



UNIVERSIDADE FEDERAL DA PARAÍBA
CAMPUS I - UNIDADE ACADÊMICA REITOR LYNALDO CAVALCANTI DE
ALBUQUERQUE
CENTRO DE INFORMÁTICA

ESPECIFICAÇÕES DE REQUISITOS DE SOFTWARE - MARCO FINAL:
DOCUMENTO DE REQUISITOS PARA ABERTURA DO DESENVOLVIMENTO DO
JOGO DIGITAL HEFESTO

DIEGO LUIS REIS DA SILVA
LEONARDO DO NASCIMENTO PEIXOTO DA SILVA
MARCOS ANDRE BEZERRA DA SILVA
NATERCIO ROQUE DA SILVA NETO

JOÃO PESSOA
2022

DIEGO LUIS REIS DA SILVA

LEONARDO DO NASCIMENTO PEIXOTO DA SILVA

MARCOS ANDRE BEZERRA DA SILVA

NATERCIO ROQUE DA SILVA NETO

**ESPECIFICAÇÕES DE REQUISITOS DE SOFTWARE - MARCO FINAL:
DOCUMENTO DE REQUISITOS PARA ABERTURA DO DESENVOLVIMENTO
DO JOGO DIGITAL HEFESTO**

Projeto final - Documento de abertura para desenvolvimento de jogo digital Hefesto - realizado para obtenção de nota parcial da disciplina Especificações de Requisitos de Software.

Professora: Danielle Rousy Dias Ricarte

JOÃO PESSOA

2022



VERSÃO DO DOCUMENTO 1.0.0

1 INTRODUÇÃO

1.1 OBJETIVOS

1.1.1 Objetivo do documento

Esse documento visa ambientar, justificar, especificar e promover a criação do jogo focado em aprendizagem Hefesto por meio da empresa CI-Games, produtora e companhia de jogos no Centro de Informática da Universidade Federal da Paraíba.

Versão do documento: 1.0.0.

1.1.2 Objetivo do jogo Hefesto

O jogo Hefesto tem como seu principal objetivo ajudar, estimular, aprimorar e desenvolver conceitos básicos do paradigma de linguagem Programação Orientada a Objetos de uma maneira lúdica, sem ser cansativa, para seus jogadores-estudantes.

1.2 PÚBLICO ALVO

O principal público alvo do Hefesto são os estudantes iniciantes (de semestres iniciais) do Centro de Informática da Universidade Federal da Paraíba que estão cursando ou ainda cursarão a disciplina de Linguagem de Programação I, na qual é abordado o conceito de Programação Orientada a Objetos.

1.3 ORGANIZAÇÃO DO DOCUMENTO

Esse documento, para uma melhor compreensão, foi dividido em seções e subseções apresentadas da seguinte forma:

1 INTRODUÇÃO	4
1.1 OBJETIVOS	4
1.1.1 Objetivo do documento	4
1.1.2 Objetivo do jogo Hefesto	4
1.2 PÚBLICO ALVO	4

1.3 ORGANIZAÇÃO DO DOCUMENTO	4
1.4 DEFINIÇÃO DE SIGLAS E NOMENCLATURAS	5
2 DESCRIÇÃO DO PROBLEMA E DO SISTEMA	6
2.1 DOMÍNIO DO PROBLEMA E CONTEXTO DE SUA APLICAÇÃO	6
2.2 IDENTIFICAÇÃO E MISSÃO DO SISTEMA E DIAGRAMA DE CONTEXTO	8
2.3 DESCRIÇÃO DOS USUÁRIOS DO SISTEMA	9
3 REQUISITOS DA APLICAÇÃO	9
3.1 REQUISITOS FUNCIONAIS	9
3.2 REQUISITOS NÃO FUNCIONAIS	10
4 CASOS DE USO	11
4.1 DIAGRAMA DE CASO DE USO	11
4.2 LISTA DE CASOS DE USO	12
4.3 DESCRIÇÃO DE CASOS DE USO	13
5 MODELO NAVEGACIONAL	15
6 DESCRIÇÃO DE INTERFACE COM USUÁRIO (PROTOTIPAÇÃO)	16
7 RASTREABILIDADE	18
7.1 CASOS DE USO VS. REQUISITOS FUNCIONAIS	19
7.2 CASOS DE USO VS. REQUISITOS NÃO FUNCIONAIS	19
7.3 REQUISITOS FUNCIONAIS VS. REQUISITOS FUNCIONAIS	20
7.4 REQUISITOS NÃO FUNCIONAIS VS. REQUISITOS NÃO FUNCIONAIS	20
7.5 REQUISITOS FUNCIONAIS VS. REQUISITOS NÃO FUNCIONAIS	20
8 BIBLIOGRAFIA	21
9 APÊNDICES	21
9.1 PROCESSO DE DESENVOLVIMENTO	22
9.2 - PLANO DE ELICITAÇÃO	22
9.2.1 Metodologia	23
9.2.2 Cronograma	23
9.2.3 Resultados	23
9.2.4 Conclusão	27
9.3 DIAGRAMA DE CLASSES	27

1.4 DEFINIÇÃO DE SIGLAS E NOMENCLATURAS

1. UFPB - Universidade Federal da Paraíba
2. CI - Centro de Informática

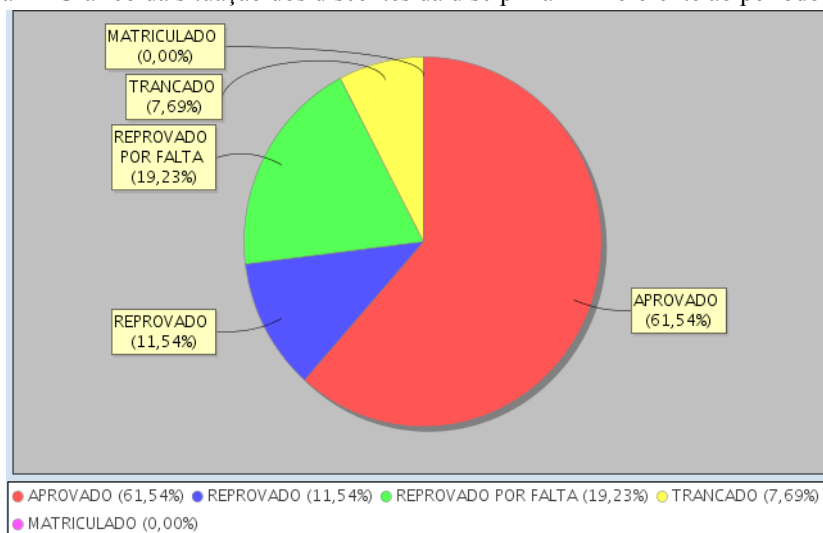
3. POO - Programação Orientada a Objetos
4. LPI - Linguagem de Programação I
5. *game design* - Design de jogos
6. *drag and drop* - Arrastar e soltar
7. *craft* - Criação
8. Objeto - Instância concreta em paradigma de POO

2 DESCRIÇÃO DO PROBLEMA E DO SISTEMA

2.1 DOMÍNIO DO PROBLEMA E CONTEXTO DE SUA APLICAÇÃO

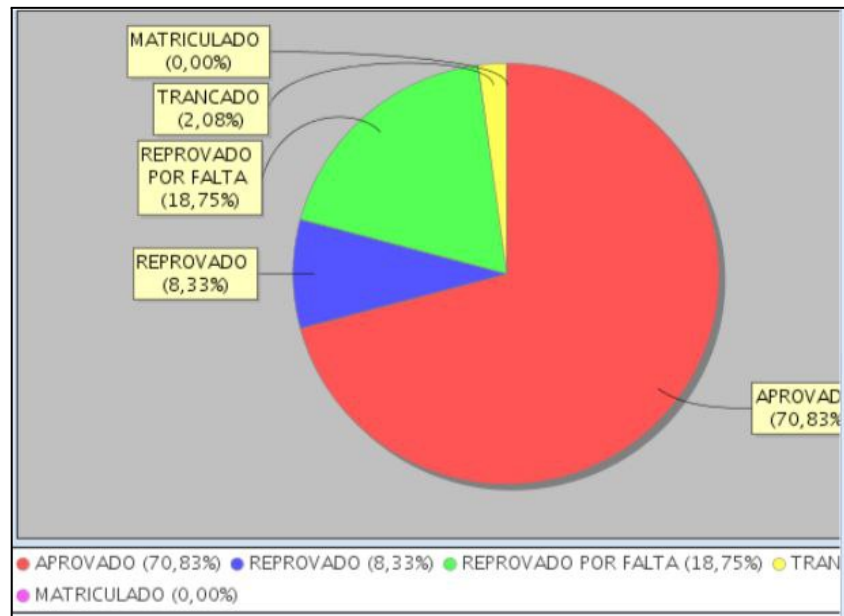
Muitos estudantes sofrem dificuldade em aprender o paradigma de POO e seus conceitos. Geralmente, começando no mundo da programação com a Programação Estruturada, esses podem sentir um imenso impasse ao realizar a passagem para a POO. O histórico de reprovações e trancamentos dos anos 2018.1, 2019.2, 2020.2, 2021.1 (Figuras 1, 2, 3 e 4 respectivamente), mostrados a seguir, ajudam a comprovar tal situação:

Figura 1 - Gráfico da situação dos discentes da disciplina LPI referente ao período 2018.1



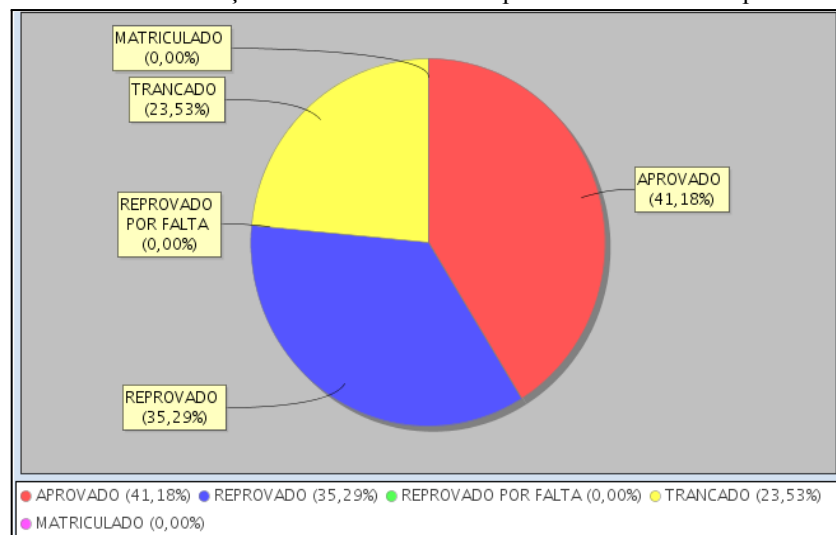
Fonte: SIGAA, 2022.

Figura 2 - Gráfico da situação dos discentes da disciplina LPI referente ao período 2019.2.



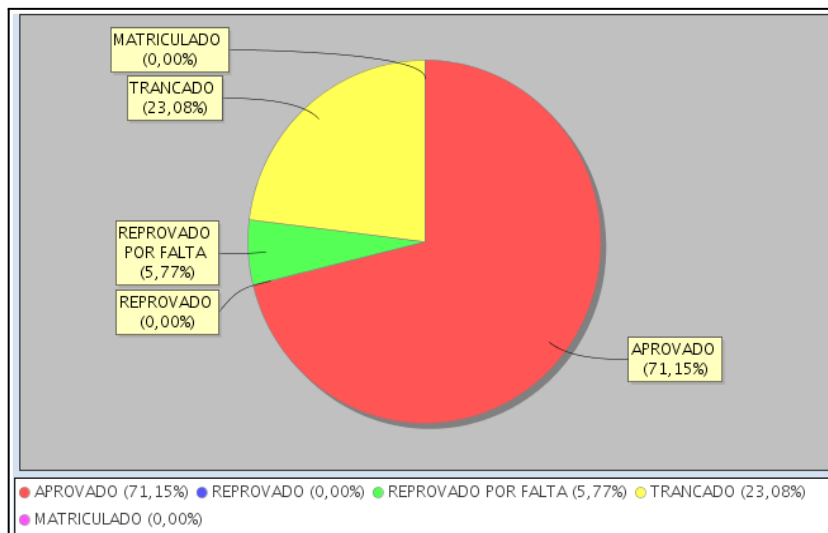
Fonte: SIGAA, 2022.

Figura 3 - Gráfico da situação dos discentes da disciplina LPI referente ao período 2020.2.



Fonte: SIGAA, 2022.

Figura 4 - Gráfico da situação dos discentes da disciplina LPI referente ao período 2021.1.



Fonte: SIGAA, 2022.

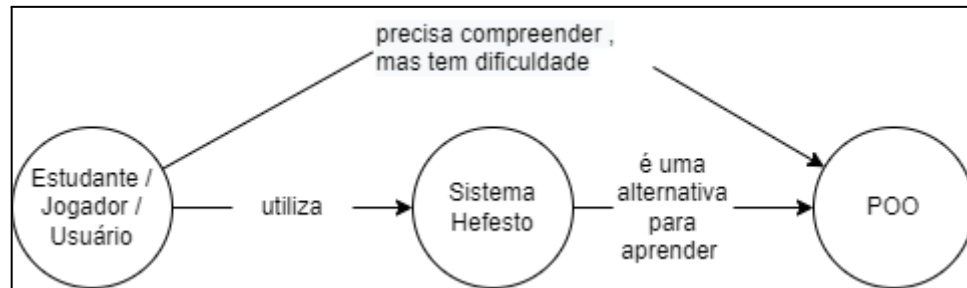
Além de tais números apresentados, um plano de elicitação foi levantado para a turma de Laboratório de Linguagem de Programação I presente no curso de Engenharia da Computação, sendo observado dificuldades em relação à aprendizagem de POO por parte dos estudantes como o apêndice 9.2 pode confirmar.

Tendo em vista essas circunstâncias, o Hefesto pode ser uma solução viável para a resolução desse problema, ajudando, assim, estudantes a atingir um melhor aproveitamento da disciplina.

2.2 IDENTIFICAÇÃO E MISSÃO DO SISTEMA E DIAGRAMA DE CONTEXTO

A missão da ferramenta Hefesto é facilitar e aprimorar o aprendizado de Programação Orientada a Objetos (POO), fazendo uso de artifícios do *game design* para criar uma experiência abstrata e de fácil entendimento. Usando o conceito de gamificação, esse sistema traz em si uma solução inovadora e robusta para superar uma das principais barreiras presentes durante o aprendizado de POO: fazer interligação e comunicação entre os objetos e o programa principal. Um diagrama de contexto, figura 5, foi elaborado para compreender como o sistema Hefesto pode ser inserido nesta problemática de estudantes - POO.

Figura 5 - Diagrama de contexto estudante-Hefesto-POO.



Fonte: própria, 2022.

2.3 DESCRIÇÃO DOS USUÁRIOS DO SISTEMA

Os usuários deste produto são os alunos dos semestres iniciais dos cursos do CI que estão cursando ou cursarão a disciplina de Linguagem de Programação I, na qual é apresentado o paradigma POO e seus conceitos. Geralmente, por estarem no início do aprendizagem da programação, esses sentem dificuldade ao realizar trocas entre paradigmas de programação, especialmente ao se tratar da Programação Orientada a Objetos (a “temível” POO).

3 REQUISITOS DA APLICAÇÃO

Nesta seção, os requisitos avaliados como **funcionais** (RF) e **não funcionais** (RNF) para o jogo Hefesto podem ser analisados nas subseções abaixo. Além dessa divisão em relação a funcionalidades desses requisitos, foram separados em ordem de prioridade também:

Essencial: o software não é aceitável a não ser que estes requisitos sejam implementados.

Importante: devem ser implementados, mas, sem eles, o sistema poderá ser aceitável mesmo assim.

Desejável: podem ser deixados para versões posteriores, caso não haja tempo hábil para implementá-los.

Observação: as descrições dessas prioridades foram retiradas dos slides da Professora Danielle Rousy.

3.1 REQUISITOS FUNCIONAIS

Os requisitos caracterizados como funcionais podem ser encontrados na figura 6 abaixo.

Figura 6 - Requisitos funcionais do jogo Hefesto com algumas especificações próprias do jogo (aba de busca, aba de criação, sons e fx, game design, etc.).



Fonte: própria, 2022. Disponível em: <<https://trello.com/b/eJxURT4N/hefesto-prioridades>>

3.2 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais também estão definidos, na figura 7, abaixo.

Figura 7 - Requisitos não funcionais definidos para o jogo Hefesto.



Fonte: própria, 2022. Disponível em: <<https://trello.com/b/eJxURT4N/hefesto-prioridades>>.

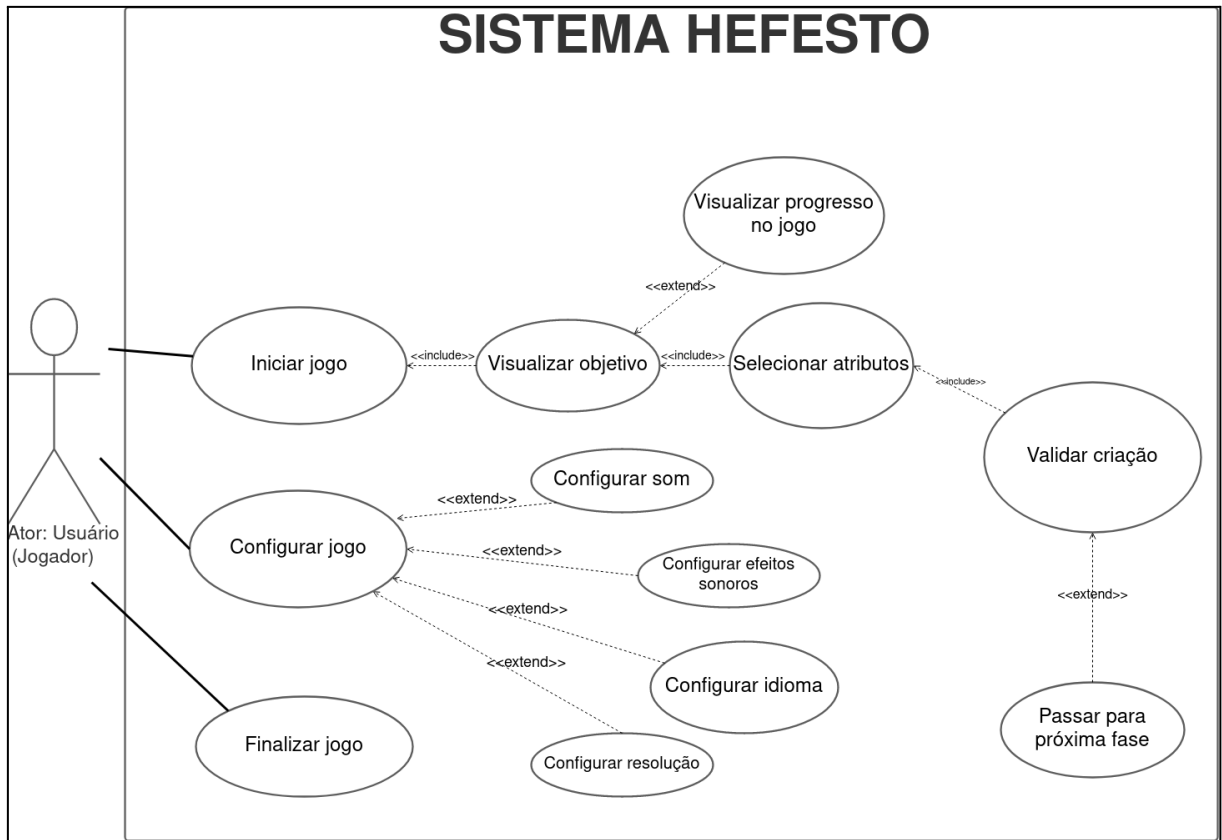
4 CASOS DE USO

Nesta seção, estão definidos: o diagrama de caso de uso, lista e descrição de casos de uso do jogo Hefesto.

4.1 DIAGRAMA DE CASO DE USO

Como se pode notar na figura 8, o diagrama de caso de uso possui três casos centrais (iniciar, configurar e finalizar jogo), sendo que há casos que se estendem ao caso de uso “Iniciar Jogo” e “Configurar jogo”.

Figura 8 - Diagrama de caso de uso simples do sistema Hefesto.



Fonte: própria, 2022.

4.2 LISTA DE CASOS DE USO

Tabela 1 - Lista de casos de uso apresentados no Sistema Hefesto.

ID do Caso de Uso	Nome do caso de uso	Descrição
CSU001	Selecionar Atributos	Arrastar e soltar atributos na mesa de criação
CSU002	Configurar Jogo	Acessar menu para mudar opções do jogo
CSU003	Visualizar Objetivo	Mostrar ao jogador qual objeto deverá ser criado
CSU004	Validar Criação	Verificar se o objeto criado está correto
CSU005	Iniciar Jogo	Começa uma fase ativa
CSU006	Finalizar Jogo	Termina a fase ativa

CSU007	Configurar Som	Alterar volume
CSU008	Configurar Idioma	Alterar idioma
CSU009	Configurar Efeitos Sonoros	Habilitar ou desabilitar efeitos sonoros
CSU010	Configurar Resolução	Mudar resolução do jogo
CSU011	Visualizar Progresso No Jogo	Ver quais fases foram concluídas e quais ainda podem ser liberadas
CSU012	Passar Para a Próxima Fase	Finaliza a fase ativa atualmente e inicia uma nova

Fonte: própria, 2022.

4.3 DESCRIÇÃO DE CASOS DE USO

- **Selecionar atributos (CSU001)**

Prioridade: Essencial

Sumário: O jogador seleciona atributos(características) de uma classe (dentre as pré-disponibilizados pelo sistema) para achar o objeto.

Ator Primário: Jogador.

Ator Secundário: Sistema Hefesto.

Precondições: O jogador precisa estar em uma fase ativa.

Fluxo de Eventos Principal:

1. O jogador seleciona um atributo dentre os exibidos pelo sistema através da mecânica de arrastar e soltar.
2. O sistema atualiza o objeto que está sendo construído pelo jogador.

- **Configurar jogo (CSU002)**

Prioridade: Essencial

Sumário: O jogador pode configurar jogo (som, brilho, idioma, nível de dificuldade, mudar resolução de tela).

Ator Primário: Jogador.

Ator Secundário: Sistema Hefesto.

Precondições: Nenhuma.

Fluxo de Eventos Principal:

1. O jogador, antes de iniciar o jogo ou durante uma fase ativa, pode selecionar a opção de alterar configurações.
2. O sistema abre o menu de configurações.
3. O jogador pode alterar uma ou várias das configurações disponíveis (som, brilho, idioma, nível de dificuldade, resolução).
4. O sistema exige uma confirmação em caso de alteração na resolução ou idioma e atualiza as configurações desejadas.

- **Visualizar objetivo (CSU003)**

Prioridade: Essencial

Sumário: O jogador visualiza qual objeto deve ser construído.

Ator Primário: Jogador.

Ator Secundário: Sistema Hefesto.

Precondições: Nenhuma.

Fluxo de Eventos Principal:

1. O jogador inicia o jogo.
2. O sistema exibe um indicativo de qual objeto deve ser construído pelo jogador.

- **Validar criação (CSU004)**

Prioridade: Essencial

Sumário: O jogador deve pedir ao sistema que avalie a sua resposta, verificando se o objeto construído por ele satisfaz o objetivo.

Ator Primário: Jogador.

Ator Secundário: Sistema Hefesto.

Precondições: Estar em uma fase ativa. Ter elementos selecionados na aba de criação

Fluxo de Eventos Principal:

1. O jogador solicita ao sistema que avalie sua resposta.
2. O sistema avalia a resposta do jogador.

Fluxos Alternativos / Exceção:**Fluxo Alternativo CSU004.FA001:**

1. O sistema avalia a resposta como correta.
2. O jogador segue para a próxima fase.

Fluxo Alternativo CSU004.FA002:

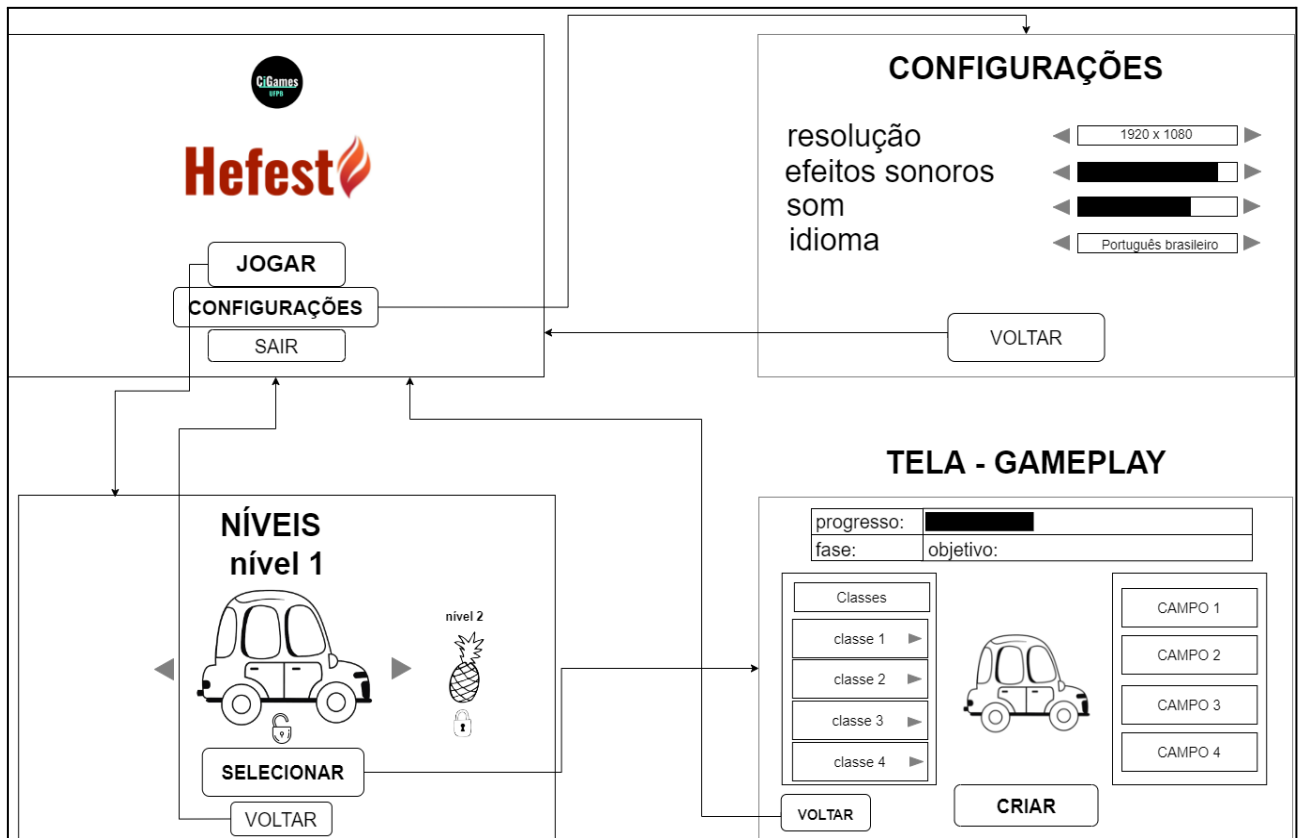
1. O sistema avalia a resposta como incorreta.
2. O jogador pode jogar novamente.

5 MODELO NAVEGACIONAL

O jogo Hefesto possui 4 telas principais (Figura 9):

- 1) Tela Inicial com as opções: jogar, configurações e sair;
- 2) Tela Configurações relacionada apenas com a tela inicial;
- 3) Tela Níveis relacionada com a Tela Inicial e Tela Gameplay;
- 4) Tela Gameplay relacionada a Tela Níveis (o usuário é obrigado a passar na Tela Níveis) para acessar essa tela.

Figura 9 - Modelo Navegacional (design das telas não refletem a versão protótipo do produto).



Fonte: própria, 2022.

6 DESCRIÇÃO DE INTERFACE COM USUÁRIO (PROTOTIPAGEM)

A interface com usuário do jogo Hefesto foi pensada para transmitir sensação de fluidez e simplicidade, com um pouco de toque cartunesco. A seguir, são apresentados como as telas planejadas no modelo navegacional foram projetadas no mundo real.

A primeira tela (Figura 10), a Tela Inicial, o usuário terá as três opções focando em praticidade: JOGAR, CONFIGURAÇÕES e SAIR.

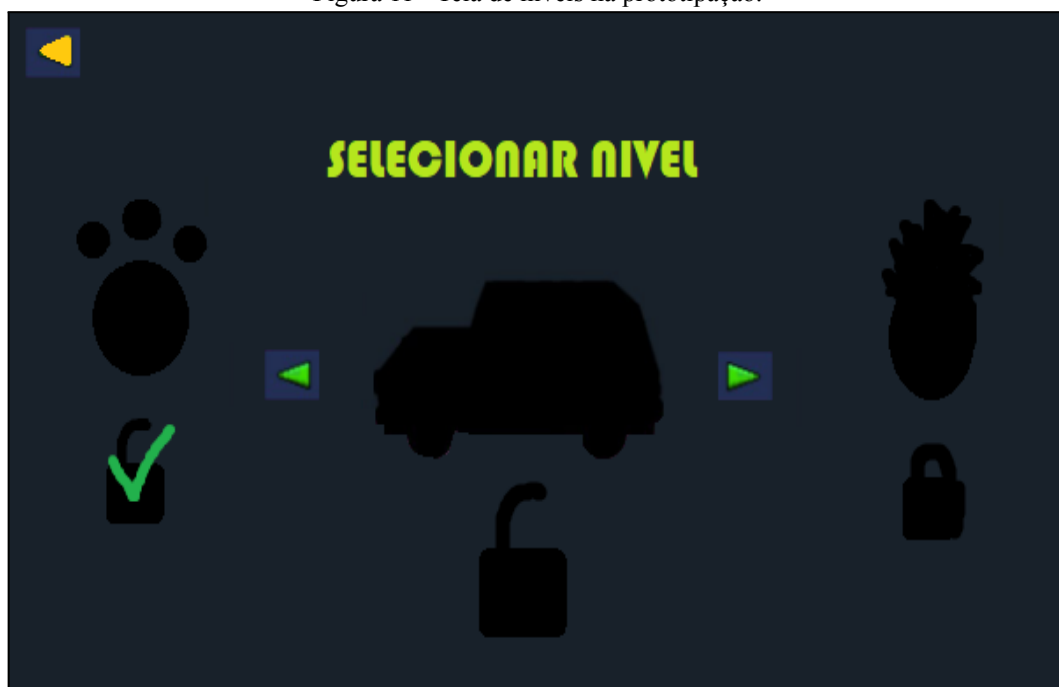
Figura 10 - Menu inicial com as opções de ir para tela de gameplay (Jogar), configurar jogo e sair do jogo.



Fonte: própria, 2022.

Já em relação a Tela de Níveis (Figura 11), essa sofreu uma alteração em relação onde o botão VOLTAR ficará nela comparado com o modelo navegacional, mas possui a mesma função: o usuário pode continuar o progresso do jogo e ver a prévia do tema dos níveis seguintes, além de acessar os níveis já completados.

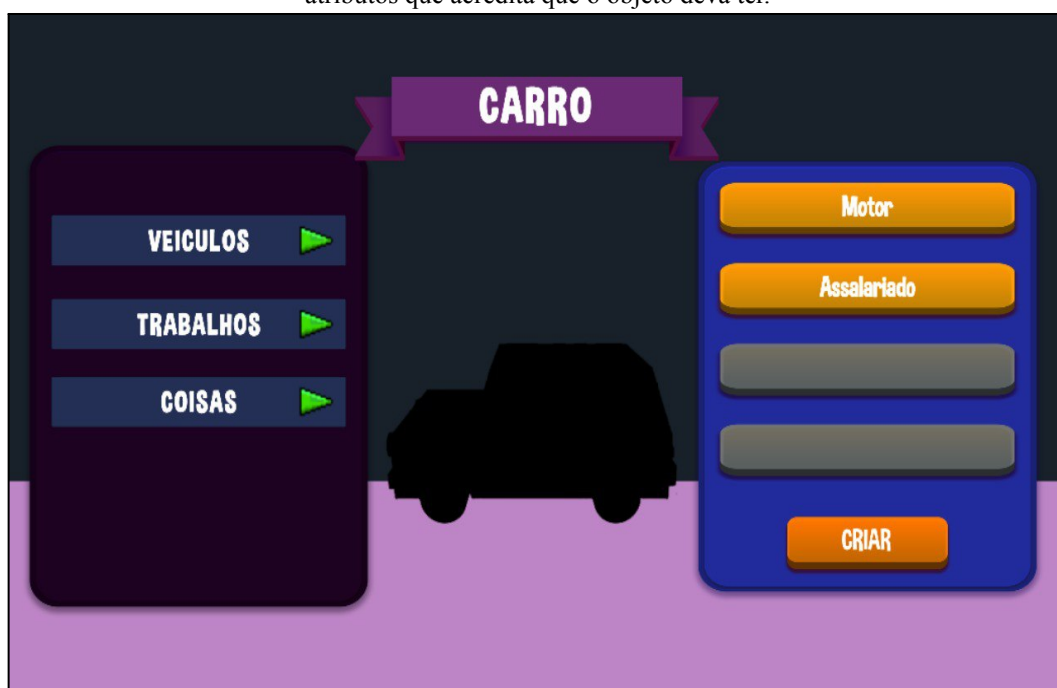
Figura 11 - Tela de níveis na prototipação.



Fonte: própria, 2022.

Por último, a Tela da Gameplay (Figura 12) propriamente dita foi projetada para ter uma menor poluição visual possível, para que o usuário consiga associar o que deve ser feito de modo intuitivo. Percebe-se, ainda, que a mesma não tem a barra de progresso nem mecanismo de experiência e passagem de níveis, implementações que serão realizadas posteriormente.

Figura 12 - Tela de jogo principal na qual o jogador recebe o objetivo (Carro, por exemplo), uma lista de atributos que pode usar (coluna à esquerda) e a mesa de criação (coluna à direita), em que pode colocar os atributos que acredita que o objeto deva ter.



Fonte: própria, 2022.

Em suma, como se pôde perceber por meio dessas telas, o objetivo da interface com o usuário do Hefesto é ser um ambiente focado, sem muitas interações na mesma tela que façam o estudante ficar desviando a atenção do que realmente importa. Além de ser planejado para uma interação amigável para com o jogador.

7 RASTREABILIDADE

A rastreabilidade do sistema Hefesto pode ser vista a seguir em diferentes contextos comparativos:

7.1 CASOS DE USO VS. REQUISITOS FUNCIONAIS

Figura 13 - Rastreabilidade dos casos de uso vs. requisitos funcionais. R = O caso de uso da coluna está relacionado com o requisito da linha.

	CSU001	CSU002	CSU003	CSU004
RF001	R		R	
RF002	R		R	
RF003			R	
RF004			R	
RF005		R		
RF006				R
RF007		R		
RF008				
RF009				
RF010				

Fonte: própria, 2022.

7.2 CASOS DE USO VS. REQUISITOS NÃO FUNCIONAIS

Figura 14 - Rastreabilidade dos casos de uso vs. requisitos não funcionais. R = O caso de uso da coluna está relacionado com o requisito da linha.

	CSU001	CSU002	CSU003	CSU004
RNF001				
RNF002	R	R	R	R
RNF003	R		R	
RNF004	R		R	
RNF005				
RNF006		R		R
RNF007	R		R	
RNF008		R		
RNF009				
RNF010		R		
RNF011	R		R	

Fonte: própria, 2022.

7.3 REQUISITOS FUNCIONAIS VS. REQUISITOS FUNCIONAIS

Figura 15 - Rastreabilidade dos requisitos funcionais vs. requisitos funcionais. R = o requisito da coluna está relacionado com o requisito da linha e D = o requisito da coluna é dependente do requisito da linha

	RF001	RF002	RF003	RF004	RF005	RF006	RF007	RF008	RF009	RF010
RF001		D	R				R			
RF002	D		R	D			R			
RF003	R	R		D		D	R		D	
RF004	R	R								
RF005							R			
RF006	R	R						D	D	D
RF007	R	R								
RF008										
RF009										
RF010		R					R			

Fonte: própria, 2022.

7.4 REQUISITOS NÃO FUNCIONAIS VS. REQUISITOS NÃO FUNCIONAIS

Figura 16 - Rastreabilidade dos requisitos não funcionais vs. requisitos não funcionais. R = o requisito da coluna está relacionado com o requisito da linha e D = o requisito da coluna é dependente do requisito da linha

	RNF001	RNF002	RNF003	RNF004	RNF005	RNF006	RNF007	RNF008	RNF009	RNF010	RNF011
RNF001					D			D	R		
RNF002			R	R			R				R
RNF003		R									
RNF004		R									
RNF005	D										
RNF006											
RNF007		R									
RNF008	R										
RNF009	R									R	
RNF010									R		
RNF011		R									

Fonte: própria, 2022.

7.5 REQUISITOS FUNCIONAIS VS. REQUISITOS NÃO FUNCIONAIS

Figura 17 - Rastreabilidade requisitos funcionais vs. requisitos não funcionais. A = requisito da linha afeta o requisito da coluna e R = requisito da linha tem algum relacionamento com o requisito da coluna.

	RNF001	RNF002	RNF003	RNF004	RNF005	RNF006	RNF007	RNF008	RNF009	RNF010	RNF011
RF001											
RF002				A			A				
RF003				A							
RF004				A							
RF005					A						
RF006				A			A				
RF007						A					
RF008											
RF009											
RF010	R						A				R

Fonte: própria, 2022.

8 BIBLIOGRAFIA

ROUSY, DANIELLE. **Engenharia de Requisitos (ER) N07 - Requisitos**. 2022. Slides.

Disponível em:

<<https://sig-arq.ufpb.br/arquivos/202200913910f64081999c62bb9959d3f/ers-n07-requisitos.pdf>>. Acesso em: 18 de junho de 2022.

Bibliografia utilizada pela professora durante a disciplina:

PLEEGER, SHARILAWRENCE. **Engenharia de Software: Teoria e Prática**. 2a.. Prentice Hall. 2004.

RESSMAN, ROGER S; TRAVIESO, MÔNICA MARIA G. **Engenharia de software**. 5.ed. Rio de Janeiro: McGraw-Hill, 2002. 843p : il. ISBN: 8586804258.

BROCKMAN, JAY B; BIASI, RONALDO SÉRGIO DE. **Introdução à Engenharia: Modelagem e Solução de Problemas**. Rio de Janeiro: LTC, 2013. 294p. ISBN: 9788521617266.

9 APÊNDICES

9.1 PROCESSO DE DESENVOLVIMENTO

O desenvolvimento da engenharia de requisitos e prototipação do jogo seguiu uma variação do método ágil SCRUM.

Organização da equipe:

- Diego (Líder) - Desenvolvedor do principal, projetista e idealista do jogo;
- Leonardo (Product Owner) - Gamedesigner e co-desenvolvedor;
- Natércio (Scrum master) - Gamedesigner e co-desenvolvedor;
- Marcos (Developer) - Gamedesigner e co-desenvolvedor.

Nas primeiras semanas, foram realizadas sessões de brainstorming e matriz CSD (Certezas, Suposições e Dúvidas) para definir o conceito do jogo. Logo após isso, foram adotadas reuniões em frequência semanal para planejar a execução das etapas de elicitação da documentação de requisitos.

Uma vez definida uma versão inicial dos requisitos depois de várias interações semanais, foi possível documentar os casos de uso, as funcionalidades principais do jogo e começar a prototipar. Baseado nos casos de uso, em uma sprint mais intensa no desenvolvimento, foi construída uma prototipação de baixa fidelidade e um primeiro protótipo da mecânica de drag and drop na engine Unity.

Finalmente, com essa base de Hefesto montada, o jogo estava pronto para uma primeira apresentação em público visando demonstrar a sua importância e diferencial.

9.2 - PLANO DE ELICITAÇÃO

9.2.1 Metodologia

A técnica utilizada foi a de entrevista com questionários, tendo como stakeholders os discentes das disciplinas de Linguagem de Programação I e Laboratório de Linguagem de Programação I.

Devido à grande quantidade de alunos a entrevista dos discentes foi realizada através de um Formulário Google. As principais informações a serem levantadas neste questionário dizem respeito a quais as principais dificuldades enfrentadas pelos discentes concernentes ao aprendizado dos conceitos da orientação a objetos, quão convidativo é o ambiente de aprendizado atual para pessoas que estão tendo seu primeiro contato com conceitos de programação e quais esforços já estão sendo feitos e/ou planejados para auxiliar esses discentes.

9.2.2 Cronograma

O plano de execução da elicitação de requisitos foi elaborado para ter a duração de oito dias, sendo utilizado o primeiro dia para a elaboração em equipe das perguntas dos questionários, cinco dias para as entrevistas com stakeholders e o último dia para a apuração das respostas e definição da conclusão dos dados adquiridos.

9.2.3 Resultados

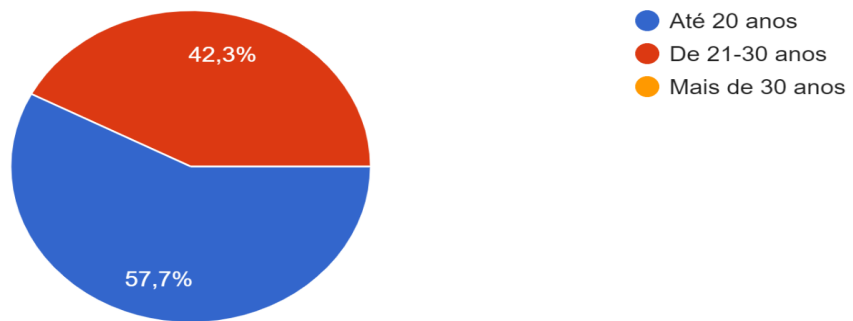
Em relação ao questionário, têm-se os seguintes resultados observados:

- Perfil do público alvo (Figura 18):

Figura 18 - Gráficos relacionados ao perfil do público alvo.

Idade

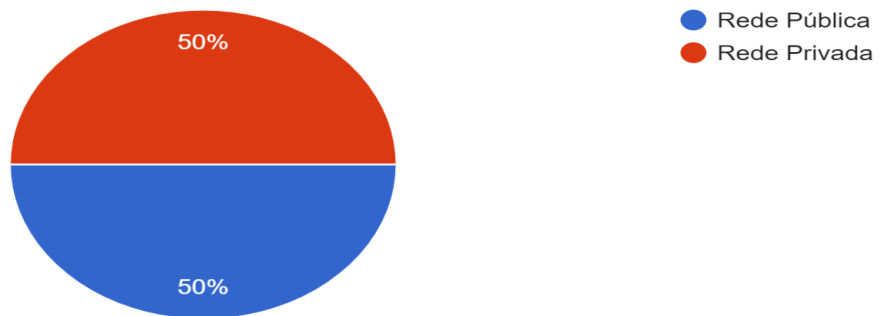
26 respostas



(a)

Em qual tipo de rede de ensino você cursou o ensino médio?

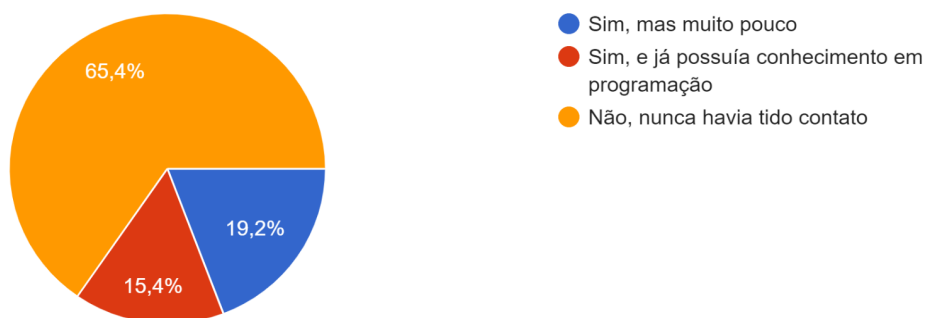
26 respostas



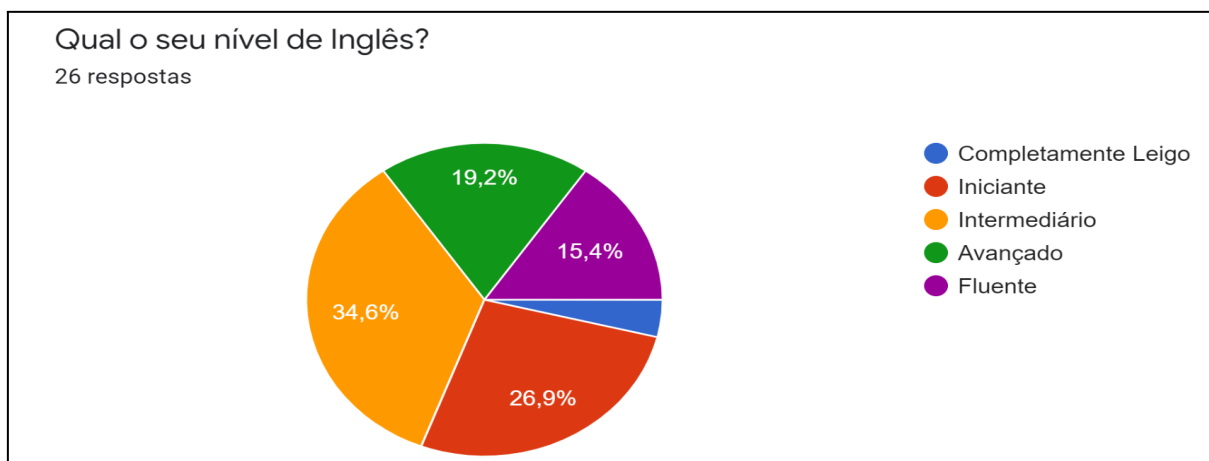
(b)

Antes da graduação você já havia tido contato com logica e/ou linguagens de programação?

26 respostas



(c)

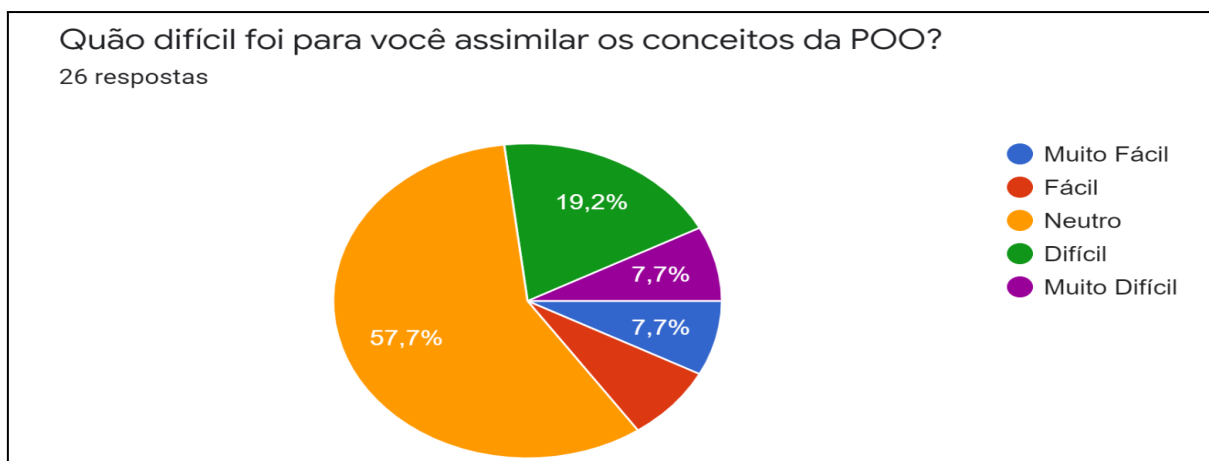


(d)

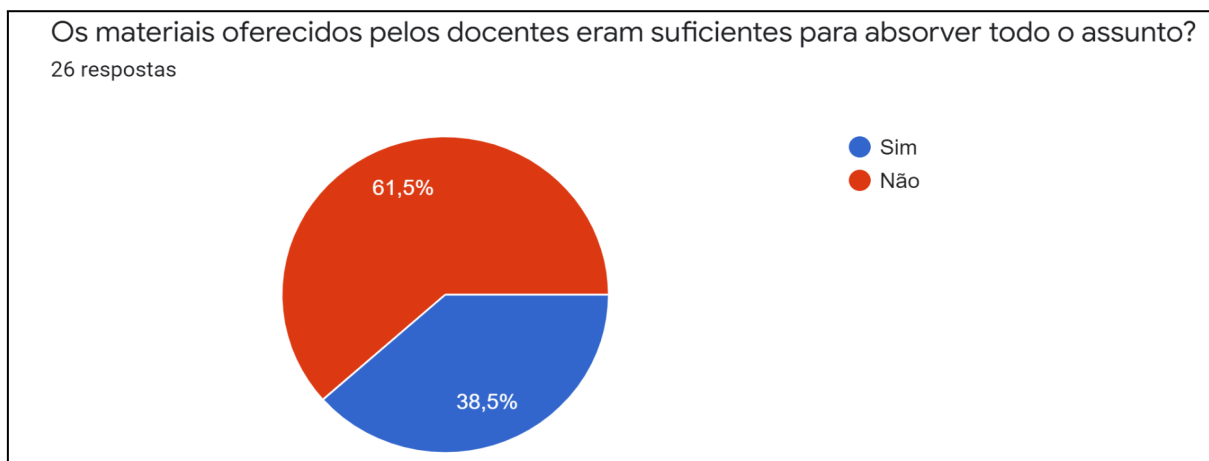
Fonte: própria, 2022.

- Aprendizagem de POO (Figura 19):

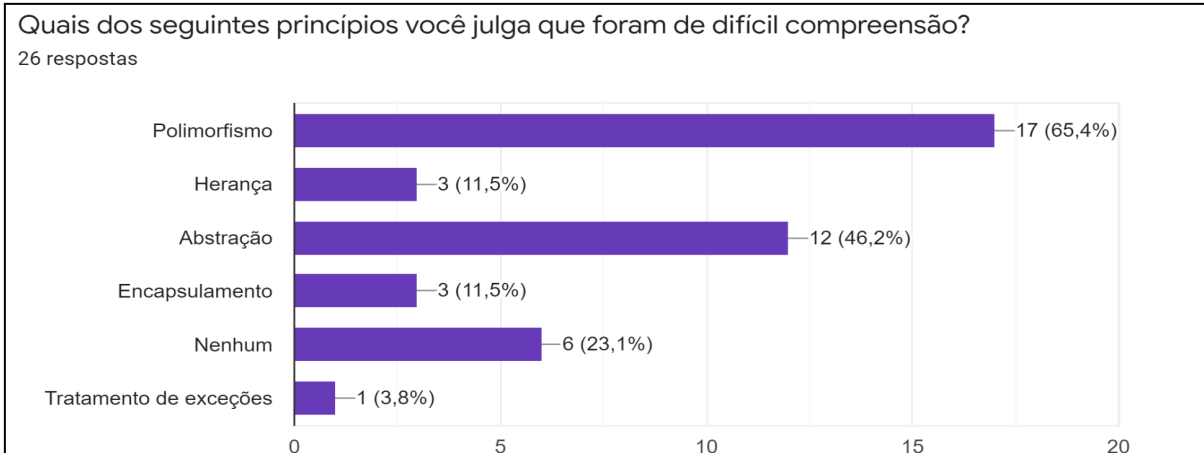
Figura 19 - Gráficos relacionados à aprendizagem de POO na visão do público alvo.



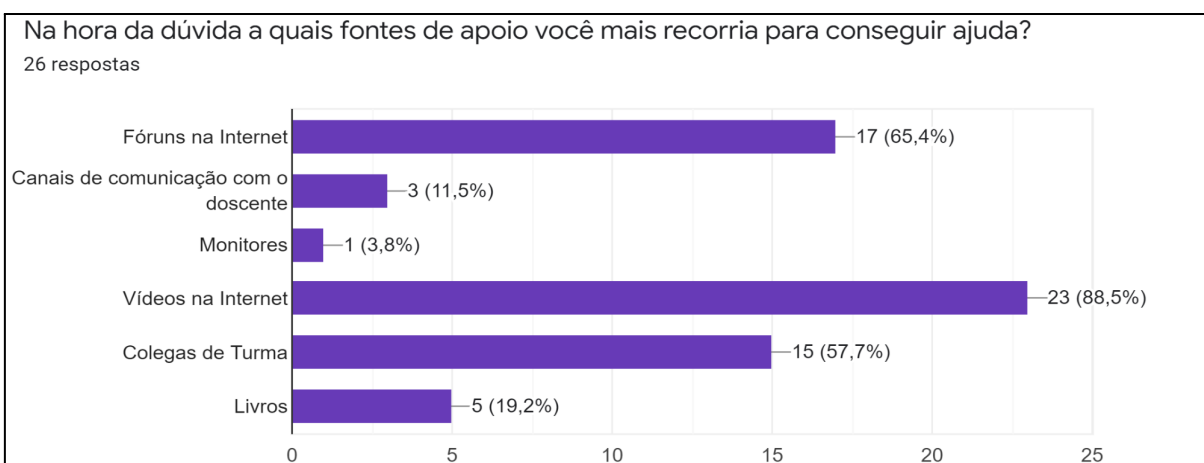
(a)



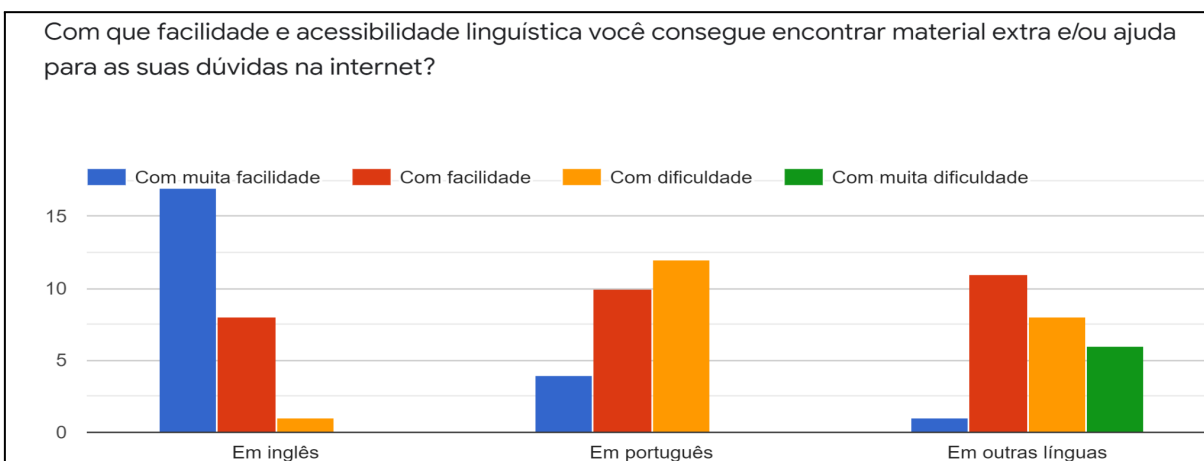
(b)



(c)



(d)



(e)

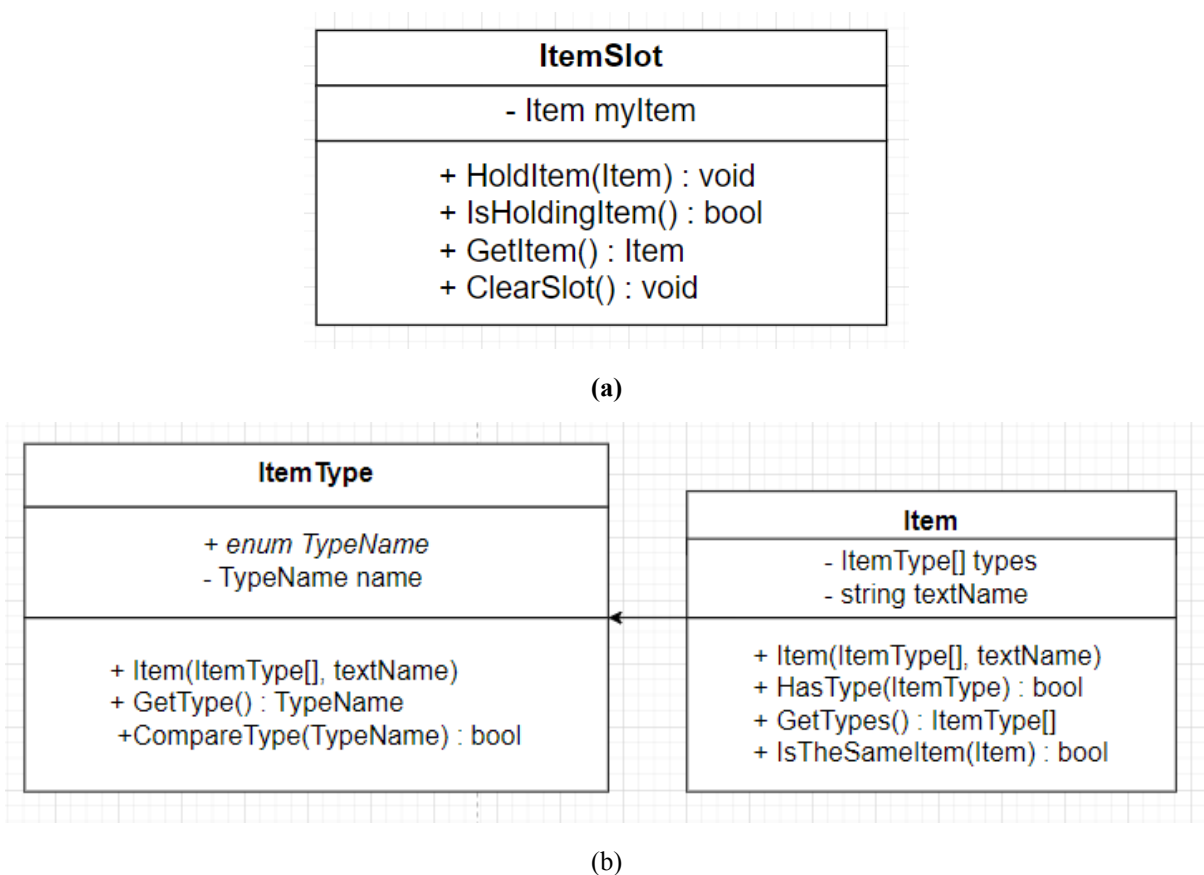
Fonte: própria, 2022.

9.2.4 Conclusão

Com base na pesquisa foi possível concluir algumas das principais dificuldades na aprendizagem dos discentes. Mais da metade dos entrevistados nunca havia tido contato com conceitos de programação ao mesmo tempo em que julgavam possuir um nível de conhecimento da língua inglesa intermediário ou inferior. Preferem buscar auxílio em conteúdo online em detrimento de mecanismos disponibilizados pela instituição, como discentes, monitores e livros, mesmo que a maior parte deste conteúdo esteja disponível em outras línguas, principalmente o inglês. Os dados também apontaram que o conceito de maior dificuldade de compreensão na Orientação a Objeto é o Polimorfismo, seguido pela Abstração.

9.3 DIAGRAMA DE CLASSES

Figura 20 - Diagrama de classes do sistema Hefesto.



Fonte: própria, 2022.