit(ilm\_cont)” do arquivo “iassign-integration-functions.js”. O arquivo “main.js” reúne as funções necessárias para a execução da aplicação do iVProg, importando e exportando essas funções.

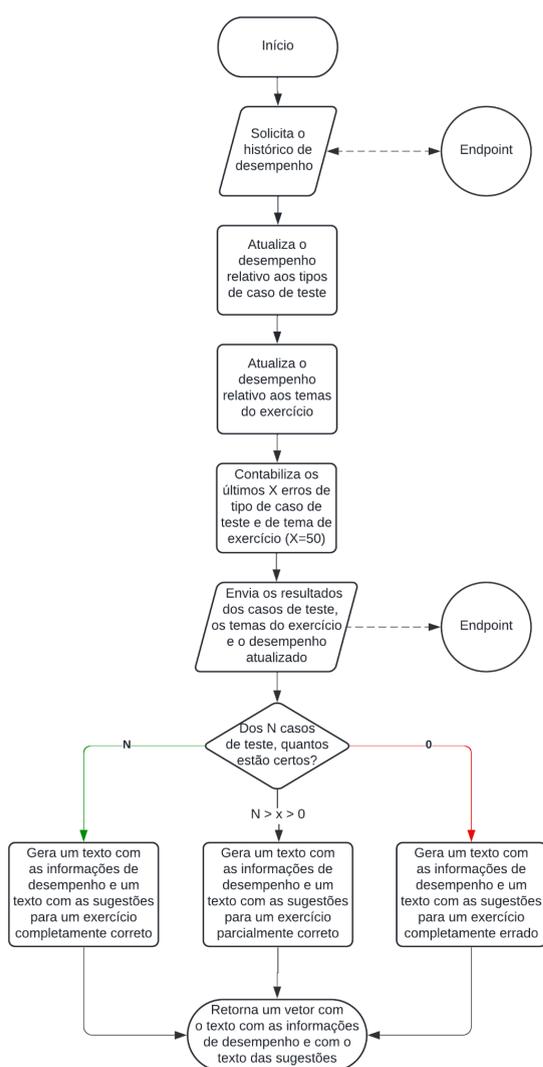
## 6.2 ASSISTENTE PEDAGÓGICO

Para implementar o assistente pedagógico foi criada a pasta “assistant” na pasta “js” do código fonte do iVProg, com a finalidade de conter os arquivos relativos ao assistente pedagógico que serão utilizados para gerar a aplicação do iVProg. Dentro da pasta “assistant” foi criado o arquivo “assistant.js” para armazenar a definição da classe exportável “Assistant()”. A classe exportável “Assistant()” apresenta os seguintes métodos: “get(url)”, “updateTags(testCases, performance)”, “updatedTypes(exerciseType, performance)”, “lastErrors(performance)”, “post(url, data)”, “addPerformance(exerciseType, wrongTags, performance)”, “findObjects(array, key, value)”, “addSuggestion(exerciseType,

wrongTags)” e “analise(url, exerciseType, testCases)”, e possui o atributo "suggestions".

O método “analise(url, exerciseType, testCases)” executa o assistente pedagógico, encadeando os outros métodos de forma que seja cumprido o ciclo de vida projetado para o assistente pedagógico. Ele recebe como entrada um texto com o endereço do endpoint responsável pela comunicação do iTarefa com a aplicação do iVProg que viabiliza o acesso e armazenamento de dados no banco de dados do Moodle, um vetor com os temas do exercício e um vetor com o resultado dos casos teste. Ele retorna um vetor contendo um texto com as informações de desempenho e um texto com as sugestões.

Figura 22 - Ciclo de vida do assistente pedagógico



Fonte: Autoria própria

### **6.2.1 Solicitação do histórico de desempenho**

A solicitação do histórico de desempenho é feita através do uso do método “`get(url)`”. Ele recebe o endereço do `endpoint` responsável pela comunicação do iTarefa com a aplicação do iVProg que viabiliza o acesso e armazenamento de dados no banco de dados do Moodle, e retorna um JSON com histórico de desempenho obtido do banco de dados do Moodle.

### **6.2.2 Atualização do desempenho relativo aos tipos de caso de teste**

O método “`updateTags(testCases, performance)`” atualiza o desempenho relativo aos tipos de casos de teste do exercício, sinaliza se houve a ocorrência de algum erro no exercício, sinaliza se o usuário errou todos os casos de teste e define a quantidade de erros relativos aos tipos de casos de teste a serem contabilizados. Possui como entradas o resultado dos casos de teste do exercício e o histórico de desempenho do usuário.

### **6.2.3 Atualização do desempenho relativo aos temas do exercício**

O método “`updateTypes(exerciseType, performance)`” atualiza o desempenho relativo aos temas do exercício e define a quantidade de erros relativos aos temas de exercício a serem contabilizados. Possui como entradas os temas do exercício e o histórico de desempenho do usuário.

### **6.2.4 Contabilização dos últimos X erros**

A contabilização dos últimos erros é realizada pelo método “`lastErrors(performance)`”, utilizando as quantidades de erros definidas previamente nas atualizações de desempenho para contabilizar as últimas ocorrências de erros relativos aos temas de exercício e de erros relativos aos tipos de casos de teste. Possui como entrada o histórico de desempenho do usuário.

### 6.2.5 Envio de dados ao endpoint

O envio de dados ao `endpoint` responsável pela comunicação do iTarefa com a aplicação do iVProg que viabiliza o acesso e armazenamento de dados no banco de dados do Moodle é realizado através do uso do método `post(url, data)`. Ele recebe o endereço desse `endpoint` e um objeto que contém os dados que devem ser enviados. Esse objeto é transformado em JSON antes do envio, e ele é composto por um vetor que contém os temas do exercício, por um vetor que contém os resultados dos casos de teste e por um objeto com o desempenho atualizado.

### 6.2.6 Geração do texto com as informações de desempenho

A geração do texto com as informações de desempenho do usuário em relação ao resultado obtido no exercício executado é feita pelo método `addPerformance(exerciseType, wrongTags, performance)`, que gera e retorna esse texto. Possui como entradas os temas do exercício, um vetor com os tipos de casos de teste do exercício em que ocorreu um erro e o histórico de desempenho atualizado.

### 6.2.7 Geração do texto com as sugestões

A geração do texto com as sugestões ao usuário em relação ao resultado obtido no exercício executado é realizada através do método `addSuggestion(exerciseType, wrongTags)`. Ele gera esse texto utilizando o atributo `suggestions` e retorna o texto gerado. Possui como entradas os temas do exercício e um vetor com os tipos de casos de teste do exercício em que ocorreu um erro.

O atributo `suggestions` contém um vetor de objetos, onde cada um dos objetos apresenta um texto de sugestão para um tema de exercício ou para um tipo de caso de teste. Nesses textos é possível utilizar imagens e *links*.

Para que as imagens sejam exibidas pela aplicação do iVProg é necessário adicionar essas imagens à pasta “img” presente no código fonte do iVProg. Essa pasta tem como função conter as imagens que serão utilizadas pela aplicação do iVProg. Também é necessário adicionar os caminhos para as imagens adicionadas à pasta “img” no arquivo “webpack.config.js” presente no código fonte do iVProg. Ele é responsável por empacotar os módulos estáticos do código do iVProg para a aplicação do iVProg, gerando um gráfico que mapeia cada módulo e suas dependências.

### **6.2.8 Retorno de dados**

O retorno dos dados gerados na execução do assistente pedagógico é realizado ao fim da execução do método “`analise(url, exerciseType, testCases)`”, no qual é retornado um vetor contendo o texto gerado com as informações de desempenho e o texto gerado com as sugestões.

### **6.2.9 Integração com a aplicação do iVProg**

Foi necessário alterar alguns arquivos já existentes no código fonte do iVProg a fim de se integrar o assistente pedagógico à aplicação do iVProg. Para que os textos gerados pelo assistente pedagógico sejam exibidos em uma página web foi necessário realizar alterações no arquivo “`assessment_result.js`” da pasta “`output_matching`”, presente na pasta “`assessment`”. A pasta “`output_matching`” contém os arquivos relativos à comparação realizada pelo avaliador automático entre saídas obtidas e esperadas. O arquivo “`assessment_result.js`” define a classe exportável “`OutputAssessmentResult()`”, relativa ao resultado da avaliação automática. Essa classe utiliza propriedades estáticas que retornam texto em formato HTML para definir trechos de página web.

A primeira alteração realizada no arquivo foi criar a propriedade estática “`ASSISTANT_TEMPLATE()`”, que retorna um texto em formato HTML. O texto retornado é um trecho específico de página web referente a interface do assistente

pedagógico na aplicação do iVProg. É esse trecho que diferencia essa página web das outras geradas pela aplicação do iVProg. A segunda alteração realizada no arquivo foi a criação do método `generateAssistantOutput(performanceText, suggestionText)`, que tem como função gerar a página web referente a interface do assistente pedagógico na aplicação do iVProg, ele recebe como entrada o texto que contém as informações de desempenho no exercício e o texto que contém as sugestões.

Para definir os estilos que serão utilizados na página web referente a interface do assistente pedagógico na aplicação do iVProg foi necessário modificar o arquivo `ivprog-assessment.css`, que está contido na pasta `css`. Essa pasta contém os arquivos `css` que serão utilizados pela aplicação do iVProg. O arquivo `ivprog-assessment.css` define os estilos que serão utilizados nas páginas web geradas a partir dos dados do avaliador automático. As modificações realizadas nesse arquivo consistiram na criação das classes `.assistant-div-left` e `.assistant-div-right`. A classe `.assistant-div-left` define o estilo que deve ser aplicado às informações de desempenho que são exibidas ao usuário na página web referente a interface do assistente pedagógico na aplicação do iVProg, já a classe `.assistant-div-right` define o estilo que deve ser aplicado às sugestões.

Assim como ocorre com as páginas web referentes aos casos de teste na versão atual do iVProg (versão IV), para que a página web referente a interface do assistente pedagógico na aplicação do iVProg seja contida em um texto que pode ser exibido no terminal da aplicação do iVProg, foi necessário alterar o arquivo `message.csv` presente na pasta `i18n`. Essa pasta contém os arquivos que serão utilizados para aplicar Internacionalização na aplicação do iVProg, ou seja, contém os arquivos que vão permitir que a aplicação tenha mais de uma opção de idioma. O arquivo `message.csv` define as mensagens que podem ser exibidas no terminal da aplicação do iVProg. Os identificadores dessas mensagens são chamados de `ids`. A alteração realizada foi a adição da mensagem do assistente pedagógico que deve ser exibida no terminal da aplicação do iVProg após a execução do avaliador automático. O identificador (id) definido para essa mensagem foi `assistant_content`.

Para que o assistente pedagógico seja utilizado ao se realizar uma avaliação automática na aplicação do iVProg foi necessário realizar alterações no arquivo “ivprogAssessment.js” da pasta “assessment”, a fim de se adicionar a execução do assistente pedagógico. No arquivo “ivprogAssessment.js” ocorre a definição da classe exportável “IVProgAssessment(ast\_code, exerciseType, testCases, domConsole)”, filha da classe “Assessment(ast\_code, exerciseType, testCases, domConsole)”, relativa ao resultado da avaliação automática do iVProg. A primeira alteração realizada foi a adição da importação da classe “Assistant()” do arquivo “assistant.js” contido na pasta “assistant”. A segunda alteração necessária foi a adição da execução do assistente após a obtenção da nota total da avaliação automática no método “runTest()”. Para isso foi adicionada a criação de um objeto da classe “Assistant()”. A este objeto é aplicado o método “analise(url, exerciseType, testCases)”, recebendo como entradas: um texto com o endereço do endpoint responsável pela comunicação do iTarefa com a aplicação do iVProg, que viabiliza o acesso e armazenamento de dados no banco de dados do Moodle; um vetor com os temas do exercício; e um vetor com o resultado dos casos teste. Após o término da execução do método “analise(url, exerciseType, testCases)” é escrita a mensagem “assistant\_content” no terminal da aplicação do iVProg. Em seguida, executa-se o método “generateAssistantOutput(performanceText, suggestionText)” para gerar a página web referente a interface do assistente pedagógico. Essa página está contida na mensagem “assistant\_content”.

### 6.3 BANCO DE DADOS

Para implementar o banco de dados foi necessário alterar o arquivo “install.xml” contido na pasta “db” do código fonte do iTarefa. A pasta “db” armazena todos os arquivos relacionados ao banco de dados do iTarefa. O arquivo “install.xml” contém a estrutura das tabelas do banco de dados do iTarefa.

A alteração realizada consiste na adição da estrutura da tabela que deve armazenar o histórico de desempenho dos usuários nos exercícios do iVProg de

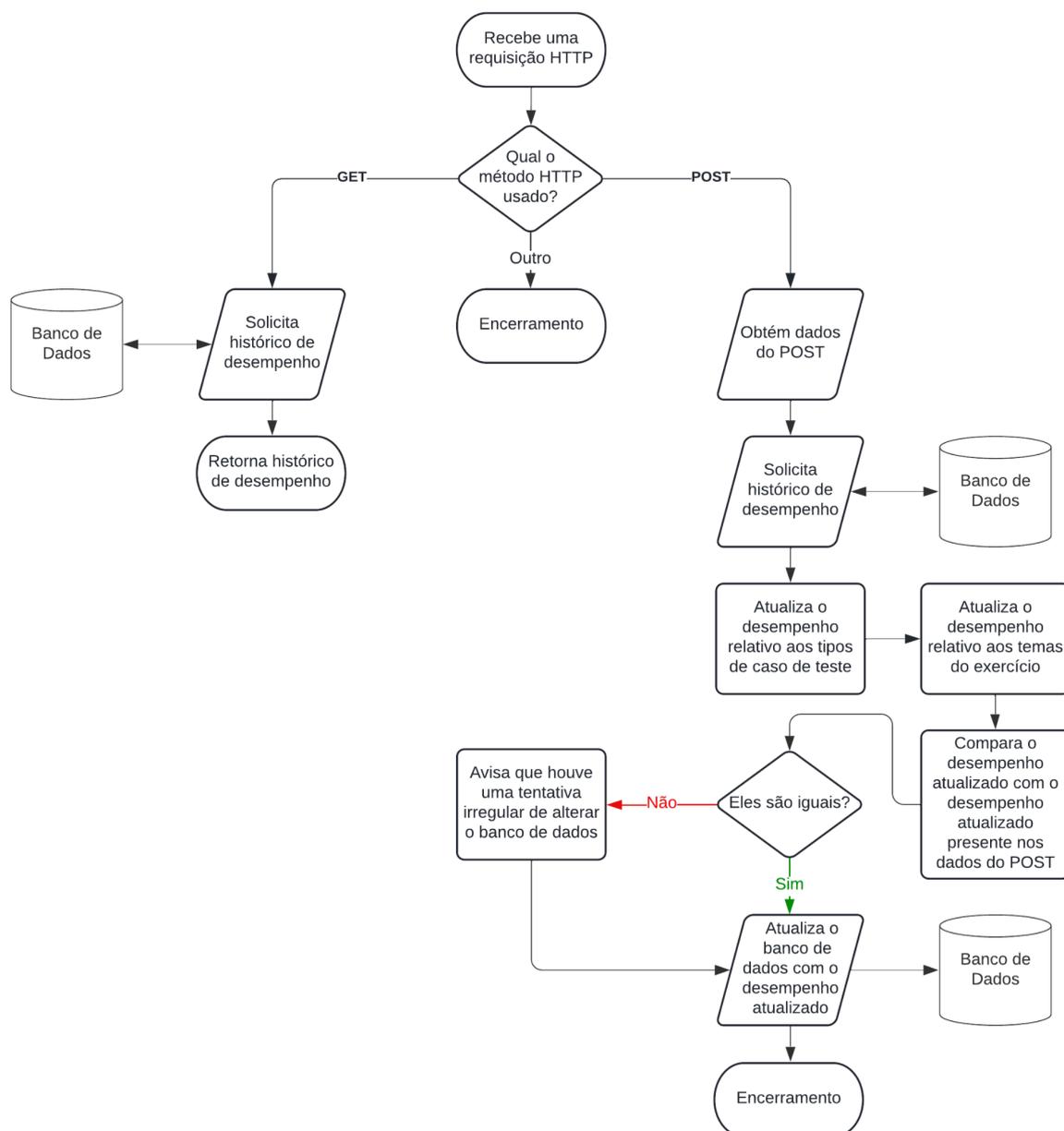
acordo com seu curso. Essa tabela foi nomeada como “iassign\_performance”, e é composta por: número de registro (id); número de registro do aluno; número de registro do curso; e histórico de desempenho do aluno no curso.

#### 6.4 ENDPOINT

Para implementar o Endpoint foi criado o arquivo “endpoint.php” no código fonte do iTarefa. Esse arquivo tem caráter provisório, pois o ideal é que seu código seja incorporado em um local apropriado do código fonte do iTarefa, local onde estejam códigos que realizam tarefas semelhantes a essa. Esta alteração é fortemente sugerida para futuras versões do Endpoint.

O arquivo “endpoint.php” contém a classe “Performance()” e os códigos para a comunicação do iTarefa com a aplicação do iVProg que viabilizam o acesso e armazenamento de dados no banco de dados do Moodle. A classe “Performance()” tem “emptyPerformance” como atributo privado, apresenta os seguintes métodos privados: “add(string \$course, string \$student)”, “updateTypes(array \$exerciseType, array \$types, object \$performance, bool \$incorrect)” e “updateTags(array \$testCases, array \$tags, object \$performance)”, e os seguintes métodos públicos: “find(string \$course, string \$student)” e “update(string \$course, string \$student, array \$testCases, array \$exerciseType, object \$jsPerformance)”.

Figura 23 - Ciclo de vida do endpoint



Fonte: Autoria própria

#### 6.4.1 Identificação do método HTTP usado

A identificação do método HTTP usado é feita de forma contínua pelo “endpoint.php”, ou seja, ele realizará a identificação sempre que fizerem uma requisição HTTP para ele. Se for detectado o uso do método HTTP GET será retornado o registro de desempenho do usuário no curso onde está sendo executada a aplicação do iVProg que fez essa requisição. Se for detectado o uso do

método HTTP POST será atualizado o registro de desempenho do usuário no curso onde está sendo executada a aplicação do iVProg que fez essa requisição. Se for detectado o uso de outro método HTTP nenhuma ação é realizada. O usuário precisa estar logado para que os códigos sejam executados.

#### **6.4.2 Solicitação do histórico de desempenho**

A solicitação do histórico de desempenho é feita através do uso do método público `find(string $course, string $student)`. Ele retorna um registro de desempenho de um usuário contido na tabela `iassign_performance` do banco de dados a partir do número de registro (id) do aluno e do curso fornecidos. Caso não haja registro para a combinação fornecida de número de registro do aluno e do curso, aciona-se o método privado `add(string $course, string $student)` antes de retornar o registro.

O método privado `add(string $course, string $student)` usa o valor do atributo privado `emptyPerformance` para criar o registro de desempenho de um usuário na tabela `iassign_performance` do banco de dados a partir do número de registro (id) do aluno e do curso fornecidos. O atributo privado `emptyPerformance` é um texto em formato de JSON que contém o estado inicial do registro de desempenho de um usuário.

#### **6.4.3 Retorno do histórico de desempenho**

O retorno do histórico de desempenho é realizado no próprio endpoint, dessa forma ele pode ser acessado pelo método HTTP GET caso o usuário esteja logado.

#### **6.4.4 Obtenção dos dados do POST**

Ao se identificar uma requisição HTTP que utiliza o método POST, o `endpoint.php` realiza a obtenção dos dados presentes na requisição. A obtenção dos dados é realizada armazenando esses dados recebidos no formato JSON em uma variável com o formato de objeto. Esse objeto é aquele que é enviado pelo

assistente pedagógico contendo os dados que o `endpoint` deve receber. Ele é composto por: um vetor que contém os resultados dos casos de teste; um vetor que contém os temas do exercício; e um objeto com o desempenho atualizado pelo assistente pedagógico na aplicação do iVProg.

#### **6.4.5 Atualização do desempenho relativo aos tipos de caso de teste**

O método privado `updateTags(array $testCases, array $tags, object $performance)` atualiza o desempenho relativo aos tipos de casos de teste e retorna um booleano que indica se houve a ocorrência de algum erro em algum dos casos de teste. Ele recebe como entrada: um vetor que contém os resultados dos casos de teste; um vetor com o histórico de desempenho relativo aos tipos de casos de teste; e um objeto com o histórico de desempenho.

#### **6.4.6 Atualização do desempenho relativo aos temas do exercício**

O método privado `updateTypes(array $exerciseType, array $types, object $performance, bool $incorrect)` atualiza o desempenho relativo aos temas de exercício. Ele recebe como entrada: um vetor que contém os temas do exercício; um vetor com o histórico de desempenho relativo aos temas de exercício; um objeto com o histórico de desempenho; e um booleano que indica se houve a ocorrência de algum erro em algum dos casos teste.

#### **6.4.7 Comparação do desempenho atualizado com o recebido**

O método público `update(string $course, string $student, array $testCases, array $exerciseType, object $jsPerformance)` realiza a comparação do desempenho atualizado pelos métodos privados `updateTags(array $testCases, array $tags, object $performance)` e `updateTypes(array $exerciseType, array $types, object $performance, bool $incorrect)` com o desempenho atualizado pelo assistente pedagógico na aplicação do iVProg que foi recebido via método

HTTP POST. Caso a comparação indique que os desempenhos são diferentes, informa-se que houve uma tentativa irregular de alterar o banco de dados.

#### 6.4.8 Atualização do banco de dados

A atualização do banco de dados também é realizada pelo método público `update(string $course, string $student, array $testCases, array $exerciseType, object $jsPerformance)`, através dele atualiza-se o registro do histórico de desempenho contido na tabela `"iassign_performance"` do banco de dados com o desempenho atualizado pelos métodos privados `updateTags(array $testCases, array $tags, object $performance)` e `updateTypes(array $exerciseType, array $types, object $performance, bool $incorrect)`.

O método público `update(string $course, string $student, array $testCases, array $exerciseType, object $jsPerformance)` realiza todo o processo de atualização do histórico de desempenho de um usuário contido na tabela `"iassign_performance"` do banco de dados, encadeando métodos descritos nas seções anteriores. Ele recebe como entrada: um texto com o número de registro (id) do aluno; um texto com o número de registro (id) do curso; um vetor que contém os temas do exercício; um vetor que contém os resultados dos casos de teste; e um objeto com o desempenho atualizado pelo assistente pedagógico na aplicação do iVProg. Ele utiliza o método público `find(string $course, string $student)` para obter o registro do histórico de desempenho. Em seguida utiliza os métodos privados `updateTags(array $testCases, array $tags, object $performance)` e `updateTypes(array $exerciseType, array $types, object $performance, bool $incorrect)` para atualizar o desempenho. Após isso, compara esse desempenho atualizado com o objeto recebido que contém o desempenho atualizado pelo assistente pedagógico na aplicação do iVProg. Caso haja alguma diferença informa que houve uma tentativa irregular de alterar o banco de dados. Em seguida atualiza o registro de desempenho contido na tabela `"iassign_performance"` do banco de dados com o desempenho atualizado pelos

métodos privados “updateTags(array \$testCases, array \$tags, object \$performance)” e “updateTypes(array \$exerciseType, array \$types, object \$performance, bool \$incorrect)”.

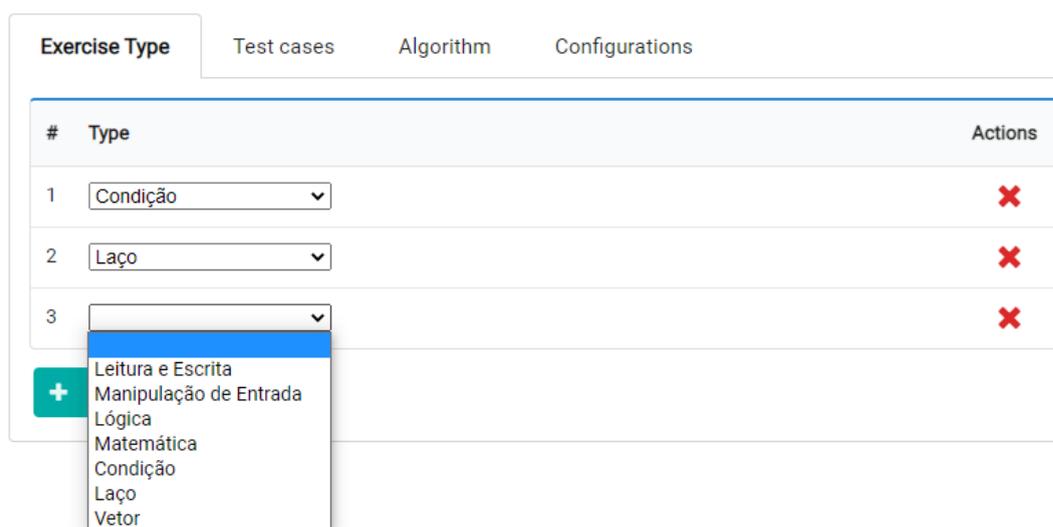
## 7. TESTES E AVALIAÇÃO

Para possibilitar a realização dos testes do assistente pedagógico desenvolvido e a verificação do recepcionamento dos requisitos funcionais, foram desenvolvidos exercícios baseados em exercícios de iVProg presentes no curso “2021: Introdução à Programação (Diurno)” do SAW. Os exercícios foram desenvolvidos de forma que contemplem o caso de uso atual do iVProg e de forma que englobe todos os temas de exercício e tipos de casos de teste definidos. O enunciado dos exercícios e o seus arquivos `.ivph` que contém a descrição de tarefa do avaliador estão no Apêndice A.

### 7.1 TESTAGEM DO RF1

Para verificar se o Requisito Funcional RF1, “o sistema deve permitir a classificação de exercícios na interface de autoria”, foi recepcionado, todos os exercícios de iVProg desenvolvidos foram criados na interface de autoria do iVProg no Moodle através do módulo de extensão iTarefa. A criação dos exercícios englobou todas as classificações definidas, ou seja, englobou todos os temas de exercício e tipos de casos de teste definidos.

Figura 24 - Classificação de exercício na interface de autoria



Fonte: Execução da versão deste projeto do iVProg

Figura 25 - Classificação de casos de teste na interface de autoria

#	Input	Output	Tag	Actions
1	5	Sim	Caso Geral	✗
2	23	Não	Caso Limite	✗
3	0	Sim	Caso Nulo	✗
4	-3	Não	Caso Negativo	✗
5	1	Sim	Caso Geral	✗

Fonte: Execução da versão deste projeto do iVProg

Figura 26 - Arquivo .i\_vph que contém a descrição de tarefa do avaliador

```

{
  "exercisetype" : ["Condição","Laço","Vetor" ],
  "testcases" : [
    {
      "input": ["5"],
      "output": ["Sim"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["23"],
      "output": ["Não"],
      "tag": ["Caso Limite"]
    },
    {
      "input": ["0"],
      "output": ["Sim"],
      "tag": ["Caso Nulo"]
    },
    {
      "input": ["-3"],
      "output": ["Não"],
      "tag": ["Caso Nulo"]
    },
    {
      "input": ["1"],
      "output": ["Sim"],
      "tag": ["Caso Geral"]
    }
  ],
  "settings_programming_type":
  [{"name":"programming_type","value":"visual"}],
  "settings_data_types":
  [{"name":"integer_data_type","value":"on"}, {"name":"real_data_type","value":"on"},
  {"name":"text_data_type","value":"on"}, {"name":"boolean_data_type","value":"on"},
  {"name":"void_data_type","value":"on"}],
  "settings_commands":
  [{"name":"commands_read","value":"on"}, {"name":"commands_write","value":"on"},
  {"name":"commands_comment","value":"on"}, {"name":"commands_attribution","value":"on"},
  {"name":"commands_functioncall","value":"on"},
  {"name":"commands_iftrue","value":"on"},
  {"name":"commands_repeatNtimes","value":"on"}, {"name":"commands_while","value":"on"},
  {"name":"commands_dowhile","value":"on"}, {"name":"commands_switch","value":"on"}],
  "settings_functions":

```

Fonte: Execução da versão deste projeto do iVProg

Ao longo da testagem sempre foi possível selecionar todas as classificações definidas, e todas foram selecionadas, além disso o arquivo de exercício gerado pela interface de autoria apresentou o formato correto em todos os testes.

## 7.2 TESTAGEM DO RF2

Para verificar se o Requisito Funcional RF2, “o sistema deve ser capaz de se comunicar com o iTarefa para viabilizar o acesso e armazenamento de dados no banco de dados do Moodle”, foi recepcionado, foram realizadas avaliações automáticas de todos os exercícios desenvolvidos três vezes: uma vez para um exercício completamente correto; uma vez para um exercício parcialmente correto; e uma vez para um exercício completamente incorreto. As avaliações foram realizadas em quatro usuários diferentes do Moodle. A execução das avaliações dessa forma tem como objetivo verificar se os dados são gerados e armazenados corretamente no banco de dados do Moodle.

Para auxiliar na testagem foi construída uma página que retorna a tabela completa de desempenho do iVProg que está armazenada no banco de dados do Moodle. Por questões de segurança e privacidade esta página só existe no ambiente de testagem.

Figura 27 - Tabela de desempenho antes da primeira avaliação automática de um exercício em um curso de um novo aluno

```

tag: Caso Geral, total: 83, errors: 11
tag: Caso Limite, total: 11, errors: 0
tag: Caso Nulo, total: 17, errors: 1
tag: Caso Negativo, total: 16, errors: 0
tag: Igualdade, total: 9, errors: 2
tag: Tipo de Entrada, total: 10, errors: 0
lastTypeErrors: Matemática, Matemática, Leitura e Escrita
lastTagErrors: Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Geral, Igualdade,
Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Nulo, Igualdade,
Caso Geral

id: 2, course: 115, student: 2
performance:
type: Leitura e Escrita, total: 0, errors: 0
type: Manipulação de Entrada, total: 0, errors: 0
type: Lógica, total: 0, errors: 0
type: Matemática, total: 3, errors: 2
type: Condição, total: 1, errors: 0
type: Laço, total: 1, errors: 0
type: Vetor, total: 3, errors: 2
tag: Caso Geral, total: 33, errors: 16
tag: Caso Limite, total: 6, errors: 2
tag: Caso Nulo, total: 7, errors: 3
tag: Caso Negativo, total: 6, errors: 4
tag: Igualdade, total: 6, errors: 3
tag: Tipo de Entrada, total: 4, errors: 2
lastTypeErrors: Matemática, Matemática, Vetor, Vetor
lastTagErrors: Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Geral,
Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Nulo, Caso Geral, Caso Nulo,
Caso Negativo, Caso Negativo, Caso Geral, Caso Limite, Tipo de Entrada, Caso Geral,
Igualdade, Igualdade, Caso Geral, Caso Nulo, Caso Negativo, Caso Negativo, Caso
Geral, Caso Limite, Tipo de Entrada, Caso Geral, Igualdade

```

Fonte: Execução da versão deste projeto do iVProg

Figura 28 - Tabela de desempenho após a primeira avaliação automática de exercício em um curso de um novo aluno

```

--
type: Condição, total: 1, errors: 0
type: Laço, total: 1, errors: 0
type: Vetor, total: 3, errors: 2
tag: Caso Geral, total: 33, errors: 16
tag: Caso Limite, total: 6, errors: 2
tag: Caso Nulo, total: 7, errors: 3
tag: Caso Negativo, total: 6, errors: 4
tag: Igualdade, total: 6, errors: 3
tag: Tipo de Entrada, total: 4, errors: 2
lastTypeErrors: Matemática, Matemática, Vetor, Vetor
lastTagErrors: Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Geral,
Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Nulo, Caso Geral, Caso Nulo,
Caso Negativo, Caso Negativo, Caso Geral, Caso Limite, Tipo de Entrada, Caso Geral,
Igualdade, Igualdade, Caso Geral, Caso Nulo, Caso Negativo, Caso Negativo, Caso
Geral, Caso Limite, Tipo de Entrada, Caso Geral, Igualdade

id: 3, course: 115, student: 5
performance:
type: Leitura e Escrita, total: 0, errors: 0
type: Manipulação de Entrada, total: 0, errors: 0
type: Lógica, total: 0, errors: 0
type: Matemática, total: 0, errors: 0
type: Condição, total: 1, errors: 0
type: Laço, total: 0, errors: 0
type: Vetor, total: 0, errors: 0
tag: Caso Geral, total: 2, errors: 0
tag: Caso Limite, total: 0, errors: 0
tag: Caso Nulo, total: 2, errors: 0
tag: Caso Negativo, total: 2, errors: 0
tag: Igualdade, total: 0, errors: 0
tag: Tipo de Entrada, total: 1, errors: 0
lastTypeErrors:
lastTagErrors:

```

Fonte: Execução da versão deste projeto do iVProg

Figura 29 - Tabela de desempenho após a segunda avaliação automática de exercício desse novo aluno no mesmo curso

```

type: Condição, total: 1, errors: 0
type: Laço, total: 1, errors: 0
type: Vetor, total: 3, errors: 2
tag: Caso Geral, total: 33, errors: 16
tag: Caso Limite, total: 6, errors: 2
tag: Caso Nulo, total: 7, errors: 3
tag: Caso Negativo, total: 6, errors: 4
tag: Igualdade, total: 6, errors: 3
tag: Tipo de Entrada, total: 4, errors: 2
lastTypeErrors: Matemática, Matemática, Vetor, Vetor
lastTagErrors: Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Geral,
Caso Geral, Caso Geral, Caso Geral, Caso Geral, Caso Nulo, Caso Geral, Caso Nulo,
Caso Negativo, Caso Negativo, Caso Geral, Caso Limite, Tipo de Entrada, Caso Geral,
Igualdade, Igualdade, Caso Geral, Caso Nulo, Caso Negativo, Caso Negativo, Caso
Geral, Caso Limite, Tipo de Entrada, Caso Geral, Igualdade

id: 3, course: 115, student: 5
performance:
type: Leitura e Escrita, total: 0, errors: 0
type: Manipulação de Entrada, total: 0, errors: 0
type: Lógica, total: 0, errors: 0
type: Matemática, total: 0, errors: 0
type: Condição, total: 2, errors: 1
type: Laço, total: 1, errors: 1
type: Vetor, total: 1, errors: 1
tag: Caso Geral, total: 4, errors: 0
tag: Caso Limite, total: 1, errors: 1
tag: Caso Nulo, total: 4, errors: 1
tag: Caso Negativo, total: 2, errors: 0
tag: Igualdade, total: 0, errors: 0
tag: Tipo de Entrada, total: 1, errors: 0
lastTypeErrors: Condição, Laço, Vetor
lastTagErrors: Caso Limite, Caso Nulo

```

Fonte: Execução da versão deste projeto do iVProg

Ao longo da testagem sempre observou-se que os registros são gerados de forma correta no banco de dados do Moodle e que também são atualizados de forma correta.

A evolução da tabela de desempenho de um novo aluno em um curso exibida nas figuras acima é compilada nas tabelas a seguir:

Tabela 4 - Evolução da tabela de desempenho relativa aos temas de exercício

type	Pré 1ª Avaliação		Pós 1ª Avaliação		Pós 2ª Avaliação	
	total	errors	total	errors	total	errors
Leitura e Escrita	-----	-----	0	0	0	0
Manipulação de Entrada	-----	-----	0	0	0	0
Lógica	-----	-----	0	0	0	0
Matemática	-----	-----	0	0	0	0
Condição	-----	-----	1	0	2	1
Laço	-----	-----	0	0	1	1
Vetor	-----	-----	0	0	1	1

Tabela 5 - Evolução da tabela de desempenho relativa aos tipos de caso de teste

tag	Pré 1ª Avaliação		Pós 1ª Avaliação		Pós 2ª Avaliação	
	total	errors	total	errors	total	errors
Caso Geral	-----	-----	2	0	4	0
Caso Limite	-----	-----	0	0	1	1
Caso Nulo	-----	-----	2	0	4	1
Caso Negativo	-----	-----	2	0	2	0
Igualdade	-----	-----	0	0	0	0
Tipo de Entrada	-----	-----	1	0	1	0

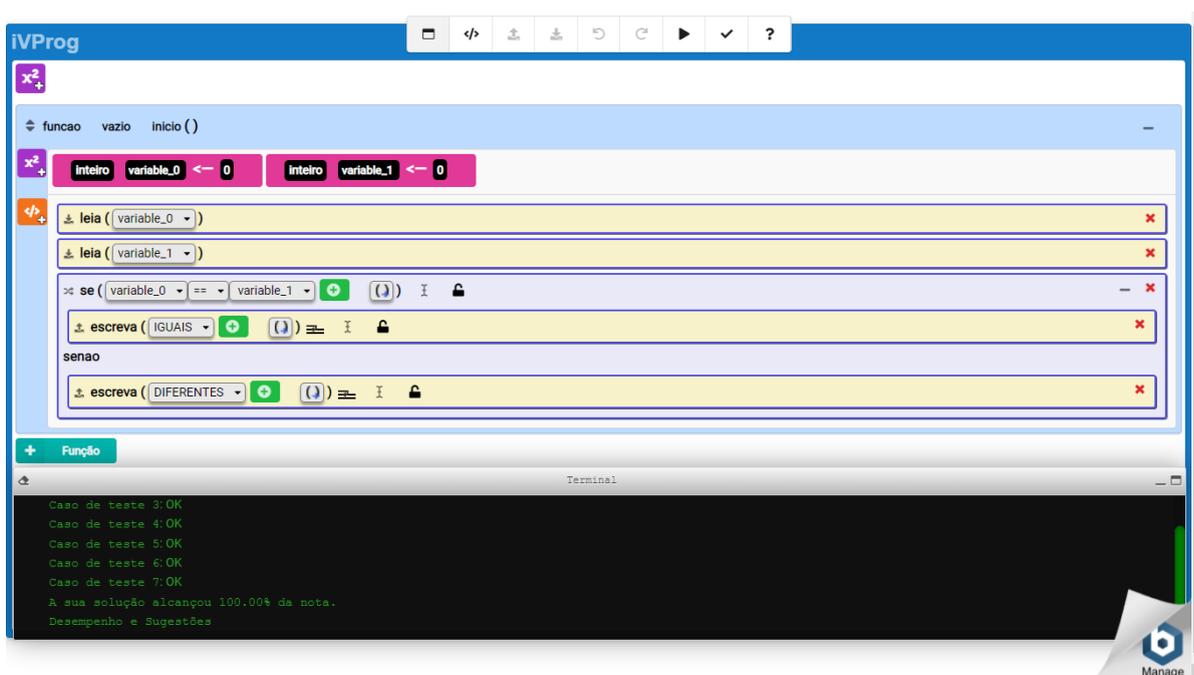
Tabela 6 - Evolução da tabela de desempenho relativa aos últimos X erros ocorridos

	Pré 1ª Avaliação	Pós 1ª Avaliação	Pós 2ª Avaliação
lastTypeErrors	-----	[]	[Condição, Laço, Vetor]
lastTagErrors	-----	[]	[Caso Limite, Caso Nulo]

### 7.3 TESTAGEM DO RF3

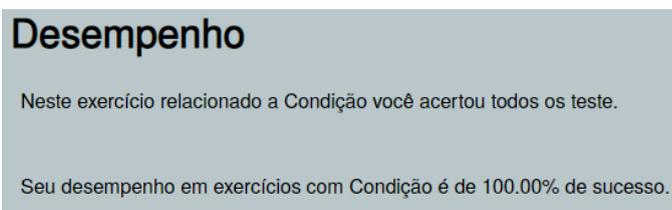
Para verificar se o Requisito Funcional RF3, “o sistema deve permitir que os alunos possam visualizar informações do seu desempenho relacionadas ao resultado da avaliação automática do exercício”, foi recepcionado, foram realizadas avaliações automáticas de todos os exercícios desenvolvidos três vezes: uma vez para um exercício completamente correto; uma vez para um exercício parcialmente correto; e uma vez para um exercício completamente incorreto. O intuito das avaliações é verificar se as informações de desempenho são exibidas corretamente de acordo com o resultado e se o desempenho é atualizado de forma correta.

Figura 30 - Execução da avaliação automática de um exercício



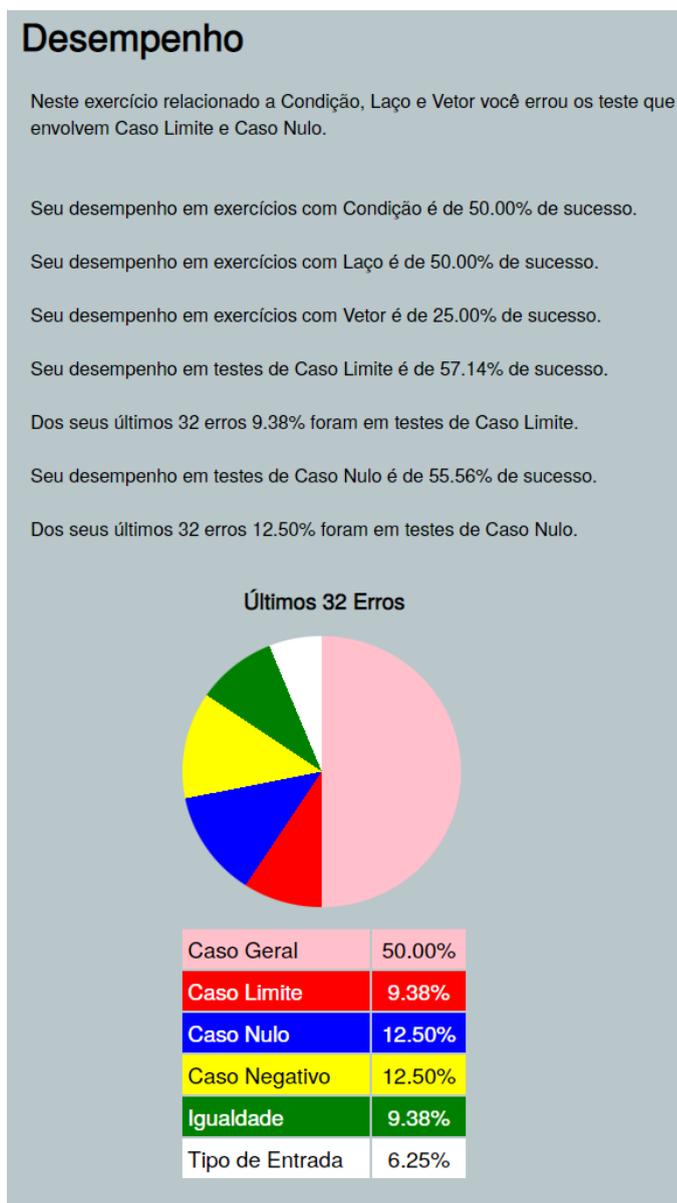
Fonte: Execução da versão deste projeto do iVProg

Figura 31 - Informações de desempenho de um exercício completamente correto



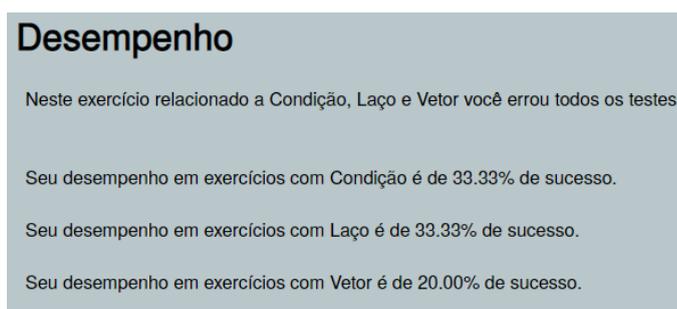
Fonte: Execução da versão deste projeto do iVProg

Figura 32 - Informações de desempenho de um exercício parcialmente correto



Fonte: Execução da versão deste projeto do iVProg

Figura 33 - Informações de desempenho de um exercício completamente incorreto



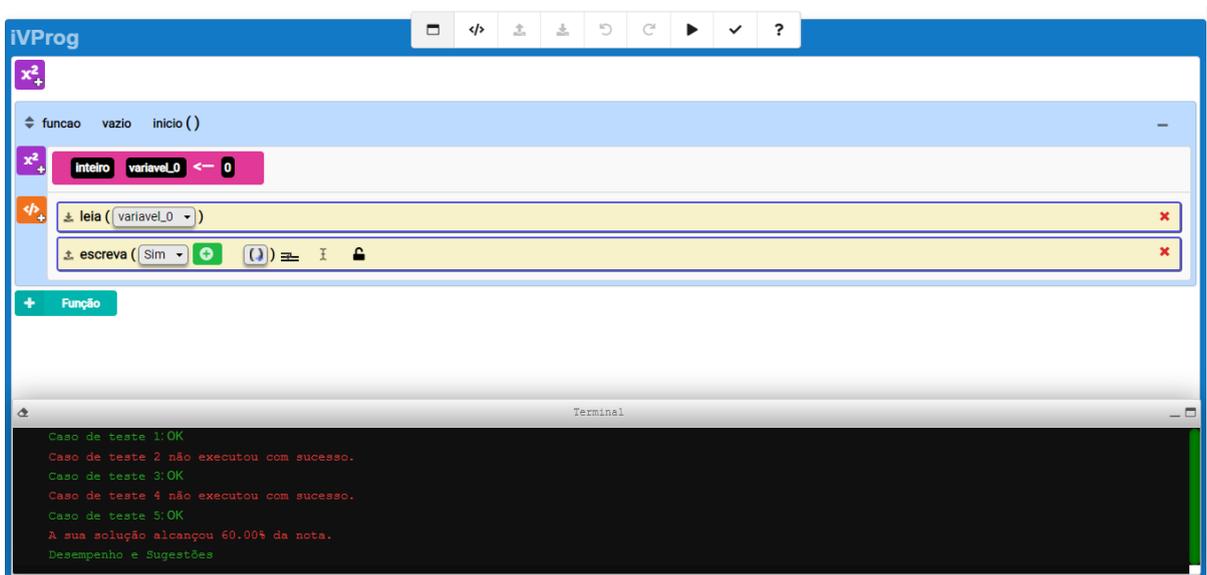
Fonte: Execução da versão deste projeto do iVProg

Ao longo da testagem as informações de desempenho foram exibidas corretamente de acordo com o resultado e o desempenho foi atualizado de forma correta em todas as avaliações automáticas realizadas.

#### 7.4 TESTAGEM DO RF4

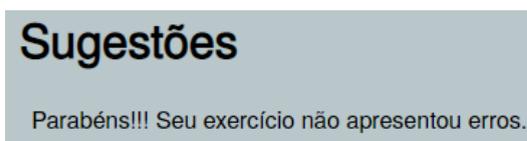
Para verificar se o Requisito Funcional RF4, “o sistema deve permitir que os alunos possam visualizar sugestões baseadas no seu desempenho relacionadas ao resultado da avaliação automática do exercício”, foi recepcionado, foram realizadas avaliações automáticas de todos os exercícios desenvolvidos três vezes: uma vez para um exercício completamente correto; uma vez para um exercício parcialmente correto; e uma vez para um exercício completamente incorreto. A execução das avaliações dessa forma tem como objetivo verificar se as sugestões são exibidas corretamente de acordo com o resultado.

Figura 34 - Execução da avaliação automática de um exercício



Fonte: Execução da versão deste projeto do iVProg

Figura 35 - Sugestões para um exercício completamente correto



Fonte: Execução da versão deste projeto do iVProg

Figura 36 - Sugestões para um exercício parcialmente correto

## Sugestões

Levantar quais são os Casos Limites do problema a ser resolvido é fundamental para se avaliar o funcionamento adequado do programa desenvolvido, pois ao se testá-los podem ser encontrados erros que não são observados nos Casos Gerais.

Os Casos Limites ocorrem nos valores que se encontram no limite da mudança de comportamento do programa. Por exemplo, em um programa que retorna se um número inteiro é maior que 5, os Casos Limites seriam os valores 5 e 6.

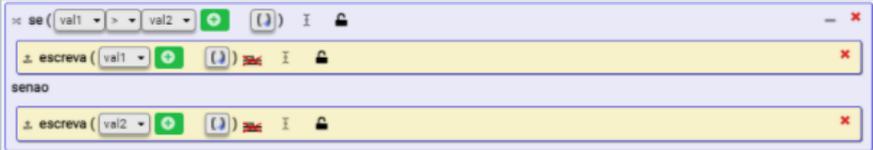
<https://edisciplinas.usp.br/mod/assign/view.php?id=CasoLimite>

É sempre importante testar o funcionamento adequado do programa desenvolvido em casos de entrada nula (Caso Nulo), pois o zero pode ter um impacto muito grande em diferentes cálculos, sempre deve se tomar cuidado com possíveis multiplicações e divisões por zero.

O Caso Nulo ocorre quando a entrada do programa é nula. Por exemplo, em um programa que trabalha com números, um Caso Nulo ocorre quando uma das entradas é zero, já em um programa que trabalha com texto, um Caso Nulo ocorre quando a entrada é um texto vazio.

<https://edisciplinas.usp.br/mod/assign/view.php?id=CasoNulo>

O uso de Condições é muito importante na programação, pois permite que se construa programas com comportamentos diferentes para situações diferentes, permitindo uma grande aplicabilidade aos códigos.



Erros em exercícios de Condição estão geralmente relacionados ao limite da condição, ou seja, nos valores adjacentes aos valores que satisfazem a condição.

<https://edisciplinas.usp.br/mod/assign/view.php?id=Condicao>

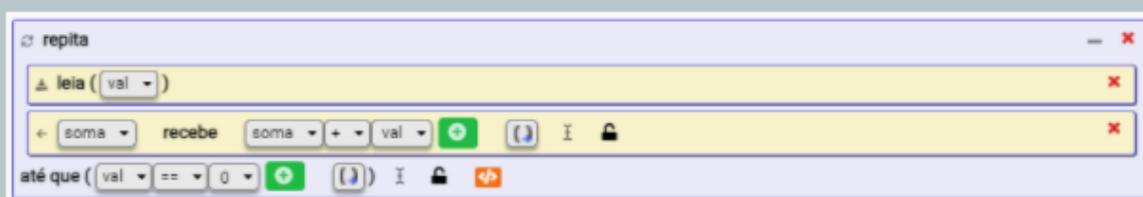
Fonte: Execução da versão deste projeto do iVProg

Figura 37 - Sugestões para um exercício completamente incorreto

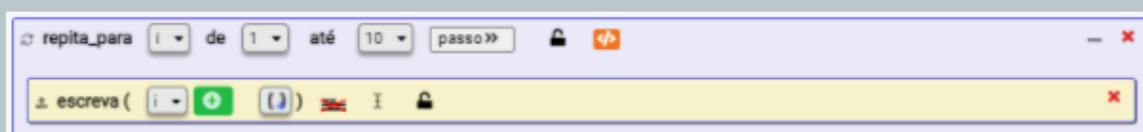
## Sugestões

Sempre é importante testar o programa desenvolvido antes de aplicá-lo em um ambiente real, lembre-se de testar o programa antes de submetê-lo à avaliação automática.

O conceito de Laço tem muita importância para a programação, pois permite a repetição de uma tarefa até que uma condição de parada seja satisfeita, eliminando a necessidade de ter que ficar escrevendo diversas vezes o código da tarefa.



```
repita
└─ leia ( val )
└─ soma recebe soma + val
└─ até que ( val == 0 )
```



```
repita para i de 1 até 10 passo 1
└─ escreva ( i )
```



```
repita enquanto ( variavel_2 < 10 )
└─ variavel_2 recebe variavel_2 + 1
└─ escreva ( variavel_2 )
```

Erros em exercícios de Laços geralmente estão relacionados à condição de parada, à atribuição da variável da condição de parada antes do laço e à manipulação da variável da condição de parada depois do laço.

<https://edisciplinas.usp.br/mod/assign/view.php?id=Lacos>

Fonte: Execução da versão deste projeto do iVProg

Ao longo da testagem as sugestões foram exibidas corretamente de acordo com o resultado em todas as avaliações automáticas realizadas.

## 8. CONSIDERAÇÕES FINAIS

### 8.1 CONCLUSÕES DO PROJETO DE FORMATURA

O projeto final conseguiu alcançar os objetivos e satisfazer os requisitos de sistema que foram definidos na elaboração do projeto, apresentando as funcionalidades propostas nos formatos planejados. Foi desenvolvido um assistente pedagógico para o sistema educacional iVProg em integração com o ambiente Moodle, que permite colaborar com o aprendizado do aluno. O assistente pedagógico desenvolvido apresenta informações sobre o desempenho do aluno, fornecendo sugestões pré determinadas de como melhorar nos pontos fracos.

No entanto, acreditava-se que seriam alcançados objetivos extras, como funcionalidades adicionais, estudo mais aprofundado para definir temas de exercício e tipos de caso de teste, otimização da comunicação entre o assistente pedagógico e o iTarefa que viabilizasse o acesso e armazenamento de dados no banco de dados do Moodle. Esses objetivos extras são detalhados nas perspectivas de continuidade.

O primeiro empecilho encontrado foi em relação à definição de temas de exercício e tipos de caso de teste, pois os dados disponíveis sobre os erros cometidos por alunos no iVProg são escassos. Com isso foi possível definir apenas temas de exercício e tipos de casos de teste aplicáveis a problemas simples de programação, o que foi suficiente já que esse é o caso de uso atual do iVProg.

Uma outra dificuldade, porém em menor grau, foi o aprendizado sobre as tecnologias utilizadas. Havia sido estimada uma curva de aprendizado menor. No entanto, a falta de familiaridade com a maioria das tecnologias utilizadas estendeu a curva de aprendizado além do estimado.

A maior dificuldade encontrada foi o entendimento do código fonte do iVProg e do iTarefa. Ambas as aplicações possuem diversos arquivos e diversas funções que realizam uma grande variedade de tarefas, motivo que gerou grande dificuldade para entendê-las em sua totalidade, bem como os seus usos.

## 8.2 CONTRIBUIÇÕES

A principal contribuição deste trabalho consistiu na adição de um assistente pedagógico ao iVProg. Ele colabora com a aprendizagem do aluno no contexto da avaliação automática de um exercício, exibindo informações sobre o desempenho e sugestões pré determinadas de como melhorar nos pontos fracos.

A existência de um assistente pedagógico no iVProg com as funções básicas necessárias e que faz a comunicação com o iTarefa que viabiliza o acesso e armazenamento de dados no banco de dados do Moodle, facilita o aprimoramento de funcionalidades já existentes e a adição de novas funcionalidades. Assim a adição do assistente pode servir de base para a construção de outros agentes pedagógicos com os quais ele possa interagir.

## 8.3 PERSPECTIVAS DE CONTINUIDADE

Uma forma de dar seguimento a esse projeto é realizar um estudo mais aprofundado sobre temas e tipos de caso de teste de exercícios de programação. Tal estudo pode ser conduzido acompanhando o uso do assistente pedagógico em cursos ministrados; analisando o seu impacto no aprendizado e na quantidade de erros cometidos; colhendo opiniões sobre os temas de exercício e tipos de caso de teste; estudando a fundo as aplicações dos conceitos de programação; dentre outras opções.

Outra maneira de dar continuidade a esse projeto é focar no desenvolvimento de uma otimização na comunicação com o iTarefa. Nesse caso o desenvolvimento da maior parte do projeto seria no iTarefa, aprimorando a segurança da comunicação feita com o Moodle.

Outra perspectiva de continuidade desse projeto se dá através do desenvolvimento de novas funcionalidades para o assistente pedagógico. Tais funcionalidades podem ser elaboradas a partir do acompanhamento do uso do assistente pedagógico em cursos ministrados.

Finalmente, também é possível dar seguimento ao projeto desenvolvendo outros agentes pedagógicos para o iVProg. Eles podem ser integrados de uma forma que resulte em um Sistema Tutor Inteligente.

## REFERÊNCIAS

- [1] RAZA, S. A.; QAZI, W.; KHAN, K.A.; SALAM, J. Social Isolation and Acceptance of the Learning Management System (LMS) in the time of COVID-19 Pandemic: An Expansion of the UTAUT Model. **Journal of Educational Computing Research**, v. 59, n. 2, p. 183-208, 2021. Disponível em: <[https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7509242/pdf/10.1177\\_0735633120960421.pdf](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7509242/pdf/10.1177_0735633120960421.pdf)>. Acesso em: 15 jan. 2022.
- [2] UNESCO. **Report on UNESCO's Response to the COVID-19 Pandemic and on the Impact of the Pandemic on the Implementation of the Programme and Budget for 2020-2021 (40 C/5)**. Paris: UNESCO Executive Board 210th Session, 2020. 34 p. (210 EX/17 + Corr). Disponível em: <<https://unesdoc.unesco.org/ark:/48223/pf0000374540>>. Acesso em: 15 jan. 2022.
- [3] PRAT, J.; LLORENS, A.; SALVADOR, F.; ALIER, M.; AMO, D. A Methodology to Study the University's Online Teaching Activity from Virtual Platform Indicators: The Effect of the Covid-19 Pandemic at Universitat Politècnica de Catalunya. **Sustainability**, v. 13, n. 9: 5177, 2021. Disponível em: <<https://www.mdpi.com/2071-1050/13/9/5177/htm>>. Acesso em: 15 jan. 2022.
- [4] ALZHRANI, L.; SETH, K. P. Factors influencing students satisfaction with continuous use of learning management systems during the COVID-19 pandemic: An empirical study. **Education and Information Technologies**, v. 26, n. 6, p. 6787-6805, 2021. Disponível em: <<https://link.springer.com/content/pdf/10.1007/s10639-021-10492-5.pdf>>. Acesso em: 15 jan. 2022.
- [5] DINDAR, M.; SUORSA, A.; HERMES, J.; KARPPINEN, P; NÄYKKI, P. Comparing technology acceptance of K-12 teachers with and without prior experience of learning management systems: A Covid-19 pandemic study. **Journal of Computer Assisted Learning**, v. 37, n. 6, p. 1553-1565, 2021. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8447015/pdf/JCAL-9999-0.pdf>>. Acesso em: 15 jan. 2022.
- [6] GIRAFFA, L. M. M. **Uma Arquitetura de Tutor Utilizando Estados Mentais**. 1999. 177 p. Tese (Doutorado em Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, Rio Grande do Sul, 1999. Disponível em: <<https://www.lume.ufrgs.br/bitstream/handle/10183/17620/000269142.pdf>>. Acesso em: 15 jan. 2022.
- [7] FINCH, D. K.; EDWARDS, S. H. Using a Pedagogical Agent to Support Students Learning to Program. In: ASEE VIRTUAL ANNUAL CONFERENCE CONTENT ACCESS, 2020. **Computers in Education Division Technical Session 9: Pedagogical Tools**. ASEE'S Virtual Conference, id. 28735. Disponível em: <<https://peer.asee.org/using-a-pedagogical-agent-to-support-students-learning-to-program>>. Acesso em: 15 mai. 2022.

- [8] BRANDÃO, A. A. F.; BRANDÃO, L. O.; SOUZA, L. M.; FELIX, I.; FERREIRA, B. iVProg: Programação Interativa Visual e Textual na Internet. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 2019, Brasília. **Anais dos Workshops do VIII Congresso Brasileiro de Informática na Educação**. Brasília: Sociedade Brasileira de Computação, 2019. p. 1164-1171. Disponível em: <[https://www.researchgate.net/publication/337528860\\_iVProg\\_Programacao\\_Interativa\\_Visual\\_e\\_Textual\\_na\\_Internet](https://www.researchgate.net/publication/337528860_iVProg_Programacao_Interativa_Visual_e_Textual_na_Internet)>. Acesso em: 15 jan. 2022.
- [9] CANNING, J.; PELLAND, D.; SLIGER, S. Visual Programming in a X Windows Workstation Environment. In: SYMPOSIUM ON SMALL SYSTEMS, 1991, Toronto. **Proceedings of the 1991 ACM SIGSMALL/PC Symposium on Small Systems**. Nova York: Association for Computing Machinery, 1991, p. 4-10. Disponível em: <<https://dl.acm.org/doi/10.1145/111048.111049>>. Acesso em: 15 jan. 2022.
- [10] IVPROG. **Sobre o iVProg**. Universidade de São Paulo. Disponível em: <[https://www.usp.br/line/ivprog/sobre\\_ivprog.html](https://www.usp.br/line/ivprog/sobre_ivprog.html)>. Acesso em: 15 jan. 2022.
- [11] BRITAIN, S.; LIBER, O. **A Framework for Pedagogical Evaluation of Virtual Learning Environments**. Bangor: Joint Information Systems Committee (JISC) Technology Applications Programme, 1999. 44 p. (Report 41). Disponível em: <<https://files.eric.ed.gov/fulltext/ED443394.pdf>>. Acesso em: 16 jan. 2022.
- [12] ELLIS, R. K. **A Field Guide to Learning Management Systems**. Alexandria: American Society for Training & Development (ASTD) Learning Circuits, 2009. 8 p. E-Learning Document. Disponível em: <<https://www.academia.edu/5312172/Www>>. Acesso em: 16 jan. 2022.
- [13] DISTEFANO, A.; RUDESTAM, K. E.; SILVERMAN, R. J. **Encyclopedia of Distributed Learning**. 1. ed. Thousand Oaks: SAGE Publications Inc., 2003. 576 p. Disponível em: <<https://sk.sagepub.com/reference/distributedlearning/n99.xml>>. Acesso em: 16 jan. 2022.
- [14] WATSON, W. R. An Argument for Clarity: What are Learning Management Systems, What are They Not, and What Should They Become?. **TechTrends**, v. 51, n. 2, p. 28-34, 2007. Disponível em: <<https://hal.archives-ouvertes.fr/file/index/docid/692067/filename/Watson-2007.pdf>>. Acesso em: 16 jan. 2022.
- [15] PINNER, R. **VLE or LMS?**. UniLiterate. Disponível em: <<https://uniliterate.com/2010/10/vle-or-lms/#.Yi33eXrMLcc>>. Acesso em: 16 jan. 2022.
- [16] EDTICK. **What is the Difference Between a Learning Management System (LMS) and a Virtual Learning Environment (VLE)?**. Disponível em: <<https://www.edtick.com/en/news/what-are-the-differences-between-lms-and-vle>>. Acesso em: 16 jan. 2022.
- [17] NADAV, D. **The Difference Between LMS and VLE**. Course For Me. Disponível em: <<https://courseforme.com/difference-between-lms-and-vle/>>. Acesso em: 16 jan. 2022.

- [18] MOODLE. **About Moodle**. Disponível em: <[https://docs.moodle.org/311/en/About\\_Moodle](https://docs.moodle.org/311/en/About_Moodle)>. Acesso em: 22 jan. 2022.
- [19] MOODLE. **Philosophy**. Disponível em: <<https://docs.moodle.org/311/en/Philosophy>>. Acesso em: 22 jan. 2022.
- [20] RODRIGUES, P. A. **iTarefa**: Componente Moodle para Incorporar Módulos de Aprendizagem Interativa em Cursos Web. 2011. 110 p. Dissertação (Mestrado em Ciência da Computação) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2011. Disponível em: <<https://www.teses.usp.br/teses/disponiveis/45/45134/tde-11042011-095825/publico/iAssignMoodle.pdf>>. Acesso em: 22 jan. 2022.
- [21] IASSIGN. **Repositório do iAssign**. Laboratório de Informática na Educação. Disponível em: <<http://200.144.254.107/git/LInE/iassign>>. Acesso em: 22 jan. 2022.
- [22] PSOTKA, J.; MASSEY, L. D.; MUTTER, S.A. **Intelligent Tutoring Systems: Lessons Learned**. 1. ed. Hove: Psychology Press, 1988. 576 p.
- [23] FREEDMAN, R. What is an intelligent tutoring system?. **Intelligence**, v. 11, n. 3, p. 15-16, 2000. Disponível em: <<http://faculty.cs.niu.edu/~freedman/papers/link2000.pdf>>. Acesso em: 23 jan. 2022.
- [24] JONASSEN, D.H.; WANG, S. The Physics Tutor: Integrating Hypertext and Expert Systems. **Journal of Educational Technology Systems**, v. 22, n. 1, p. 19-28, 1993. Disponível em: <<https://journals.sagepub.com/doi/10.2190/7L7F-57H5-APYU-AAPA>>. Acesso em: 23 jan. 2022.
- [25] URRETAVIZCAYA, L. M. Sistemas Inteligentes em el Ámbito de la Educación. **Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial**, v. 5, n. 12, p. 5-12, 2001. Disponível em: <<https://www.redalyc.org/pdf/925/92551202.pdf>>. Acesso em: 23 jan. 2022.
- [26] JOST, B.; KETTERL, M.; BUDDE, R.; LEIMBACH, T. Graphical Programming Environments for Educational Robots: Open Roberta - Yet Another One?. In: 2014 IEEE International Symposium on Multimedia, 2014. **Proceedings - 2014 IEEE International Symposium on Multimedia**. Washington: IEEE Computer Society, 2014, p. 381-386. Disponível em: <[https://www.academia.edu/29368636/Graphical\\_Programming\\_Environments\\_for\\_Educational\\_Robots\\_Open\\_Roberta\\_Yet\\_Another\\_One](https://www.academia.edu/29368636/Graphical_Programming_Environments_for_Educational_Robots_Open_Roberta_Yet_Another_One)>. Acesso em: 23 jan. 2022.
- [27] REPENNING, A. Moving Beyond Syntax: Lessons from 20 Years of Blocks Programming in AgentSheets. **Journal of Visual Languages and Sentient Systems**, v. 3, p. 68-91, 2017. Disponível em: <[http://ksiresearchorg.ipage.com/vlss/journal/VLSS2017/vlss17paper\\_10.pdf](http://ksiresearchorg.ipage.com/vlss/journal/VLSS2017/vlss17paper_10.pdf)>. Acesso em: 29 jan. 2022.
- [28] BURNETT, M. M. Visual Programming. In: WEBSTER, J. G. **Wiley Encyclopedia of Electrical and Electronics Engineering**. 1. ed. Nova York: John Wiley & Sons Inc., 1999.

- [29] SCRATCH WIKI. **Scratch**. Disponível em: <<https://en.scratch-wiki.info/wiki/Scratch>>. Acesso em: 29 jan. 2022.
- [30] SCRATCH WIKI. **Getting Started with Scratch**. Disponível em: <[https://en.scratch-wiki.info/wiki/Getting\\_Started\\_with\\_Scratch](https://en.scratch-wiki.info/wiki/Getting_Started_with_Scratch)>. Acesso em: 29 jan. 2022.
- [31] ALICE. **Alice**. Disponível em: <<http://www.alice.org/>>. Acesso em: 29 jan. 2022.
- [32] ALICE. **Alice 3**. Disponível em: <<https://www.alice.org/get-alice/alice-3/>>. Acesso em: 29 jan. 2022.
- [33] ALICE. **About Alice**. Disponível em: <<https://www.alice.org/about/>>. Acesso em: 29 jan. 2022.
- [34] JAVASCRIPT. **Padrão JavaScript**. ECMA International. Disponível em: <<https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>>. Acesso em: 5 fev. 2022.
- [35] JAVASCRIPT. **Tutoriais de JavaScript**. MDN Web Docs. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 5 fev. 2022.
- [36] W3TECHS. **Usage Statistics of JavaScript as Client-Side Programming Language on Websites**. W<sup>3</sup>Techs Web Technology Surveys. Disponível em: <<https://w3techs.com/technologies/details/cp-javascript/>>. Acesso em: 5 fev. 2022.
- [37] PHP. **PHP**. Disponível em: <<https://www.php.net/>>. Acesso em: 5 fev. 2022.
- [38] PHP. **Manual do PHP**. Disponível em: <[https://www.php.net/manual/pt\\_BR/](https://www.php.net/manual/pt_BR/)>. Acesso em: 5 fev. 2022.
- [39] PHP. **O Que é o PHP?**. Manual do PHP. Disponível em: <[https://www.php.net/manual/pt\\_BR/intro-what-is.php](https://www.php.net/manual/pt_BR/intro-what-is.php)>. Acesso em: 5 fev. 2022.
- [40] PHP. **O Que o PHP Pode Fazer?**. Manual do PHP. Disponível em: <[https://www.php.net/manual/pt\\_BR/intro-whatcando.php](https://www.php.net/manual/pt_BR/intro-whatcando.php)>. Acesso em: 5 fev. 2022.
- [41] IVPROG. **Repositório do iVProg**. Laboratório de Informática na Educação. Disponível em: <<http://200.144.254.107/git/LInE/iVProg>>. Acesso em: 5 fev. 2022.

## APÊNDICE A - EXERCÍCIOS DE IVPROG DESENVOLVIDOS

### A.1 - OLA MUNDO!

Esse é um exercício inicial típico de programação, escrever um "Ola mundo!". Para isso você deverá apenas inserir um comando de impressão (saída de dados: opção escrita de dados -> escreva), clicar em sua mensagem Selecione, escolher a opção Valor e digitar a frase: Ola mundo! (sem acento e com exclamação ao final).

```
{
  "exercisetype" : ["Leitura e Escrita" ],
  "testcases" : [
    { "input": [],
      "output": ["Ola mundo!"],
      "tag": ["Caso Geral"]
    }
  ],
  "settings_programming_type":
  [{"name": "programming_type", "value": "visual"}],
  "settings_data_types":
  [{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
  "value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "boolean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
  ],
  "settings_commands":
  [{"name": "commands_read", "value": "on"}, {"name": "commands_write", "value": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "commands_attribution", "value": "on"}, {"name": "commands_functioncall", "value": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"}, {"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch", "value": "on"}],
  "settings_functions":
  [{"name": "functions_creation", "value": "on"}, {"name": "functions_move", "value": "on"}],
  "settings_filter":
  [] }
::algorithm::{"functions": [{"type": "function", "name": "main", "return_type": "void", "return_dimensions": 0, "parameters_list": [], "is_main":
```

```

true,"is_hidden":false,"variables_list":[],"function_comment":{"type":
"comment","comment_text":"This is the main
funcion"},"commands":[{"type":"writer","content":[{"type":"var_valu
e","variable_and_value":7,"content":"Ola
mundo!","row":null,"column":null,"include_constant":true,"dimension
s":0}], "newline":true}]],"globals":[]}

```

## A.2 - LER UM INTEIRO E IMPRIMI-LO

Nesse exercício você deverá fazer um programa para ler um valor (inteiro) e imprimir esse valor. Para isso você deve criar uma variável do tipo inteiro, usar um comando para ler (entrada de dados) um valor (a ser digitado pelo usuário) e registrá-lo na variável e, ao final, seu programa deve imprimir (saída de dados) o valor digitado pelo usuário.

```

{
"exercisetype" : ["Leitura e Escrita" ],
"testcases" : [
{
"input": ["-1726"],
"output": ["-1726"],
"tag": ["Caso Negativo"]
},
{
"input": ["-4064"],
"output": ["-4064"],
"tag": ["Caso Negativo"]
},
{
"input": ["5588"],
"output": ["5588"],
"tag": ["Caso Geral"]
},
{
"input": ["9320"],
"output": ["9320"],
"tag": ["Caso Geral"]
},
{

```

```

"input": ["0"],
"output": ["0"],
"tag": ["Caso Nulo"]
},
{
"input": ["-5188"],
"output": ["-5188"],
"tag": ["Caso Negativo"]
},
{
"input": ["2877"],
"output": ["2877"],
"tag": ["Caso Geral"]
},
{
"input": ["7980"],
"output": ["7980"],
"tag": ["Caso Geral"]
},
{
"input": ["7547"],
"output": ["7547"],
"tag": ["Caso Geral"]
},
{
"input": ["-6293"],
"output": ["-6293"],
"tag": ["Caso Negativo"]
},
{
"input": ["2.5"],
"output": ["2"],
"tag": ["Tipo de Entrada"]
}
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "boolean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}

```

```

],
"settings_commands":
[{"name":"commands_read","value":"on"},{"name":"commands_write","value":"on"},{"name":"commands_comment","value":"on"},{"name":"commands_attribution","value":"on"},{"name":"commands_functioncall","value":"on"},{"name":"commands_iftrue","value":"on"},{"name":"commands_repeatNtimes","value":"on"},{"name":"commands_while","value":"on"},{"name":"commands_dowhile","value":"on"},{"name":"commands_switch","value":"on"}],
"settings_functions":
[{"name":"functions_creation","value":"on"},{"name":"functions_move","value":"on"}],
"settings_filter":
[] }
::algorithm:{"functions":[{"type":"function","name":"main","return_type":"void","return_dimensions":0,"parameters_list":[],"is_main":true,"is_hidden":false,"variables_list":[{"type":"integer","name":"variable_0","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0}], "function_comment":{"type":"comment","comment_text":"This is the main function"},"commands":[{"type":"reader","variable_value_menu":{"type":"var_value","variable_and_value":1,"content":{"type":"integer","name":"variable_0","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":false,"dimensions":0}},{"type":"writer","content":{"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"variable_0","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}},{"type":"newline"}]}], "globals":[]}

```

### A.3 - CONSUMO MÉDIO

Durante uma viagem de carro, Barrichello percebeu que o consumo médio apontado pelo sistema do carro não fazia sentido, pois apresentava valores muito fora dos padrões. Diante desse problema, Barrichello lhe procurou para que fosse desenvolvido um novo sistema de cálculo do consumo médio do veículo. Sendo assim, a sua tarefa é elaborar um programa que receba como entradas a distância

total percorrida pelo automóvel e o total de combustível gasto. Em seguida, deverá calcular e apresentar a média de quilômetros percorridos a cada litro de combustível.

```
{
  "exercisetype" : ["Matemática" ],
  "testcases" : [
    {
      "input": ["180.6", "11.8"],
      "output": ["15.30508475"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["250.5", "12.9"],
      "output": ["19.41860465"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["345.8", "26.4"],
      "output": ["13.09848485"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["843.28", "116.4"],
      "output": ["7.24467354"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["600.89", "68.4"],
      "output": ["8.78494152"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["0", "12.4"],
      "output": ["0.0"],
      "tag": ["Caso Nulo"]
    },
    {
      "input": ["105.4", "105.4"],
      "output": ["1.0"],
      "tag": ["Igualdade"]
    }
  ]
}
```

```

],
"settings_programming_type":
[{"name":"programming_type","value":"visual"}],
"settings_data_types":
[{"name":"integer_data_type","value":"on"}, {"name":"real_data_type",
"value":"on"}, {"name":"text_data_type","value":"on"}, {"name":"bool
ean_data_type","value":"on"}, {"name":"void_data_type","value":"on"}
],
"settings_commands":
[{"name":"commands_read","value":"on"}, {"name":"commands_write","va
lue":"on"}, {"name":"commands_comment","value":"on"}, {"name":"comman
ds_attribution","value":"on"}, {"name":"commands_functioncall","valu
e":"on"}, {"name":"commands_iftrue","value":"on"}, {"name":"commands_
repeatNtimes","value":"on"}, {"name":"commands_while","value":"on"},
{"name":"commands_dowhile","value":"on"}, {"name":"commands_switch",
"value":"on"}],
"settings_functions":
[{"name":"functions_creation","value":"on"}, {"name":"functions_move
","value":"on"}],
"settings_filter":
[] }
::algorithm::{
"functions":[{"type":"function","name":"main","return
_type":"void","return_dimensions":0,"parameters_list":[],"is_main":
true,"is_hidden":false,"variables_list":[{"type":"real","name":"dis
tancia","value":0,"dimensions":0,"is_constant":false,"rows":0,"colu
mns":0}, {"type":"real","name":"combustivel","value":0,"dimensions":
0,"is_constant":false,"rows":0,"columns":0}], "function_comment":{"t
ype":"comment","comment_text":"This is the main
funcion"}, "commands":[{"type":"reader","variable_value_menu":{"type
":"var_value","variable_and_value":1,"content":{"type":"real","name
":"distancia","value":0,"dimensions":0,"is_constant":false,"rows":0
,"columns":0},"row":null,"column":null,"include_constant":false,"di
mensions":0}}, {"type":"reader","variable_value_menu":{"type":"var_v
alue","variable_and_value":1,"content":{"type":"real","name":"combu
stivel","value":0,"dimensions":0,"is_constant":false,"rows":0,"colu
mns":0},"row":null,"column":null,"include_constant":false,"dimensio
ns":0}}, {"type":"writer","content":[{"type":"var_value","variable_a
nd_value":7,"content":{"type":"real","name":"distancia","value":0,"
dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,
"column":null,"include_constant":true,"dimensions":0}, {"type_op":"e
xp_arithmetic","item":"division"}, {"type":"var_value","variable_and

```

```
_value":7,"content":{"type":"real","name":"combustivel","value":0,"
dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,
"column":null,"include_constant":true,"dimensions":0}], "newline":tr
ue}}}], "globals":[]}
```

#### A.4 - TROCA DE VALORES

Nesse exercício, você deverá implementar um algoritmo que receba dois valores inteiros e armazene-os, respectivamente, nas variáveis: a e b. Em seguida, utilizando apenas o comando de atribuição, troque o conteúdo das variáveis, fazendo com que o valor que está em a passe para b e vice-versa.

```
{
  "exercisetype" : ["Manipulação de Entrada" ],
  "testcases" : [
    {
      "input": ["-8", "5"],
      "output": ["5", "-8"],
      "tag": ["Caso Negativo"]
    },
    {
      "input": ["9", "4"],
      "output": ["4", "9"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["0", "8"],
      "output": ["8", "0"],
      "tag": ["Caso Nulo"]
    },
    {
      "input": ["-7", "16"],
      "output": ["16", "-7"],
      "tag": ["Caso Negativo"]
    },
    {
      "input": ["68", "-4"],
      "output": ["-4", "68"],
      "tag": ["Caso Negativo"]
    }
  ]
}
```

```

},
{
  "input": ["4.5", "1"],
  "output": ["1", "4"],
  "tag": ["Tipo de Entrada"]
}
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "boolean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":
[{"name": "commands_read", "value": "on"}, {"name": "commands_write", "value": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "commands_attribution", "value": "on"}, {"name": "commands_functioncall", "value": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"}, {"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch", "value": "on"}],
"settings_functions":
[{"name": "functions_creation", "value": "on"}, {"name": "functions_move", "value": "on"}],
"settings_filter":
[] }
::algorithm: {"functions": [{"type": "function", "name": "main", "return_type": "void", "return_dimensions": 0, "parameters_list": [], "is_main": true, "is_hidden": false, "variables_list": [{"type": "integer", "name": "variable_0", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, {"type": "integer", "name": "variable_1", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}], "function_comment": {"type": "comment", "comment_text": "This is the main function"}, "commands": [{"type": "reader", "variable_value_menu": {"type": "var_value", "variable_and_value": 1, "content": {"type": "integer", "name": "variable_0", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, "row": null, "column": null, "include_constant": false, "dimensions": 0}}, {"type": "reader", "variable_value_menu": {"type": "var_value", "variable_and_value": 1, "content": {"type": "integer", "name": "variable_1", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0

```

```
, "columns": 0}, "row": null, "column": null, "include_constant": false, "dimensions": 0}}, {"type": "writer", "content": [{"type": "var_value", "variable_and_value": 7, "content": {"type": "integer", "name": "variable_1", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, "row": null, "column": null, "include_constant": true, "dimensions": 0}], "newline": true}, {"type": "writer", "content": [{"type": "var_value", "variable_and_value": 7, "content": {"type": "integer", "name": "variable_0", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, "row": null, "column": null, "include_constant": true, "dimensions": 0}], "newline": true}]]], "globals": []}
```

## A.5 - ANTECESSOR E SUCESSOR

Nesse exercício, você deverá implementar um algoritmo que receba um único valor inteiro. Em seguida, o algoritmo deve calcular e informar o antecessor e o sucessor do número recebido na entrada.

```
{
  "exercisetype" : ["Matemática" ],
  "testcases" : [
    {
      "input": ["6"],
      "output": ["5", "7"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["68"],
      "output": ["67", "69"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["-5"],
      "output": ["-6", "-4"],
      "tag": ["Caso Negativo"]
    },
    {
      "input": ["0"],
      "output": ["-1", "1"],
      "tag": ["Caso Nulo"]
    }
  ]
}
```

```

},
{
  "input": ["685"],
  "output": ["684", "686"],
  "tag": ["Caso Geral"]
},
{
  "input": ["4.5"],
  "output": ["3", "5"],
  "tag": ["Tipo de Entrada"]
}
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "bool
ean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":
[{"name": "commands_read", "value": "on"}, {"name": "commands_write", "va
lue": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "comman
ds_attribution", "value": "on"}, {"name": "commands_functioncall", "valu
e": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_
repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"},
{"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch",
"value": "on"}],
"settings_functions":
[{"name": "functions_creation", "value": "on"}, {"name": "functions_move
", "value": "on"}],
"settings_filter":
[] }
::algorithm::{"functions":[{"type": "function", "name": "main", "return
_type": "void", "return_dimensions": 0, "parameters_list": [], "is_main":
true, "is_hidden": false, "variables_list": [{"type": "integer", "name": "
variable_0", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "
columns": 0}], "function_comment": {"type": "comment", "comment_text": "T
his is the main
function"}, "commands": [{"type": "reader", "variable_value_menu": {"type
": "var_value", "variable_and_value": 1, "content": {"type": "integer", "n
ame": "variable_0", "value": 0, "dimensions": 0, "is_constant": false, "row

```

```
s":0,"columns":0},"row":null,"column":null,"include_constant":false
,"dimensions":0}},{"type":"writer","content":[{"type":"var_value","
variable_and_value":7,"content":{"type":"integer","name":"variable_
0","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":
0},"row":null,"column":null,"include_constant":true,"dimensions":0}
,{"type_op":"exp_arithmetic","item":"minus"},"type":"var_value","v
variable_and_value":7,"content":"1","row":null,"column":null,"includ
e_constant":true,"dimensions":0}], "newline":true}, {"type":"writer",
"content":[{"type":"var_value","variable_and_value":7,"content":{"t
ype":"integer","name":"variable_0","value":0,"dimensions":0,"is_con
stant":false,"rows":0,"columns":0},"row":null,"column":null,"includ
e_constant":true,"dimensions":0}, {"type_op":"exp_arithmetic","item"
:"plus"},"type":"var_value","variable_and_value":7,"content":"1","
row":null,"column":null,"include_constant":true,"dimensions":0}], "n
ewline":true}}]], "globals":[]}]
```

## A.6 - IMPOSTO EM MARCIANÓPOLIS

A cidade de Marciánópolis (GO) decidiu que já estava na hora de atualizar o sistema de cálculo do IPTU (Imposto Predial e Territorial Urbano). Para isso, optou por simplificar a fórmula de cálculo: para cada metro quadrado de área construída, a prefeitura cobrará R\$ 1,18. E você está incumbido de ajudá-los nessa tarefa! Você deverá implementar um algoritmo que receba o total de metros quadrados de área construída. Em seguida, deverá calcular e apresentar o valor total que o contribuinte deverá pagar.

```
{
  "exercisetype" : ["Matemática" ],
  "testcases" : [
    {
      "input": ["150.3"],
      "output": ["177.35400000"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["84.4"],
      "output": ["99.59200000"],
      "tag": ["Caso Geral"]
    }
  ]
}
```

```

},
{
  "input": ["45.0"],
  "output": ["53.10000000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["254.6"],
  "output": ["300.42800000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["140.4"],
  "output": ["165.67200000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["114.3"],
  "output": ["134.87400000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["0"],
  "output": ["0.0"],
  "tag": ["Caso Nulo"]
}
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "boolean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":
[{"name": "commands_read", "value": "on"}, {"name": "commands_write", "value": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "commands_attribution", "value": "on"}, {"name": "commands_functioncall", "value": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"}, {"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch",

```

```

"value":"on"}],
"settings_functions":
[{"name":"functions_creation","value":"on"},{"name":"functions_move
","value":"on"}],
"settings_filter":
[] }
::algorithm:{"functions":[{"type":"function","name":"main","return
_type":"void","return_dimensions":0,"parameters_list":[],"is_main":
true,"is_hidden":false,"variables_list":[{"type":"real","name":"var
iable_0","value":0,"dimensions":0,"is_constant":false,"rows":0,"col
umns":0}], "function_comment":{"type":"comment","comment_text":"This
is the main
funcion"},"commands":[{"type":"reader","variable_value_menu":{"type
":"var_value","variable_and_value":1,"content":{"type":"real","name
":"variable_0","value":0,"dimensions":0,"is_constant":false,"rows":
0,"columns":0},"row":null,"column":null,"include_constant":false,"d
imensions":0}},{type":"writer","content":[{"type":"var_value","var
iable_and_value":7,"content":{"type":"real","name":"variable_0","va
lue":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"ro
w":null,"column":null,"include_constant":true,"dimensions":0}},{typ
e_op":"exp_arithmetic","item":"multiplication"}, {"type":"var_value"
,"variable_and_value":7,"content":"1.18","row":null,"column":null,"
include_constant":true,"dimensions":0}], "newline":true}]]], "globals
":[]}]

```

## A.7 - FOLHA DE PAGAMENTO

A empresa de confecção Indiana Jeans, decidiu incluir na remuneração mensal de seus vendedores, uma comissão de 15% sobre as vendas efetuadas naquele mês. Porém, essa bonificação não estava prevista em seu sistema antigo. E claro, você foi convidado a implementar essa nova funcionalidade! Para essa implementação, seu programa deverá receber o nome do vendedor, o seu salário fixo e o valor total de vendas efetuadas por ele no mês. Em seguida, calcule e imprima o resultado na seguinte ordem: nome do vendedor, salário fixo, comissão das vendas e o salário a receber (salário fixo + comissão).

```

{
"exercisetype" : ["Matemática","Lógica","Manipulação de Entrada" ],

```

```

"testcases" : [
{
  "input": ["Ana Clara", "1468.22", "10684.80"],
  "output": ["Ana Clara", "1468.22000000", "1602.72000000",
"3070.94000000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["Rita Oliveira", "1849.0", "14900.55"],
  "output": ["Rita Oliveira", "1849.0", "2235.08250000",
"4084.08250000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["Alberto Teixeira", "1345.8", "11930.66"],
  "output": ["Alberto Teixeira", "1345.80000000", "1789.59900000",
"3135.39900000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["Joaquina Costa", "1654.3", "9854.6"],
  "output": ["Joaquina Costa", "1654.30000000", "1478.19000000",
"3132.49000000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["Murilo Ribeiro", "1678.9", "11650.3"],
  "output": ["Murilo Ribeiro", "1678.90000000", "1747.54500000",
"3426.44500000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["Carlos Silva", "1650.5", "0"],
  "output": ["Carlos Silva", "1650.50000000", "0.0",
"1650.50000000"],
  "tag": ["Caso Nulo"]
},
{
  "input": ["André Germano", "1600.8", "1600.8"],
  "output": ["André Germano", "1600.80000000", "240.12000000",
"1840.92000000"],

```

```

    "tag": ["Igualdade"]
  }
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "bool
ean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":
[{"name": "commands_read", "value": "on"}, {"name": "commands_write", "va
lue": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "comman
ds_attribution", "value": "on"}, {"name": "commands_functioncall", "valu
e": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_
repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"},
{"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch",
"value": "on"}],
"settings_functions":
[{"name": "functions_creation", "value": "on"}, {"name": "functions_move
", "value": "on"}],
"settings_filter":
[] }
::algorithm::{"functions":[{"type": "function", "name": "main", "return
_type": "void", "return_dimensions": 0, "parameters_list": [], "is_main":
true, "is_hidden": false, "variables_list": [{"type": "text", "name": "nom
e", "value": "text", "dimensions": 0, "is_constant": false, "rows": 0, "colu
mns": 0}, {"type": "real", "name": "salario", "value": 0, "dimensions": 0, "i
s_constant": false, "rows": 0, "columns": 0}, {"type": "real", "name": "vend
as", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns"
: 0}], "function_comment": {"type": "comment", "comment_text": "This is
the main
funcion"}, "commands": [{"type": "reader", "variable_value_menu": {"type
": "var_value", "variable_and_value": 1, "content": {"type": "text", "name
": "nome", "value": "text", "dimensions": 0, "is_constant": false, "rows": 0
, "columns": 0}, "row": null, "column": null, "include_constant": false, "di
mensions": 0}}, {"type": "reader", "variable_value_menu": {"type": "var_v
alue", "variable_and_value": 1, "content": {"type": "real", "name": "salar
io", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns"
: 0}, "row": null, "column": null, "include_constant": false, "dimensions":
0}}, {"type": "reader", "variable_value_menu": {"type": "var_value", "var

```

```

iable_and_value":1,"content":{"type":"real","name":"vendas","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":false,"dimensions":0}},{"type":"writer","content":[{"type":"var_value","variable_and_value":7,"content":{"type":"text","name":"nome","value":"text","dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}], "newline":true}, {"type":"writer","content":[{"type":"var_value","variable_and_value":7,"content":{"type":"real","name":"salario","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}], "newline":true}, {"type":"writer","content":[{"type":"var_value","variable_and_value":7,"content":{"type":"real","name":"vendas","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}, {"type_op":"exp_arithmetic","item":"multiplication"}, {"type":"var_value","variable_and_value":7,"content":"0.15","row":null,"column":null,"include_constant":true,"dimensions":0}], "newline":true}, {"type":"writer","content":[{"type":"var_value","variable_and_value":7,"content":{"type":"real","name":"salario","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}, {"type_op":"exp_arithmetic","item":"plus"}, {"type":"var_value","variable_and_value":7,"content":{"type":"real","name":"vendas","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}, {"type_op":"exp_arithmetic","item":"multiplication"}, {"type":"var_value","variable_and_value":7,"content":"0.15","row":null,"column":null,"include_constant":true,"dimensions":0}], "newline":true}}], "globals": []}

```

## A.8 - TOTAL DE DIAS

Nesse exercício, você deverá implementar um algoritmo que calcule o total de dias de  $n$  meses. Para isso, seu programa deve receber um valor inteiro, representando o total de meses. Em seguida, deverá calcular e imprimir o resultado. Considere que todos os meses tenham 30 dias.

```

{
  "exercisetype" : ["Matemática" ],
  "testcases" : [

```

```
{
  "input": ["10"],
  "output": ["300"],
  "tag": ["Caso Geral"]
},
{
  "input": ["4"],
  "output": ["120"],
  "tag": ["Caso Geral"]
},
{
  "input": ["5"],
  "output": ["150"],
  "tag": ["Caso Geral"]
},
{
  "input": ["6"],
  "output": ["180"],
  "tag": ["Caso Geral"]
},
{
  "input": ["12"],
  "output": ["360"],
  "tag": ["Caso Geral"]
},
{
  "input": ["0"],
  "output": ["0"],
  "tag": ["Caso Nulo"]
},
{
  "input": ["4.5"],
  "output": ["120"],
  "tag": ["Tipo de Entrada"]
}
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "bool
```

```

ean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":
[{"name": "commands_read", "value": "on"}, {"name": "commands_write", "va
lue": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "comman
ds_attribution", "value": "on"}, {"name": "commands_functioncall", "valu
e": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_
repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"},
{"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch",
"value": "on"}],
"settings_functions":
[{"name": "functions_creation", "value": "on"}, {"name": "functions_move
", "value": "on"}],
"settings_filter":
[] }
::algorithm::{"functions":[{"type": "function", "name": "main", "return
_type": "void", "return_dimensions": 0, "parameters_list": [], "is_main":
true, "is_hidden": false, "variables_list": [{"type": "integer", "name": "
variable_0", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "
columns": 0}], "function_comment": {"type": "comment", "comment_text": "T
his is the main
funcion"}, "commands": [{"type": "reader", "variable_value_menu": {"type
": "var_value", "variable_and_value": 1, "content": {"type": "integer", "n
ame": "variable_0", "value": 0, "dimensions": 0, "is_constant": false, "row
s": 0, "columns": 0}, "row": null, "column": null, "include_constant": false
, "dimensions": 0}}, {"type": "writer", "content": [{"type": "var_value", "
variable_and_value": 7, "content": {"type": "integer", "name": "variable_
0", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns":
0}, "row": null, "column": null, "include_constant": true, "dimensions": 0}
, {"type_op": "exp_arithmetic", "item": "multiplication"}, {"type": "var_
value", "variable_and_value": 7, "content": "30", "row": null, "column": nu
ll, "include_constant": true, "dimensions": 0}], "newline": true}}], "glo
bals": []}

```

## A.9 - ÁREA DO CÍRCULO

Nesse exercício, você deverá implementar um algoritmo que calcule a área de um círculo. Para isso, seu programa deverá receber o raio, calcular  $\text{área} = \pi \cdot \text{raio}^2$  e imprimir o resultado. Considere o pi com duas casas decimais.

```
{
  "exercisetype" : ["Matemática" ],
  "testcases" : [
    {
      "input": ["15"],
      "output": ["706.50000000"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["6.8"],
      "output": ["145.19360000"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["9.1"],
      "output": ["260.02340000"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["4.6"],
      "output": ["66.44240000"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["17.2"],
      "output": ["928.93760000"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["0"],
      "output": ["0.0"],
      "tag": ["Caso Nulo"]
    }
  ],
  "settings_programming_type":
```

```

[{"name":"programming_type","value":"visual"}],
"settings_data_types":
[{"name":"integer_data_type","value":"on"}, {"name":"real_data_type",
,"value":"on"}, {"name":"text_data_type","value":"on"}, {"name":"bool
ean_data_type","value":"on"}, {"name":"void_data_type","value":"on"}
],
"settings_commands":
[{"name":"commands_read","value":"on"}, {"name":"commands_write","va
lue":"on"}, {"name":"commands_comment","value":"on"}, {"name":"comman
ds_attribution","value":"on"}, {"name":"commands_functioncall","valu
e":"on"}, {"name":"commands_iftrue","value":"on"}, {"name":"commands_
repeatNtimes","value":"on"}, {"name":"commands_while","value":"on"},
{"name":"commands_dowhile","value":"on"}, {"name":"commands_switch",
"value":"on"}],
"settings_functions":
[{"name":"functions_creation","value":"on"}, {"name":"functions_move
","value":"on"}],
"settings_filter":
[] }
::algorithm:{"functions":[{"type":"function","name":"main","return
_type":"void","return_dimensions":0,"parameters_list":[],"is_main":
true,"is_hidden":false,"variables_list":[{"type":"real","name":"var
iable_0","value":0,"dimensions":0,"is_constant":false,"rows":0,"col
umns":0}], "function_comment":{"type":"comment","comment_text":"This
is the main
funcion"},"commands":[{"type":"reader","variable_value_menu":{"type
":"var_value","variable_and_value":1,"content":{"type":"real","name
":"variable_0","value":0,"dimensions":0,"is_constant":false,"rows":
0,"columns":0},"row":null,"column":null,"include_constant":false,"d
imensions":0}},{type":"writer","content":[{"type":"var_value","var
iable_and_value":7,"content":{"type":"real","name":"variable_0","va
lue":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"ro
w":null,"column":null,"include_constant":true,"dimensions":0} {"typ
e_op":"exp_arithmetic","item":"multiplication"}, {"type":"var_value"
,"variable_and_value":7,"content":{"type":"real","name":"variable_0
","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0
},"row":null,"column":null,"include_constant":true,"dimensions":0},
{"type_op":"exp_arithmetic","item":"multiplication"}, {"type":"var_v
alue","variable_and_value":7,"content":"3.14","row":null,"column":n
ull,"include_constant":true,"dimensions":0}], "newline":true}}], "gl
obals":[]}]

```

## A.10 - DIVISÃO DA CONTA DO RESTAURANTE

Três amigos, Carlos, André e Felipe costumam sair para jantar todas às sextas após o expediente. O grupo sempre divide a conta, mas o quanto cada um deve pagar é calculado de um jeito diferente. Por acordo mútuo, Carlos e André nunca pagam centavos. Assim, para facilitar o processo, Felipe decidiu pedir a você para elaborar um programa que ao ler o valor total da conta, escreva o quanto cada um deve pagar respeitando a regra da divisão entre eles. Note que a soma das partes que cada um deve pagar não pode exceder nem ficar abaixo do total devido. Desta forma, seu programa deverá ler o total da conta (valor do tipo real) e calcular o quanto André, Carlos e Felipe devem pagar respectivamente.

```
{
  "exercisetype" : ["Lógica","Matemática" ],
  "testcases" : [
    {
      "input": ["325.89"],
      "output": ["108", "108", "109.89000000"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["1356.60"],
      "output": ["452", "452", "452.60000000"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["125.69"],
      "output": ["41", "41", "43.69000000"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["1.5"],
      "output": ["0", "0", "1.50000000"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["8756.98"],
```

```
"output": ["2918", "2918", "2920.98000000"],
"tag": ["Caso Geral"]
},
{
  "input": ["4578.63"],
  "output": ["1526", "1526", "1526.63000000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["12.56"],
  "output": ["4", "4", "4.56000000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["47.88"],
  "output": ["15", "15", "17.88000000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["99.99"],
  "output": ["33", "33", "33.99000000"],
  "tag": ["Caso Geral"]
},
{
  "input": ["5"],
  "output": ["1", "1", "3.0"],
  "tag": ["Caso Geral"]
},
{
  "input": ["0"],
  "output": ["0", "0", "0.0"],
  "tag": ["Caso Nulo"]
},
{
  "input": ["6"],
  "output": ["2", "2", "2.0"],
  "tag": ["Caso Limite"]
},
{
  "input": ["6.1"],
  "output": ["2", "2", "2.10000000"],
```

```

    "tag": ["Caso Limite"]
  },
  {
    "input": ["5.9"],
    "output": ["1", "1", "3.90000000"],
    "tag": ["Caso Limite"]
  }
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "boolean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":
[{"name": "commands_read", "value": "on"}, {"name": "commands_write", "value": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "commands_attribution", "value": "on"}, {"name": "commands_functioncall", "value": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"}, {"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch", "value": "on"}],
"settings_functions":
[{"name": "functions_creation", "value": "on"}, {"name": "functions_move", "value": "on"}],
"settings_filter":
[] }
::algorithm::{"functions":[{"type": "function", "name": "main", "return_type": "void", "return_dimensions": 0, "parameters_list": [], "is_main": true, "is_hidden": false, "variables_list": [{"type": "real", "name": "conta", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, {"type": "real", "name": "contadiv3", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, {"type": "integer", "name": "containt", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, {"type": "real", "name": "contafelipe", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}], "function_comment": {"type": "comment", "comment_text": "This is the main function"}, "commands": [{"type": "reader", "variable_value_menu": {"type": "var_value", "variable_and_value": 1, "content": {"type": "real", "name": "conta", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "co

```

```

lums":0},"row":null,"column":null,"include_constant":false,"dimensions":0}},{"type":"attribution","variable":{"type":"var_value","variable_and_value":1,"content":{"type":"real","name":"contadiv3","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":false,"dimensions":0},"expression":[{"type":"var_value","variable_and_value":7,"content":{"type":"real","name":"conta","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}, {"type_op":"exp_arithmetic","item":"division"}, {"type":"var_value","variable_and_value":7,"content":"3","row":null,"column":null,"include_constant":true,"dimensions":0}]}], {"type":"attribution","variable":{"type":"var_value","variable_and_value":1,"content":{"type":"integer","name":"containt","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":false,"dimensions":0},"expression":[{"type":"var_value","variable_and_value":7,"content":null,"row":null,"column":null,"include_constant":true,"dimensions":0,"function_called":{"type":"function","identifier":"$castInt","return_type":"integer","return_dimensions":0,"parameters_list":[{"type":"var_value","variable_and_value":7,"content":null,"row":null,"column":null,"include_constant":true,"dimensions":0}], "function_comment":null,"category":"$langLib"},"parameters_list":[{"type":"var_value","variable_and_value":7,"content":{"type":"real","name":"contadiv3","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}]}]}], {"type":"attribution","variable":{"type":"var_value","variable_and_value":1,"content":{"type":"real","name":"contafelipe","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":false,"dimensions":0},"expression":[{"type":"var_value","variable_and_value":7,"content":{"type":"real","name":"conta","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}, {"type_op":"exp_arithmetic","item":"minus"}, {"type":"var_value","variable_and_value":7,"content":"2","row":null,"column":null,"include_constant":true,"dimensions":0}, {"type_op":"exp_arithmetic","item":"multiplication"}, {"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"containt","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}]}], {"type":"writer","content":[{"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"containt","value":0,"dimensions":0,"is_constant":false

```

```
, "rows":0, "columns":0}, "row":null, "column":null, "include_constant":true, "dimensions":0}], "newline":true}, {"type":"writer", "content":[{"type":"var_value", "variable_and_value":7, "content":{"type":"integer", "name":"containt", "value":0, "dimensions":0, "is_constant":false, "rows":0, "columns":0}, "row":null, "column":null, "include_constant":true, "dimensions":0}], "newline":true}, {"type":"writer", "content":[{"type":"var_value", "variable_and_value":7, "content":{"type":"real", "name":"contafelipe", "value":0, "dimensions":0, "is_constant":false, "rows":0, "columns":0}, "row":null, "column":null, "include_constant":true, "dimensions":0}], "newline":true}]]], "globals":[]}
```

### A.11 - INVERTER NÚMERO INTEIRO DE TRÊS DÍGITOS

Você deve elaborar um programa que lê um número inteiro de 3 dígitos e imprime na ordem inversa.

```
{
  "exercisetype" : ["Manipulação de Entrada", "Lógica" ],
  "testcases" : [
    {
      "input": ["123"],
      "output": ["321"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["478"],
      "output": ["874"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["656"],
      "output": ["656"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["423"],
      "output": ["324"],
      "tag": ["Caso Geral"]
    },
    {

```

```

    "input": ["222"],
    "output": ["222"],
    "tag": ["Igualdade"]
  },
  {
    "input": ["000"],
    "output": ["0"],
    "tag": ["Caso Nulo"]
  },
  {
    "input": ["-524"],
    "output": ["-425"],
    "tag": ["Caso Negativo"]
  },
  {
    "input": ["527.8"],
    "output": ["725"],
    "tag": ["Tipo de Entrada"]
  }
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "bool_
ean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":
[{"name": "commands_read", "value": "on"}, {"name": "commands_write", "va
lue": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "comman
ds_attribution", "value": "on"}, {"name": "commands_functioncall", "valu
e": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_
repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"},
{"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch",
"value": "on"}],
"settings_functions":
[{"name": "functions_creation", "value": "on"}, {"name": "functions_move
", "value": "on"}],
"settings_filter":
[] }
::algorithm::{ "functions": [{"type": "function", "name": "main", "return

```

```

_type":"void","return_dimensions":0,"parameters_list":[],"is_main":
true,"is_hidden":false,"variables_list":[{"type":"integer","name":
"entrada","value":0,"dimensions":0,"is_constant":false,"rows":0,"col
umns":0},{type":"integer","name":"div100","value":0,"dimensions":0
,"is_constant":false,"rows":0,"columns":0},{type":"integer","name"
:"dezenas","value":0,"dimensions":0,"is_constant":false,"rows":0,"c
olumns":0},{type":"integer","name":"digitodezenas","value":0,"dime
nsions":0,"is_constant":false,"rows":0,"columns":0},{type":"intege
r","name":"unidades","value":0,"dimensions":0,"is_constant":false,"
rows":0,"columns":0}],function_comment":{"type":"comment","comment
_text":"This is the main
funcion"},"commands":[{"type":"reader","variable_value_menu":{"type
":"var_value","variable_and_value":1,"content":{"type":"integer","n
ame":"entrada","value":0,"dimensions":0,"is_constant":false,"rows":
0,"columns":0},"row":null,"column":null,"include_constant":false,"d
imensions":0}},{type":"attribution","variable":{"type":"var_value"
,"variable_and_value":1,"content":{"type":"integer","name":"div100"
,"value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0}
,"row":null,"column":null,"include_constant":false,"dimensions":0},
"expression":[{"type":"var_value","variable_and_value":7,"content":
{"type":"integer","name":"entrada","value":0,"dimensions":0,"is_con
stant":false,"rows":0,"columns":0},"row":null,"column":null,"includ
e_constant":true,"dimensions":0},{type_op":"exp_arithmetic","item":
"division"},{"type":"var_value","variable_and_value":7,"content":
"100","row":null,"column":null,"include_constant":true,"dimensions":
0}]}},{type":"attribution","variable":{"type":"var_value","variable
_and_value":1,"content":{"type":"integer","name":"dezenas","value":
0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":nu
ll,"column":null,"include_constant":false,"dimensions":0},"expressi
on":[{"type":"var_value","variable_and_value":7,"content":{"type":"
integer","name":"entrada","value":0,"dimensions":0,"is_constant":fa
lse,"rows":0,"columns":0},"row":null,"column":null,"include_constan
t":true,"dimensions":0},{type_op":"exp_arithmetic","item":"minus"}
,{"type":"var_value","variable_and_value":7,"content":"100","row":n
ull,"column":null,"include_constant":true,"dimensions":0},{type_op
":"exp_arithmetic","item":"multiplication"},{"type":"var_value","va
riable_and_value":7,"content":{"type":"integer","name":"div100","va
lue":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"ro
w":null,"column":null,"include_constant":true,"dimensions":0}]}},{t
ype":"attribution","variable":{"type":"var_value","variable_and_val
ue":1,"content":{"type":"integer","name":"digitodezenas","value":0,

```

```

"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null
,"column":null,"include_constant":false,"dimensions":0},"expression
":[{"type":"var_value","variable_and_value":7,"content":{"type":"in
teger","name":"dezenas","value":0,"dimensions":0,"is_constant":fals
e,"rows":0,"columns":0},"row":null,"column":null,"include_constant
":true,"dimensions":0},{type_op":"exp_arithmetic","item":"division"
},{type":"var_value","variable_and_value":7,"content":"10","row":n
ull,"column":null,"include_constant":true,"dimensions":0}]},"type"
:"attribution","variable":{"type":"var_value","variable_and_value":
1,"content":{"type":"integer","name":"unidades","value":0,"dimensio
ns":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column"
":null,"include_constant":false,"dimensions":0},"expression":[{"type
":"var_value","variable_and_value":7,"content":{"type":"integer","n
ame":"dezenas","value":0,"dimensions":0,"is_constant":false,"rows":
0,"columns":0},"row":null,"column":null,"include_constant":true,"di
mensions":0},{type_op":"exp_arithmetic","item":"minus"},{"type":"v
ar_value","variable_and_value":7,"content":"10","row":null,"column"
":null,"include_constant":true,"dimensions":0},{type_op":"exp_arith
metic","item":"multiplication"},{"type":"var_value","variable_and_v
alue":7,"content":{"type":"integer","name":"digitodezenas","value":
0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":n
ull,"column":null,"include_constant":true,"dimensions":0}]},"type":
"writer","content":["(","{"type":"var_value","variable_and_value":7,
"content":{"type":"integer","name":"unidades","value":0,"dimensions
":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":n
ull,"include_constant":true,"dimensions":0},{type_op":"exp_arithme
tic","item":"multiplication"},{"type":"var_value","variable_and_val
ue":7,"content":"100","row":null,"column":null,"include_constant":t
rue,"dimensions":0},")","{"type_op":"exp_arithmetic","item":"plus"},
(","{"type":"var_value","variable_and_value":7,"content":{"type":"i
nteger","name":"digitodezenas","value":0,"dimensions":0,"is_constan
t":false,"rows":0,"columns":0},"row":null,"column":null,"include_co
nstant":true,"dimensions":0},{type_op":"exp_arithmetic","item":"mu
ltiplication"},{"type":"var_value","variable_and_value":7,"content"
:"10","row":null,"column":null,"include_constant":true,"dimensions"
:0},")","{"type_op":"exp_arithmetic","item":"plus"},{"type":"var_val
ue","variable_and_value":7,"content":{"type":"integer","name":"div1
00","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns"
:0},"row":null,"column":null,"include_constant":true,"dimensions":0
}],newline":true}}]},"globals":[]}]

```

## A.12 - IDENTIFICAR SE DOIS NÚMEROS SÃO IGUAIS OU DIFERENTES

Elabore um programa que solicite que o usuário digite dois valores inteiros, imprimindo IGUAIS se eles forem iguais e DIFERENTES caso eles sejam diferentes.

```
{
  "exercisetype" : ["Condição" ],
  "testcases" : [
    {
      "input": ["5", "6"],
      "output": ["DIFERENTES"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["-5", "6"],
      "output": ["DIFERENTES"],
      "tag": ["Caso Negativo"]
    },
    {
      "input": ["0", "0"],
      "output": ["IGUAIS"],
      "tag": ["Caso Nulo"]
    },
    {
      "input": ["0", "2"],
      "output": ["DIFERENTES"],
      "tag": ["Caso Nulo"]
    },
    {
      "input": ["5", "5"],
      "output": ["IGUAIS"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["8.5", "8"],
      "output": ["IGUAIS"],
      "tag": ["Tipo de Entrada"]
    },
    {

```

```


"input": ["-7", "-7"],
"output": ["IGUAIS"],
"tag": ["Caso Negativo"]
}
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "boolean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":
[{"name": "commands_read", "value": "on"}, {"name": "commands_write", "value": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "commands_attribution", "value": "on"}, {"name": "commands_functioncall", "value": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"}, {"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch", "value": "on"}],
"settings_functions":
[{"name": "functions_creation", "value": "on"}, {"name": "functions_move", "value": "on"}],
"settings_filter":
[] }
::algorithm:{"functions":[{"type": "function", "name": "main", "return_type": "void", "return_dimensions": 0, "parameters_list": [], "is_main": true, "is_hidden": false, "variables_list": [{"type": "integer", "name": "variable_0", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, {"type": "integer", "name": "variable_1", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}], "function_comment": {"type": "comment", "comment_text": "This is the main function"}, "commands": [{"type": "reader", "variable_value_menu": {"type": "var_value", "variable_and_value": 1, "content": {"type": "integer", "name": "variable_0", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, "row": null, "column": null, "include_constant": false, "dimensions": 0}}, {"type": "reader", "variable_value_menu": {"type": "var_value", "variable_and_value": 1, "content": {"type": "integer", "name": "variable_1", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, "row": null, "column": null, "include_constant": false, "dimensions": 0}}, {"type": "iftrue", "expression": [{"type": "var_value", "v

```

```

variable_and_value":7,"content":{"type":"integer","name":"variable_0
","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0
},"row":null,"column":null,"include_constant":true,"dimensions":0},
{"type_op":"exp_conditional","item":"equals_to"},{"type":"var_value
","variable_and_value":7,"content":{"type":"integer","name":"variab
le_1","value":0,"dimensions":0,"is_constant":false,"rows":0,"column
s":0},"row":null,"column":null,"include_constant":true,"dimensions
":0}},"commands_block":[{"type":"writer","content":[{"type":"var_val
ue","variable_and_value":7,"content":"IGUAIS","row":null,"column":n
ull,"include_constant":true,"dimensions":0}], "newline":true}], "comm
ands_else":[{"type":"writer","content":[{"type":"var_value","variab
le_and_value":7,"content":"DIFERENTES","row":null,"column":null,"in
clude_constant":true,"dimensions":0}], "newline":true}]]]]], "globals
":[]}]

```

### A.13 - VERIFICAR SE UM NÚMERO É DIVISÍVEL POR OUTRO

Construa um algoritmo que receba 2 números naturais não nulos (digamos, num e divi) e imprima 1 se o primeiro é divisível pelo segundo e 0 se num não for divisível por divi. Deve-se supor que o usuário digitará valor adequado.

```

{
  "exercisetype" : ["Condição" ],
  "testcases" : [
    {
      "input": ["2", "2"],
      "output": ["1"],
      "tag": ["Igualdade"]
    },
    {
      "input": ["5", "3"],
      "output": ["0"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["10", "5"],
      "output": ["1"],
      "tag": ["Caso Geral"]
    },
  ],
}

```

```

{
  "input": ["15", "5"],
  "output": ["1"],
  "tag": ["Caso Geral"]
},
{
  "input": ["57", "36"],
  "output": ["0"],
  "tag": ["Caso Geral"]
},
{
  "input": ["174", "251"],
  "output": ["0"],
  "tag": ["Caso Geral"]
},
{
  "input": ["5", "2.5"],
  "output": ["0"],
  "tag": ["Tipo de Entrada"]
},
{
  "input": ["11", "5"],
  "output": ["0"],
  "tag": ["Caso Limite"]
},
{
  "input": ["9", "5"],
  "output": ["0"],
  "tag": ["Caso Limite"]
}
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "boolean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":
[{"name": "commands_read", "value": "on"}, {"name": "commands_write", "value": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "command"}

```

```

ds_attribution", "value": "on"}, {"name": "commands_functioncall", "value": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"}, {"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch", "value": "on"}],
"settings_functions":
[{"name": "functions_creation", "value": "on"}, {"name": "functions_move", "value": "on"}],
"settings_filter":
[] }
::algorithm::{"functions":[{"type":"function","name":"main","return_type":"void","return_dimensions":0,"parameters_list":[],"is_main":true,"is_hidden":false,"variables_list":[{"type":"integer","name":"variable_0","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0}, {"type":"integer","name":"variable_1","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0}], "function_comment":{"type":"comment","comment_text":"This is the main function"}, "commands":[{"type":"reader","variable_value_menu":{"type":"var_value","variable_and_value":1,"content":{"type":"integer","name":"variable_0","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":false,"dimensions":0}}, {"type":"reader","variable_value_menu":{"type":"var_value","variable_and_value":1,"content":{"type":"integer","name":"variable_1","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":false,"dimensions":0}}, {"type":"iftrue","expression":["(", {"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"variable_0","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}, {"type_op":"exp_arithmetic","item":"module"}, {"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"variable_1","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}, ")"], {"type_op":"exp_logic","item":"equals_to"}, {"type":"var_value","variable_and_value":7,"content":"0", "row":null,"column":null,"include_constant":true,"dimensions":0}], "commands_block":[{"type":"writer","content":[{"type":"var_value","variable_and_value":7,"content":"1", "row":null,"column":null,"include_constant":true,"dimensions":0}], "newline":true}], "commands_else":[{"type":"writer","content":[{"type":"var_value","variable_and_value":7,"content":"0", "row":null,"column":null,"include_constant":true,"dimensions":0}], "newline

```

```
":true}}]}]}], "globals": []}
```

#### A.14 - IMPRIMIR O MAIOR NÚMERO ENTRE DOIS INTEIROS

Faça um programa que solicite que o usuário digite dois valores inteiros. Em seguida, verifique e imprima o maior deles.

```
{
  "exercisetype" : ["Condição" ],
  "testcases" : [
    {
      "input": ["-1", "5"],
      "output": ["5"],
      "tag": ["Caso Negativo"]
    },
    {
      "input": ["9", "47"],
      "output": ["47"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["14", "5"],
      "output": ["14"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["1", "2"],
      "output": ["2"],
      "tag": ["Caso Limite"]
    },
    {
      "input": ["2", "1"],
      "output": ["2"],
      "tag": ["Caso Limite"]
    },
    {
      "input": ["-14526", "-19586"],
      "output": ["-14526"],
      "tag": ["Caso Negativo"]
    },
  ],
}
```

```

{
  "input": ["-8596", "8596"],
  "output": ["8596"],
  "tag": ["Caso Negativo"]
},
{
  "input": ["0", "2"],
  "output": ["2"],
  "tag": ["Caso Nulo"]
},
{
  "input": ["5", "5"],
  "output": ["5"],
  "tag": ["Igualdade"]
},
{
  "input": ["4.5", "4"],
  "output": ["4"],
  "tag": ["Tipo de Entrada"]
}
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "boolean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":
[{"name": "commands_read", "value": "on"}, {"name": "commands_write", "value": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "commands_attribution", "value": "on"}, {"name": "commands_functioncall", "value": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"}, {"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch", "value": "on"}],
"settings_functions":
[{"name": "functions_creation", "value": "on"}, {"name": "functions_move", "value": "on"}],
"settings_filter":
[] }

```

```

::algorithm::{
  "functions": [
    {
      "type": "function",
      "name": "main",
      "return_type": "void",
      "return_dimensions": 0,
      "parameters_list": [],
      "is_main": true,
      "is_hidden": false,
      "variables_list": [
        {
          "type": "integer",
          "name": "variable_0",
          "value": 0,
          "dimensions": 0,
          "is_constant": false,
          "rows": 0,
          "columns": 0
        },
        {
          "type": "integer",
          "name": "variable_1",
          "value": 0,
          "dimensions": 0,
          "is_constant": false,
          "rows": 0,
          "columns": 0
        }
      ],
      "function_comment": {
        "type": "comment",
        "comment_text": "This is the main function"
      },
      "commands": [
        {
          "type": "reader",
          "variable_value_menu": {
            "type": "var_value",
            "variable_and_value": 1,
            "content": {
              "type": "integer",
              "name": "variable_0",
              "value": 0,
              "dimensions": 0,
              "is_constant": false,
              "rows": 0,
              "columns": 0
            },
            "row": null,
            "column": null,
            "include_constant": false,
            "dimensions": 0
          },
          "row": null,
          "column": null,
          "include_constant": false,
          "dimensions": 0
        },
        {
          "type": "reader",
          "variable_value_menu": {
            "type": "var_value",
            "variable_and_value": 1,
            "content": {
              "type": "integer",
              "name": "variable_1",
              "value": 0,
              "dimensions": 0,
              "is_constant": false,
              "rows": 0,
              "columns": 0
            },
            "row": null,
            "column": null,
            "include_constant": false,
            "dimensions": 0
          },
          "row": null,
          "column": null,
          "include_constant": false,
          "dimensions": 0
        },
        {
          "type": "iftrue",
          "expression": [
            {
              "type": "var_value",
              "variable_and_value": 7,
              "content": {
                "type": "integer",
                "name": "variable_0",
                "value": 0,
                "dimensions": 0,
                "is_constant": false,
                "rows": 0,
                "columns": 0
              },
              "row": null,
              "column": null,
              "include_constant": true,
              "dimensions": 0
            },
            {
              "type_op": "exp_conditional",
              "item": "greater_than_or_equals_to"
            },
            {
              "type": "var_value",
              "variable_and_value": 7,
              "content": {
                "type": "integer",
                "name": "variable_1",
                "value": 0,
                "dimensions": 0,
                "is_constant": false,
                "rows": 0,
                "columns": 0
              },
              "row": null,
              "column": null,
              "include_constant": true,
              "dimensions": 0
            }
          ],
          "commands_block": [
            {
              "type": "writer",
              "content": [
                {
                  "type": "var_value",
                  "variable_and_value": 7,
                  "content": {
                    "type": "integer",
                    "name": "variable_0",
                    "value": 0,
                    "dimensions": 0,
                    "is_constant": false,
                    "rows": 0,
                    "columns": 0
                  },
                  "row": null,
                  "column": null,
                  "include_constant": true,
                  "dimensions": 0
                },
                {
                  "type": "newline",
                  "content": true
                }
              ],
              "newlines": true
            },
            {
              "type": "writer",
              "content": [
                {
                  "type": "var_value",
                  "variable_and_value": 7,
                  "content": {
                    "type": "integer",
                    "name": "variable_1",
                    "value": 0,
                    "dimensions": 0,
                    "is_constant": false,
                    "rows": 0,
                    "columns": 0
                  },
                  "row": null,
                  "column": null,
                  "include_constant": true,
                  "dimensions": 0
                },
                {
                  "type": "newline",
                  "content": true
                }
              ],
              "newlines": true
            }
          ],
          "globals": []
        }
      ]
    }
  ]
}

```

#### A.15 - SOMATÓRIO DOS PRIMEIROS NATURAIS ATÉ UM LIMITE

Construa um programa que lê um natural (digamos na variável total) e computa a maior soma dentre os primeiros números naturais (consecutivos), que seja menor ou igual ao valor digitado ( $\leq$ total).

```
{
  "exercisetype" : ["Laço","Condição" ],
  "testcases" : [
    {
      "input": ["85"],
      "output": ["78"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["95"],
      "output": ["91"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["125"],
      "output": ["120"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["1256"],
      "output": ["1225"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["987"],
      "output": ["946"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["12"],
      "output": ["10"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["15"],
      "output": ["15"],
      "tag": ["Caso Limite"]
    },
    {
      "input": ["466"],
      "output": ["465"],
```

```

    "tag": ["Caso Limite"]
  },
  {
    "input": ["250"],
    "output": ["231"],
    "tag": ["Caso Geral"]
  },
  {
    "input": ["347"],
    "output": ["325"],
    "tag": ["Caso Geral"]
  },
  {
    "input": ["105"],
    "output": ["105"],
    "tag": ["Caso Limite"]
  },
  {
    "input": ["104"],
    "output": ["91"],
    "tag": ["Caso Geral"]
  },
  {
    "input": ["0"],
    "output": ["0"],
    "tag": ["Caso Nulo"]
  },
  {
    "input": ["5.5"],
    "output": ["3"],
    "tag": ["Tipo de Entrada"]
  }
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "boolean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":

```

```

[{"name":"commands_read","value":"on"},{"name":"commands_write","value":"on"},{"name":"commands_comment","value":"on"},{"name":"commands_attribution","value":"on"},{"name":"commands_functioncall","value":"on"},{"name":"commands_iftrue","value":"on"},{"name":"commands_repeatNtimes","value":"on"},{"name":"commands_while","value":"on"},{"name":"commands_dowhile","value":"on"},{"name":"commands_switch","value":"on"}],
"settings_functions":
[{"name":"functions_creation","value":"on"},{"name":"functions_move","value":"on"}],
"settings_filter":
[] }
::algorithm::{
  "functions":[{"type":"function","name":"main","return_type":"void","return_dimensions":0,"parameters_list":[],"is_main":true,"is_hidden":false,"variables_list":[{"type":"integer","name":"num","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},{"type":"integer","name":"soma","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},{"type":"integer","name":"i","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},{"type":"integer","name":"f","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0}],
  "function_comment":{"type":"comment","comment_text":"This is the main function"},
  "commands":[{"type":"reader","variable_value_menu":{"type":"var_value","variable_and_value":1,"content":{"type":"integer","name":"num","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":false,"dimensions":0},{"type":"whiletrue","expression":[{"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"f","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0},{"type_op":"exp_conditional","item":"not_equals_to"},{"type":"var_value","variable_and_value":7,"content":"1","row":null,"column":null,"include_constant":true,"dimensions":0}],
  "commands_block":[{"type":"attribution","variable":{"type":"var_value","variable_and_value":1,"content":{"type":"integer","name":"i","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":false,"dimensions":0},"expression":[{"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"i","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0},{"type_op":"exp_arithmetic","item":"plus"},{"type":"var_value","variable_and

```

```

_value":7,"content":"1","row":null,"column":null,"include_constant":true,"dimensions":0}}],{"type":"iftrue","expression":["(","type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"soma","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0},{"type_op":"exp_arithmetic","item":"plus"},{"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"i","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0},")",{"type_op":"exp_conditional","item":"less_than_or_equals_to"},{"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"num","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}],{"commands_block":[{"type":"attribution","variable":{"type":"var_value","variable_and_value":1,"content":{"type":"integer","name":"soma","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":false,"dimensions":0},"expression":[{"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"soma","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0},{"type_op":"exp_arithmetic","item":"plus"},{"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"i","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}]]}],{"commands_else":[{"type":"attribution","variable":{"type":"var_value","variable_and_value":1,"content":{"type":"integer","name":"f","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":false,"dimensions":0},"expression":[{"type":"var_value","variable_and_value":7,"content":"1","row":null,"column":null,"include_constant":true,"dimensions":0}]]}}],{"type":"writer","content":[{"type":"var_value","variable_and_value":7,"content":{"type":"integer","name":"soma","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,"column":null,"include_constant":true,"dimensions":0}],{"newline":true}}],{"globals":[]}]

```

#### A.16 - SOMA DAS POSIÇÕES X E Y

Nesta tarefa, construa um programa que receba 12 valores e os armazene em um vetor. Em seguida ler também dois números X e Y quaisquer, correspondentes a

duas posições no vetor. Ao final, seu programa deverá escrever a soma dos valores encontrados nas respectivas posições X e Y.

```
{
  "exercisetype" : ["Vetor" ],
  "testcases" : [
    {
      "input": ["383", "886", "777", "915", "793", "335", "386", "492",
"649", "421", "362", "27", "1", "3"],
      "output": ["1801"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["85", "130", "0", "326", "280", "475", "111", "529",
"47", "997", "898", "359", "3", "2"],
      "output": ["326"],
      "tag": ["Caso Nulo"]
    },
    {
      "input": ["307", "653", "-602", "677", "93", "347", "127", "384",
"991", "264", "318", "455", "9", "2"],
      "output": ["-338"],
      "tag": ["Caso Negativo"]
    },
    {
      "input": ["65", "327", "781", "-846", "388", "827", "748", "821",
"-45", "941", "0", "61", "3", "8"],
      "output": ["-891"],
      "tag": ["Caso Negativo"]
    },
    {
      "input": ["121", "721", "0", "978", "501", "886", "323", "450",
"761", "208", "929", "910", "8", "4"],
      "output": ["1262"],
      "tag": ["Caso Geral"]
    },
    {
      "input": ["457", "352", "818", "135", "757", "988", "496", "884",
"128", "452", "200", "7", "11", "0"],
      "output": ["464"],
      "tag": ["Caso Limite"]
    }
  ]
}
```

```

},
{
  "input": ["969.5", "669.6", "421.4", "645.2", "849.4", "530.6",
"857.8", "214.7", "623.1", "374.9", "646.2", "785.3", "7", "6"],
  "output": ["1071"],
  "tag": ["Tipo de Entrada"]
},
{
  "input": ["531", "73", "240", "315", "679", "443", "401", "230",
"326", "558", "775", "825", "0", "6"],
  "output": ["932"],
  "tag": ["Caso Geral"]
},
{
  "input": ["559", "602", "282", "405", "16", "229", "816", "490",
"257", "982", "785", "708", "2", "2"],
  "output": ["564"],
  "tag": ["Igualdade"]
},
{
  "input": ["900", "468", "803", "822", "100", "900", "391", "327",
"240", "976", "557", "919", "5", "0"],
  "output": ["1800"],
  "tag": ["Igualdade"]
}
],
"settings_programming_type":
[{"name": "programming_type", "value": "visual"}],
"settings_data_types":
[{"name": "integer_data_type", "value": "on"}, {"name": "real_data_type",
"value": "on"}, {"name": "text_data_type", "value": "on"}, {"name": "boolean_data_type", "value": "on"}, {"name": "void_data_type", "value": "on"}
],
"settings_commands":
[{"name": "commands_read", "value": "on"}, {"name": "commands_write", "value": "on"}, {"name": "commands_comment", "value": "on"}, {"name": "commands_attribution", "value": "on"}, {"name": "commands_functioncall", "value": "on"}, {"name": "commands_iftrue", "value": "on"}, {"name": "commands_repeatNtimes", "value": "on"}, {"name": "commands_while", "value": "on"}, {"name": "commands_dowhile", "value": "on"}, {"name": "commands_switch", "value": "on"}],

```

```

"settings_functions":
[{"name":"functions_creation","value":"on"},{"name":"functions_move",
"value":"on"}],
"settings_filter":
[] }
::algorithm::{
"functions":[{"type":"function","name":"main","return_type":"void",
"return_dimensions":0,"parameters_list":[],"is_main":true,"is_hidden":false,
"variables_list":[{"type":"integer","name":"vet","value":[0,0,0,0,0,0,0,0,0,0,0,0],
"dimensions":1,"is_constant":false,"rows":1,"columns":12},{"type":"integer",
"name":"i","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},
{"type":"integer","name":"x","value":0,"dimensions":0,"is_constant":false,
"rows":0,"columns":0},{"type":"integer","name":"y","value":0,"dimensions":0,
"is_constant":false,"rows":0,"columns":0}],
"function_comment":{"type":"comment","comment_text":"This is the main
function"},
"commands":[{"type":"repeatNtimes","var_attribution":{"type":"var_value",
"variable_and_value":1,"content":{"type":"integer","name":"i","value":0,
"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,
"column":null,"include_constant":false,"dimensions":0},"var_incrementation":{
"type":"var_value","variable_and_value":1,"content":{"type":"integer",
"name":"i","value":0,"dimensions":0,"is_constant":false,"rows":0,"columns":0},
"row":null,"column":null,"include_constant":false,"dimensions":0},
"expression1":{"type":"var_value","variable_and_value":7,"content":"0",
"row":null,"column":null,"include_constant":true,"dimensions":0},
"expression2":{"type":"var_value","variable_and_value":7,"content":"12",
"row":null,"column":null,"include_constant":true,"dimensions":0},
"expression":{"type":"exp_arithmetic","first_operand":{"type":"var_value",
"variable_and_value":1,"content":{"type":"integer","name":"i","value":0,
"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,
"column":null,"include_constant":false,"dimensions":0},
"second_operand":{"type":"var_value","variable_and_value":7,"content":null,
"row":null,"column":null,"include_constant":true,"dimensions":0},
"operator":"less_than"}},
"expression3":{"type_exp":"exp_op_exp","itens":[{"type":"var_value",
"variable_and_value":1,"content":{"type":"integer","name":"i","value":0,
"dimensions":0,"is_constant":false,"rows":0,"columns":0},"row":null,
"column":null,"include_constant":false,"dimensions":0},
"plus",{"type":"var_value","variable_and_value":7,"content":"1",
"row":null,"column":null,"include_constant":true,"dimensions":0}]}},
"commands_block":[{"type":"reader","variable_value_menu":{"type":"var_value",
"variable_and_value":1,"content":{"type":"integer",

```

```

"name": "vet", "value": [0,0,0,0,0,0,0,0,0,0,0,0], "dimensions": 1, "is_constant": false, "rows": 1, "columns": 12}, "row": null, "column": {"type": "var_value", "variable_and_value": 7, "content": {"type": "integer", "name": "i", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, "row": null, "column": null, "include_constant": true, "dimensions": 0}, "include_constant": false, "dimensions": 0}}], {"type": "reader", "variable_value_menu": {"type": "var_value", "variable_and_value": 1, "content": {"type": "integer", "name": "x", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, "row": null, "column": null, "include_constant": false, "dimensions": 0}, {"type": "reader", "variable_value_menu": {"type": "var_value", "variable_and_value": 1, "content": {"type": "integer", "name": "y", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, "row": null, "column": null, "include_constant": false, "dimensions": 0}}, {"type": "writer", "content": [{"type": "var_value", "variable_and_value": 7, "content": {"type": "integer", "name": "vet", "value": [0,0,0,0,0,0,0,0,0,0,0,0], "dimensions": 1, "is_constant": false, "rows": 1, "columns": 12}, "row": null, "column": {"type": "var_value", "variable_and_value": 7, "content": {"type": "integer", "name": "x", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, "row": null, "column": null, "include_constant": true, "dimensions": 0}, "include_constant": true, "dimensions": 0}, {"type_op": "exp_arithmetic", "item": "plus"}, {"type": "var_value", "variable_and_value": 7, "content": {"type": "integer", "name": "vet", "value": [0,0,0,0,0,0,0,0,0,0,0,0], "dimensions": 1, "is_constant": false, "rows": 1, "columns": 12}, "row": null, "column": {"type": "var_value", "variable_and_value": 7, "content": {"type": "integer", "name": "y", "value": 0, "dimensions": 0, "is_constant": false, "rows": 0, "columns": 0}, "row": null, "column": null, "include_constant": true, "dimensions": 0}, "include_constant": true, "dimensions": 0}], "newline": true}}], "globals": []}

```