



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

LUCIANO ROVANNI DO NASCIMENTO

**PROG-POLY: JOGO DE TABULEIRO BASEADO NO MONOPOLY PARA AJUDAR
NOS ESTUDOS DE LINGUAGEM DE PROGRAMAÇÃO E ENGENHARIA DE
SOFTWARE**

DISSERTAÇÃO DE MESTRADO

CORNÉLIO PROCÓPIO
2022

LUCIANO ROVANNI DO NASCIMENTO

**PROG-POLY: JOGO DE TABULEIRO BASEADO NO MONOPOLY
PARA AJUDAR NOS ESTUDOS DE LINGUAGEM DE
PROGRAMAÇÃO E ENGENHARIA DE SOFTWARE**

**Prog-poly: Monopoly-based board game to help with
Programming Language and Software Engineering studies**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Mestre em Informática.

Orientador: Prof. Dr. José Augusto Fabri

Co-orientador: Prof. Dr. Rodrigo Henrique Cunha Palacios

CORNÉLIO PROCÓPIO
2022



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



LUCIANO ROVANNI DO NASCIMENTO

PROG-POLY: JOGO DE TABULEIRO BASEADO NO MONOPOLY PARA AJUDAR NOS ESTUDOS DE LINGUAGEM DE PROGRAMAÇÃO E ENGENHARIA DE SOFTWARE

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Informática da Universidade Tecnológica Federal do Paraná (UTFPR).
Área de concentração: Computação Aplicada.

Data de aprovação: 25 de Fevereiro de 2022

Prof Jose Augusto Fabri, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Alexandre L Erario, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Rodrigo De Souza Poletto, Doutorado - Universidade Estadual do Norte do Paraná (Uenp)

Prof Rodrigo Henrique Cunha Palacios, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 10/02/2022.

Este trabalho é dedicado às crianças adultas que, quando pequenas, sonharam em se tornar cientistas.

AGRADECIMENTOS

A Deus, razão suprema da minha existência. Aos meus pais Daniel e Luíza, pelo amor, carinho, compreensão e exemplo de vida.

A minha esposa, Thatiana, pelo amor que cultivamos dia a dia. Ao meu filho, Luciano Júnior.

Ao meu irmão, Eduardo Rovanni do Nascimento, (*in memoriam*), além de meu irmão um amigo para todas as horas, sempre sorridente, brincalhão. Ao meu irmão Daniel, pela amizade e atenção dedicadas quando sempre precisei. Aos meus amigos, pessoas tão queridas e especiais, que seria impossível ter feito algo sem eles.

Agradeço aos meus orientadores, orientador Prof. Dr. José Augusto Fabri e co-orientador Prof. Dr. Rodrigo Henrique Cunha Palácios, pelas valiosas contribuições dadas durante todo o processo, pela disponibilidade, sabedoria, ajuda, dicas, orientações em diversos horários, que me guiaram nesta trajetória. A todos os professores do PPGI e da DACOM.

“A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê.”
(Arthur Schopenhauer)

RESUMO

ROVANNI DO NASCIMENTO, Luciano. PROG-POLY: JOGO DE TABULEIRO BASEADO NO MONOPOLY PARA AJUDAR NOS ESTUDOS DE LINGUAGEM DE PROGRAMAÇÃO E ENGENHARIA DE SOFTWARE. 132 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

Observando o cenário de desenvolvimento de software de 2020, constatou-se que apenas 31% dos projetos de software em todo o mundo, foram concluídos com sucesso. Somando a isso, tem-se que o aprendizado de linguagens de programação é considerado difícil pelos alunos do Ensino Superior em Computação. As principais razões para este fato, segundo os alunos, são: abstração, raciocínio lógico, rigor no uso das linguagens de programação. Assim, este trabalho apresenta um jogo de tabuleiro denominado Prog-poly, baseado na mecânica do clássico jogo de tabuleiro Monopólio, para tentar mitigar os fatos relatados pelos alunos e, conseqüentemente, ser utilizado como uma ferramenta de apoio para ajudar os alunos no desenvolvimento das habilidades e competências relacionadas a Introdução à Engenharia de Software (IES) e também para a Introdução Linguagem de Programação C (ILPC). Para pesquisar os trabalhos relacionados foi feita uma Revisão Sistemática da literatura sobre quais jogos foram desenvolvidos para auxiliar nos estudos de Linguagem de Programação (LP) e Engenharia de Software (ES). Também foi desenvolvido um Software de Análise de Desempenho dos Jogadores do Prog-Poly, para os participantes armazenarem os dados das partidas e assim conseguir analisar seu desempenho durante cada uma das partidas. Para validar o jogo de tabuleiro Prog-poly, foi realizado um quase-experimento com alunos do mestrado em ES para comparar sua eficiência com os métodos tradicionais de ensino na aprendizagem de LP, motivação e diversão dos alunos. Após o quase-experimento, os participantes responderam um questionário com perguntas sobre o Prog-poly, com questões sobre qualidade da construção do jogo, mecânica, dificuldade do jogo e sobre o fator sorte. Com a análise das respostas do questionário, obteve-se os seguintes dados: 95% dos alunos responderam que o Prog-poly motiva os estudos de LP, 85% responderam que o jogo ajuda os alunos desenvolver as habilidades e competências relacionadas a LP, 80% consideraram o jogo uma forma de estudo menos cansativa. Essa evidência demonstra que o jogo de tabuleiro pode ajudar nos estudos, além de tornar as aulas mais motivadoras.

Palavras-chave: Ensino. Linguagem de programação. Engenharia de Software. Gameficação. Jogos de Tabuleiro. Monopoly.

ABSTRACT

ROVANNI DO NASCIMENTO, Luciano. Prog-poly: Monopoly-based board game to help with Programming Language and Software Engineering studies. 132 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

Looking at the 2020 software development landscape, it was found that only 31% of software projects worldwide were successfully completed. In addition to this, the learning of programming languages is considered difficult by the students of Higher Education in Computing. The main reasons for this fact, according to the students, are: abstraction, logical reasoning, rigor in the use of programming languages. Thus, this work presents a board game called Prog-poly, based on the mechanics of the classic Monopoly board game, to try to mitigate the facts reported by the students and, consequently, be used as a support tool to help students in the development of skills and competences related to Introduction to Software Engineering (IES) and also to Introduction to C Programming Language (ILPC). In order to search the related works, a Systematic Review of the literature was carried out on which games were developed to assist in the studies of Programming Language (LP) and Software Engineering (ES). A Prog-Poly Player Performance Analysis Software was also developed, for participants to store match data and thus be able to analyze their performance during each match. To validate the Prog-poly board game, a quasi-experiment was carried out with students of the master's degree in ES to compare its efficiency with traditional teaching methods in LP learning, student motivation and fun. After the quasi-experiment, the participants answered a questionnaire with questions about Prog-poly, with questions about the quality of the game's construction, mechanics, game difficulty and the luck factor. With the analysis of the answers to the questionnaire, the following data were obtained: 95% of the students answered that the Prog-poly motivates the studies of LP, 85% answered that the game helps the students to develop the skills and competences related to LP, 80% considered the game a less tiring form of study. This evidence demonstrates that the board game can help in studies, in addition to making classes more motivating.

Keywords: Teaching. Programming language. Software Engineering. Gamification. Board game. Monopoly.

LISTA DE FIGURAS

FIGURA 1	– GAMIFICAÇÃO ENTRE JOGOS E BRINCADEIRA, COMPLETO E PARTES	22
FIGURA 2	– A TAXIONOMIA REVISADA DE BLOOM	29
FIGURA 3	– TABULEIRO DO PROG-POLY	53
FIGURA 4	– PARTE ONDE COMEÇA O JOGO DE TABULEIRO PROG-POLY .	54
FIGURA 5	– CARTAS DAS COMPANHIAS DO PROG-POLY	55
FIGURA 6	– CARTAS DAS EMPRESAS DO PROG-POLY	56
FIGURA 7	– CARTAS DAS EMPRESAS DO PROG-POLY, COM SEUS VALORES	56
FIGURA 8	– CARTAS DE SORTE	57
FIGURA 9	– CARTAS DE REVÉS	57
FIGURA 10	– DINHEIRO DO PROG-POLY	59
FIGURA 11	– DER do Software de Desempenho dos Jogadores do Prog-Poly ...	65
FIGURA 12	– TELA DO MENU PRINCIPAL DO SOFTWARE	65
FIGURA 13	– TELA DE CADASTRO DOS CURSOS	66
FIGURA 14	– TELA DE CADASTRO DOS JOGADORES	66
FIGURA 15	– TELA DE CADASTRO DAS PERGUNTAS	66
FIGURA 16	– TELA DE CADASTRO DAS PARTIDAS	67
FIGURA 17	– TELA DE CADASTRO DAS RESPOSTAS DOS JOGADORES	67
FIGURA 18	– CONFIGURAÇÃO DO AMBIENTE	72
FIGURA 19	– PARTICIPANTES DO QUASE-EXPERIMENTO	73
FIGURA 20	– GRÁFICO COM OS RESULTADOS DAS RESPOSTAS	74
FIGURA 21	– PERGUNTA COM OPÇÕES DE RESPOSTAS	75
FIGURA 22	– GRÁFICO COM OS RESULTADOS DAS RESPOSTAS	75

LISTA DE TABELAS

TABELA 1 – RESOLUÇÃO TRADICIONAL PARA TODOS OS PROJETOS DE SOFTWARE DE 2011 À 2015	60
--	----

LISTA DE QUADROS

QUADRO 1 – QUADRO COMPARATIVA DOS TRABALHOS RELACIONADOS . 49

LISTA DE SIGLAS

ES	Engenharia de Software
LP	Linguagem de Programação
USP	Universidade de São Paulo
IES	Introdução à Engenharia de Software
ILPC	Introdução à Linguagem de Programação C
GBL	<i>Game based learning</i>
APA	Associação Americana de Psicologia
RSL	Revisão Sistemática de Literatura
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
ACM	<i>Association for Computing Machinery</i>
PMBOK	<i>Project Management Body of Knowledge</i>
CC	<i>Creative Commons</i>
SADJP	Software de Análise de Desempenho dos Jogadores do Prog-Poly
INPI	Instituto Nacional da Propriedade Industrial
TCLE	Termo de Consentimento Livre e Esclarecido
UTFPR	Universidade Tecnológica Federal do Paraná
PPGI	Programa de Pós-Graduação em Informática

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS GERAIS	17
1.2	OBJETIVOS ESPECÍFICOS	17
2	GAMIFICAÇÃO	19
2.1	JOGO	19
2.2	<i>GAME BASED LEARNING</i>	21
2.3	GAMIFICAÇÃO	21
2.3.1	JOGOS SÉRIOS	23
2.3.2	<i>DESIGN LÚDICO</i>	23
2.3.3	GAMIFICAÇÃO NOS ESTUDOS	23
2.4	ELEMENTOS DO PROJETO DE JOGOS, UTILIZADOS NO PROG-POLY	26
3	TAXIONOMIA REVISADA DE BLOOM	28
3.1	TAXIONOMIA REVISADA DE BLOOM NO PROG-POLY	30
4	FUNDAMENTAÇÃO TEÓRICA	32
4.1	OBJETIVO DA RSL	32
4.1.1	QUESTÕES DA RSL SOBRE LP	32
4.1.2	QUESTÕES DA RSL SOBRE ES	33
4.2	SELEÇÃO DOS ESTUDOS - CRITÉRIOS DE EXCLUSÃO E INCLUSÃO	34
4.2.1	STRINGS DE BUSCAS	34
4.2.2	CRITÉRIOS DE EXCLUSÃO	34
4.2.3	CRITÉRIOS DE INCLUSÃO	35
4.3	EXTRAÇÃO DOS DADOS	35
4.4	RESULTADOS	36
4.4.1	RESPOSTAS DA RSL SOBRE LP	36
4.4.2	RESPOSTAS DA RSL SOBRE ES	37
4.5	TRABALHOS RELACIONADOS	40
4.5.1	JOGOS NÃO COMPUTACIONAIS	40
4.5.1.1	SOBRE ÁREA DO CONHECIMENTO LP	40
4.5.1.2	SOBRE ÁREA DO CONHECIMENTO ES	40
4.5.2	JOGOS DE TABULEIRO BASEADOS NO MONOPOLY	42
4.5.2.1	SOBRE ÁREA DO CONHECIMENTO LP	42
4.5.2.2	SOBRE ÁREA DO CONHECIMENTO ES	42
4.5.3	JOGOS COMPUTACIONAIS	44
4.5.3.1	SOBRE ÁREA DO CONHECIMENTO LP	44
4.5.3.2	SOBRE ÁREA DO CONHECIMENTO ES	47

4.5.4	COMPARATIVO DOS TRABALHOS RELACIONADOS	48
5	PROG-POLY	50
5.1	FUNCIONAMENTO DO PROG-POLY	51
5.2	PARTES QUE COMPÕEM O PROG-POLYT	51
5.2.1	REGRAS	51
5.2.1.1	INSTRUÇÕES PARA INICIAR UMA PARTIDA DE JOGAR PROG-POLY .	51
5.2.1.2	FUNCIONAMENTO DO JOGO	52
5.2.2	TABULEIRO	52
5.2.3	COLABORADORES	53
5.2.4	COMPANHIAS	54
5.2.5	EMPRESAS	55
5.2.6	FATOR SORTE	56
5.2.6.1	SORTE	57
5.2.6.2	REVÉS	58
5.2.7	PEÕES	58
5.2.8	PRISÃO	58
5.3	NEGOCIAÇÕES NO PROG-POLY	59
5.4	DINHEIRO	59
5.4.1	PERGUNTAS	60
5.4.2	O QUE O JOGO AUXILIA A ESTUDAR	60
5.4.3	COMO TERMINA O JOGO	62
5.4.4	DO QUE O JOGO É COMPOSTO	62
5.5	PROG-POLY ESTÁ DISPONÍVEL GRATUITAMENTE <i>ONLINE</i>	62
6	SOFTWARE DE ANÁLISE DE DESEMPENHO DOS JOGADORES DO PROG-POLY	64
6.1	PLATAFORMA DE DESENVOLVIMENTO	64
6.2	FUNCIONAMENTO DO SOFTWARE	65
6.3	REGISTRO INPI	67
6.4	MANUAL	68
6.5	O SOFTWARE ESTÁ DISPONÍVEL GRATUITAMENTE <i>ONLINE</i>	68
7	MÉTODOS E PROCEDIMENTO PARA VALIDAR PROG-POLY	69
7.1	EXPERIMENTO	69
7.1.1	PROTOCOLO EXPERIMENTAL	69
7.2	QUASE-EXPERIMENTO	70
7.3	HIPÓTESES	71
7.4	PROTOCOLO DO QUASE-EXPERIMENTO	71
7.4.1	TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO	71
7.4.2	CONFIGURAÇÃO DO AMBIENTE	72
7.4.3	FORMAÇÃO DOS PARTICIPANTES	73
7.4.4	QUANTIDADE DE PARTICIPANTES	73
7.4.5	OCORRÊNCIA TEMPORAL	73
7.4.6	COLETA DOS DADOS	74

7.4.7 ANÁLISE DOS RESULTADOS	75
7.4.8 MELHORIAS REALIZADAS APÓS QUASE-EXPERIMENTO	76
8 CONCLUSÃO	78
REFERÊNCIAS	80
Apêndice A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO	86
Apêndice B – PERGUNTAS	89
Apêndice C – REGRAS	106
Apêndice D – MANUAL DO USUÁRIO	110
Anexo A – RESPOSTAS DO QUASE-EXPERIMENTO	119

1 INTRODUÇÃO

A contemporaneidade, mais especificamente os primeiros 20 anos do século 21, têm sido marcado pelo advento das tecnologias de informação e comunicação. Tal cenário possibilitou que diversos artefatos fossem incorporados nos diversos âmbitos da vida humana, sejam estes para o trabalho ou para o lazer. No campo do lazer, surgiram os jogos em formato eletrônico que tem como usuários, crianças, jovens e adultos em todo mundo, que ganharam novos formatos e interfaces a partir do advento da web. Como exemplo, pode-se citar os jogos eletrônicos produzidos que possuem diversos estilos e/ou níveis de dificuldade, abrangendo uma faixa etária extensa em contraste com o passado, em que eram um mercado de nicho.

O aparecimento de smartphones com maior nível de processamento alavancou o crescimento e prática de jogos digitais entre os jovens que antes não possuíam acesso a outras plataformas de jogos como os consoles domésticos ou computadores, devido principalmente ao seu custo mais elevado. Tal cenário estimulou pesquisadores ao redor do mundo analisar, criar e testar as potencialidades dos jogos nos processos de estudo. Nesse contexto, a utilização de meios tecnológicos para o processo de estudo está em crescente expansão por meio da utilização de jogos digitais e de tabuleiros.

Em se tratando de jogos, uma definição mais completa é apresentada no trabalho de Hogle (1996), que define um jogo como uma atividade recreativa cuja finalidade é entreter e distrair, podendo participar uma ou mais pessoas, e que geralmente exigem dos participantes a utilização das habilidades físicas ou mentais, e que também possui um conjunto de regras que os jogadores devem seguir também para conseguir atingir os objetivos. De modo geral, em jogos de tabuleiros ou até mesmo eletrônicos, os participantes podem competir entre si, contra a máquina ou

sozinho, tentando bater seu próprio recorde. Pode também contar com a sorte mediante da imprevisibilidade ou aleatoriedade das jogadas.

Por estes motivos, cada vez mais a gamificação é utilizada no ensino, quando partes de jogos como mecânica e estética são usados em contextos de não jogos, expandindo as formas de estudar com objetivo de proporcionar engajamento e motivação as pessoas (NETO et al., 2015). De acordo com Deterding et al. (2011) a gamificação está sendo utilizada em diversas áreas como saúde, treinamentos e no ensino.

Ao ingressar no ensino superior, a maioria dos alunos opta por cursos que correspondem aos seus interesses e habilidades, e sua formação lhes proporcione competências profissionais que lhes permitam um melhor desempenho no mercado de trabalho (SOARES; CARVALHO, 2017).

Os estudantes que ingressam nos cursos de Computação, estudam diversas disciplinas, uma delas possui o nome Engenharia de Software (ES). Segundo Sommerville (2011), a ES é uma disciplina de engenharia relacionada a todos os aspectos de produção de software. A ES ensina múltiplas etapas do desenvolvimento de software: análise, especificação, projeto, desenvolvimento, testes, implementação e manutenção de software.

A disciplina de ES ensina também sobre os problemas que podem decorrer no desenvolvimento de software, ao analisar os números disponibilizados pelo Wojewoda e Hastie (2015), neste relatório eles relatam os sucessos e os fracassos no desenvolvimento de software, segundo o levantamento realizado entre 2011 à 2015, o ano em que os projetos de software obtiveram melhor resultado de sucesso foi no ano de 2013, com 41% dos softwares desenvolvidos com sucesso. O relatório mais recente Portman (2021) constatou que em 2020 apenas 31% dos projetos de software em todo o mundo, foram concluídos com sucesso.

Somando a isto, ao entrar em cursos de nível superior em computação, que estimulam habilidades computacionais mais avançadas, os alunos enfrentam dificuldades para absorver e aprender conteúdos que envolvam a Linguagem de Programação (LP).

As primeiras LP, "surgiram da evolução da lógica matemática, no qual abstrai conceitos complexos da matemática e podia ser utilizada para resolver problemas específicos"(CRISTIANO et al., 2019).

O estudo de LP não é uma tarefa fácil para a maioria dos alunos, segundo Moreira et al. (2018), LP exigem habilidades dos alunos como apreender raciocínio lógico, interpretação e resolução de problemas. Ainda agregam a este rol de qualidades segundo Cristiano et al. (2019), apreender os conceitos de semântica e sintaxe.

Por isso, segundo Raabe e Silva (2005) muitos alunos encontram dificuldades no aprendizado da LP. Os obstáculos que os alunos encontram ao estudar LP, que não são detectadas pelos professores, resultam em um alto nível de reprovação, conseqüentemente, em desistências dos cursos de Computação.

Bosse e Gerosa (2015) realizaram um estudo na (USP), e constataram que o índice de reprovação nas disciplinas de introdução a programação chegam a 30%, em vários casos este número ultrapassa os 50%, outra informação importante é que 25% dos alunos fizeram 2 ou mais vezes a mesma disciplina, para serem aprovados.

"Os alunos de hoje – do maternal à faculdade – representam as primeiras gerações que cresceram com esta nova tecnologia. Eles passaram a vida inteira cercados e usando computadores, vídeo games, tocadores de música digitais, câmeras de vídeo, telefones celulares, e todos os outros brinquedos e ferramentas da era digital. Em média, um aluno graduado atual passou menos de 5.000 horas de sua vida lendo, mas acima de 10.000 horas jogando vídeo games (sem contar as 20.000 horas assistindo à televisão). Os jogos de computadores, *e-mail*, a *Internet*, os telefones celulares e as mensagens instantâneas são partes integrais de suas vidas."(PRENSKY, 2001).

Os professores e todos os profissionais da educação percebem a desmotivação dos alunos como um grave problema, de dimensões preocupantes pela sua grande ocorrência no estudo, com graves conseqüências na formação dos alunos (BZUNECK, 2004).

"Infelizmente para os nossos professores Imigrantes Digitais, as pessoas sentadas em suas salas cresceram em uma "velocidade rápida" dos vídeo games e MTV. Eles estão acostumados à rapidez do hipertexto, baixar músicas, telefones em seus bolsos, uma biblioteca em seus *laptops*,

mensagens e mensagens instantâneas. Eles estiveram conectados a maior parte ou durante toda suas vidas. Eles têm pouca paciência com palestras, lógica passo-a-passo, e instruções que “ditam o que se fazer” (PRENSKY, 2001).

Somando a isto, os percalços que os docentes enfrentam em sala de aula. De acordo Smith e Strick (2001), isto se dá ao fato de existir cada vez mais condições que levam às distrações em sala de aula, após 21 anos do artigo escrito por Prensky (2001) ocorreu uma grande evolução digital, com o surgimento de sites de *streaming* de vídeos como YouTube, vídeos curtos como TikTok, de filmes como Netflix, redes sociais como Facebook, Instagram, programas de mensagens instantâneas como Whatsapp e Telegram, jogos, tudo isto disponível nos bolsos dos alunos em qualquer telefone celular, também pode ser acessado por meio de de qualquer dispositivo conectado a internet como, *tablets*, *notebooks*, computadores que são de fácil acesso por parte dos alunos. Dessa forma, o estudo de ES e LP precisa ser atrativo aos alunos, no entanto, a maioria dos cursos ainda ensina essas disciplinas de modo tradicional com conteúdos bastante teóricos, exercícios e trabalhos sistematizados, o que pode não ser motivadores para os estudantes (BARNES et al., 2008).

Ao encontro dos problemas e necessidades expostos, uma possível solução para resolvê-los é a utilização dos recursos de gamificação, que em grande parte, são oferecidos em formato de jogos. Segundo Silva et al. (2014) a utilização de jogos tem uma eficiência de 97% nos estudos. Consoante a esse fato, este trabalho visa a criação de um jogo de tabuleiro, em uma versão não computacional, para ser utilizado como uma ferramenta de apoio para ajudar os alunos, no desenvolvimento das habilidades e competências relacionadas a Introdução à Engenharia de Software (IES) e também na Introdução à Linguagem de Programação C (ILPC), que segundo Cristiano et al. (2019) é umas das LP mais conhecidas e utilizadas no desenvolvimento de softwares.

A principal contribuição deste trabalho é a criação de um jogo de tabuleiro para proporcionar aos alunos, maior engajamento e motivação quando estiverem estudando IES e ILPC. Assim, hipotetizamos que o jogo, denominado Prog-poly, permitirá aos alunos que desenvolvam as habilidades e competências relacionadas à IES e ILPC. Esta ferramenta educacional poderá ser aplicável nas disciplinas que

possui o estudo de Engenharia de Software ou Linguagem de Programação C, além de permitir a integração com tecnologias e recursos didáticos já existentes.

1.1 OBJETIVOS GERAIS

Este trabalho visa a criação de um jogo de tabuleiro físico, baseado na mecânica do jogo Monopoly, para ser utilizado como ferramenta para que os alunos desenvolvam as habilidades e competências relacionadas à IES e ILPC. O jogo poderá ser utilizado nas disciplinas de ES e LP, utilizando de gamificação para abordar e apresentar os conceitos e ajudando os alunos a desenvolverem as habilidades e competências relacionadas IES e ILPC, proporcionando aos alunos maior engajamento e motivação na obtenção de conhecimentos.

1.2 OBJETIVOS ESPECÍFICOS

1. Analisar métodos que utilizam gamificação de modo a levantar e criar requisitos para desenvolver o jogo;
2. Identificar os pontos fracos (por meio da compreensão dos enunciados, para formular questões que serão as cartas do jogo) gerando novas contribuições que auxiliem o aprendizado;
3. Realizar uma Revisão Sistemática da literatura sobre os jogos que foram desenvolvidos para auxiliar nos estudos de LP e ES.
4. Desenvolver um jogo de tabuleiro para ajudar os alunos no desenvolvimento de habilidades e competências relacionadas IES e ILPC, no Ensino Superior;
5. Desenvolver um Software de Análise de Desempenho dos Jogadores do Prog-Poly;
6. Avaliar por meio de um quase-experimento o uso dos jogo com alunos;
7. Melhorar o jogo por meio da avaliação obtida no quase-experimento.

O trabalho está organizado em 8 capítulos. O Capítulo 2, Página 19, apresenta a diferença entre Jogos, Brinquedo e Brincadeira, definição de jogo e gamificação e a relação entre os jogos e o processo de obtenção de conhecimento. O Capítulo 3, Página 28, apresenta a Taxionomia revisada de Bloom, assim como a sua aplicabilidade no projeto Prog-poly. O Capítulo 4, Página 32, contém a fundamentação teórica e os trabalhos relacionados com o jogo. O Capítulo 5, Página 50, apresenta o jogo de tabuleiro Prog-poly, os elementos que o compõem e seu funcionamento. O Capítulo 6, Página 64, é apresentado o Software de Análise de Desempenho dos Jogadores do Prog-Poly. No Capítulo 7, Página 69, é explicado os métodos e procedimento para validar Prog-poly, por meio de um quase-experimento e os resultados obtidos com o mesmo. O Capítulo 8, Página 78, contém a conclusão da dissertação. Na Página 85, são apresentadas as referências bibliográficas que auxiliaram no desenvolvimento deste trabalho.

2 GAMIFICAÇÃO

Neste capítulo é apresentado a definição de jogo e a diferença entre Jogos, Brinquedo e Brincadeira também é explicado a relação entre os jogos e gamificação no processo de estudos. Além disso, também é apresentado a definição de gamificação e a relação entre os jogos e o processo de obtenção de conhecimento.

2.1 JOGO

Buscando a palavra jogo no *site* de busca *Google* são encontrados cerca de 257 milhões de resultados¹.

A origem da palavra jogo vem do latim *jocos* e quando procuramos por jogo no dicionário obtemos o seguinte resultado, segundo dicionário Larousse (1999) é:

"Jogo = Ação de jogar; folguedo, brinco, divertimento". Seguem-se alguns exemplos: "jogo de futebol; Jogos Olímpicos; jogo de damas; jogos de azar; jogo de palavras; jogo de empurra"(LAROUSSE, 1999).

Já (HUIZINGA, 2000) define jogo como:

"O jogo é uma atividade ou ocupação voluntária, exercida dentro de certos e determinados limites de tempo e de espaço, segundo regras livremente consentidas, mas absolutamente obrigatórias, dotado de um fim em si mesmo, acompanhado de um sentimento de tensão e de alegria e de uma consciência de ser diferente da vida cotidiana."(HUIZINGA, 2000, p. 24).

Definir jogo não é um trabalho fácil, segundo Kishimoto (2017), cada pessoa pode entender de forma diferente. "Toda denominação pressupõe um

¹Pesquisa realizada no site www.google.com.br no dia 17 de novembro de 2021.

quadro sociocultural transmitido pela linguagem e aplicado ao real. Dessa forma, enquanto fato social, o jogo assume a imagem, o sentido que cada sociedade lhe atribui.”(KISHIMOTO, 2017).

”O segundo aspecto refere-se aos diferentes significados que uma mesma palavra pode assumir ao longo dos tempos. Se pegarmos um dicionário de 50 anos atrás certamente a acepção das palavras jogo, brinquedo e brincadeira estarão impregnadas de uma visão da época. Nos dias de hoje, observamos que há uma clara diferença entre jogo e brinquedo e entre brincadeira e brinquedo.”(BERTOLDO; RUSCHEL, 2011)

Segundo dicionário Larousse (1999) define brinquedo como: ”Brinquedo = objeto destinado a divertir uma criança”.

A definição de brincadeira segundo dicionário Larousse (1999) é:

”Brincadeira = ação de brincar, divertimento. / Gracejo,zombaria. / Festinha entre amigos ou parentes. / Qualquer coisa que se faz por imprudência ou levandade e que custa mais do que se esperava: aquela brincadeira custou-me caro”. jogo de empurra”(LAROUSSE, 1999).

Segundo Kishimoto (2017) jogo possui um sistema de regras e estrutura que define sua modalidade. O xadrez possui regras explícitas, que o diferenciam de outros jogos como damas e dominó. As regras podem ser utilizadas para diferenciar um jogo. O baralho pode ser utilizado para vários jogos como: truco, tranca, 21, buraco e pôquer, o que difere um jogo do outro são as regras.

Já o brinquedo de acordo com Kishimoto (2017) não possui sistemas de regras para sua utilização. ”Uma boneca permite à criança várias formas de brincadeiras, desde a manipulação até a realização de brincadeiras como ”mamãe e filhinha (KISHIMOTO, 2017).

Segundo (BERTOLDO; RUSCHEL, 2011) brincadeira é o ”ato de brincar”.

”Portanto, o ato de brincar é importante, é terapêutico, é prazeroso, e o prazer é ponto fundamental da essência do equilíbrio humano. Logo, podemos dizer que a ludicidade é uma necessidade interior, tanto da criança quanto do adulto. Por conseguinte a necessidade de brincar é inerente ao desenvolvimento. No brincar, quanto mais papéis a criança representar, mais amplia sua expressividade, entendida como uma totalidade. A partir

do brincar ela constrói os conhecimentos através dos papéis que representa, amplia ao mesmo tempo dois vocabulários - o linguístico e o psicomotor - além do ajustamento afetivo emocional que atinge na representação desses papéis.”(BERTOLDO; RUSCHEL, 2011).

2.2 GAME BASED LEARNING

O aprendizado baseado em jogos do inglês *Game based learning* (GBL), segundo Carvalho (2015) é uma metodologia pedagógica que utiliza a criação e utilização de jogos para finalidades didáticas.

Atualmente, jogos são excelentes ferramentas para alcançar objetivos educacionais, os bons jogos proporcionam recreação e são motivacionais ofertando desafios que exigem um nível cognitivo crescente conforme o jogador passa para a próxima etapa Hogle (1996). Em geral, tais jogos são aplicados em ambientes educacionais para serem utilizados como complemento aos recursos didáticos usados nas aulas.

A tendência no crescimento do uso de jogos nos mais diversos níveis de ensino, ampara-se por vezes em justificativas que consideram características essenciais do jogo, os aspectos lúdicos, a motivação e o envolvimento dos estudantes, como apontados nos estudos de Sena et al. (2016) e Santana et al. (2016). Para desenvolver a metodologia, o trabalho de Tsai et al. (2012) apresenta, além do aprendizado dos conceitos propostos, que os jogos favorecem o desenvolvimento cognitivo e social dos jogadores/alunos por meio da solução de problemas e cooperação entre eles.

2.3 GAMIFICAÇÃO

A gamificação do inglês *gamification* que segundo Vianna et al. (2013), o primeiro registro de utilização deste termo foi de Nick Pelling, programador e *game designer* britânico, em 2002. Segundo Deterding et al. (2011) o termo só começou a ser utilizado na literatura científica em 2008, e se tornou amplamente utilizado a partir de 2010.

Para Deterding et al. (2011) a gamificação utiliza elementos de jogos, como, mecânica, *design*, em contextos não-jogos, para desenvolver uma aplicação, um serviço ou produto, que possa ser utilizado de forma motivadora e envolvente. Além disso, segundo Zichermann e Cunningham (2011) a utilização da gamificação podem trazer benefício como engajar usuários para resolver problemas. "O uso dos pensamentos e da mecânica de jogos para engajar os usuários a resolver problemas" (ZICHERMANN; CUNNINGHAM, 2011).

A Figura 1 explica como a Gamificação esta situada entre as duas dimensões jogar/brincar, completo/parcial, sendo utilizada de uma forma completa nos jogos e brinquedos, e em partes na gamificação e nos jogos sérios. Segundo Deterding et al. (2011) a gamificação contrasta com outros conceitos relacionados por meio de das dimensões utilizadas em jogar/brincar e completo/parcial, onde os "jogos sérios", se diferem dos "jogos", pela utilização de partes dos jogos como mecânica, estética, etc. O design lúdico e os brinquedos são diferenciados pela dimensão de brincar/jogar.

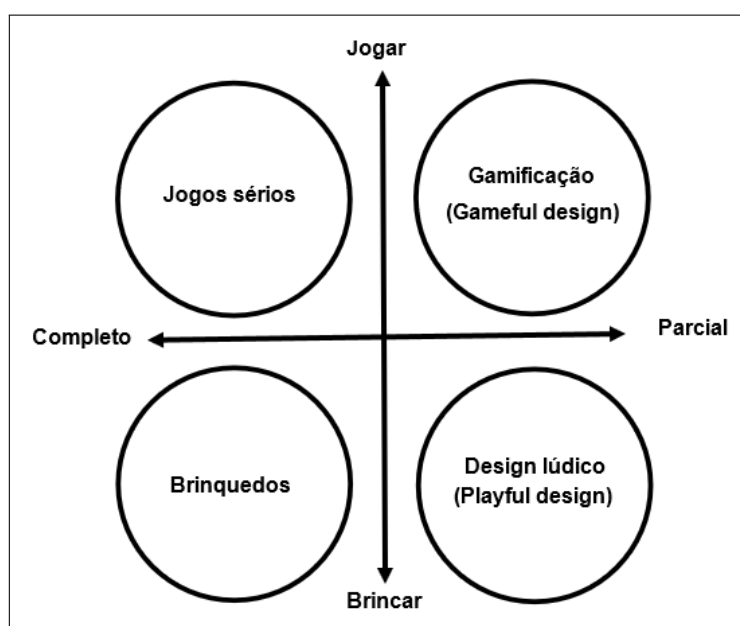


Figura 1: Gamificação entre jogos e brincadeira, completo e partes.

Fonte: Adaptado de (DETERDING et al., 2011)

2.3.1 JOGOS SÉRIOS

Jogos sérios, segundo Borges et al. (2014), são jogos projetados para ambientes não-jogos ou não-entretenimento, principalmente utilizados para fins educacionais. O termo "sério" é usado porque esses jogos possuem como função principal o aprendizado, deixando a diversão em segundo plano. Esses podem ser utilizados em diversos campos, como: educação, economia, saúde, engenharia, indústria, política e militar. Em ambientes criados utilizando os conceitos de jogos sérios, situações reais podem ser simuladas sem custos e riscos eventuais. O principal objetivo desse ambiente de treinamento é transmitir informações ao usuário.

2.3.2 DESIGN LÚDICO

Design lúdico do inglês *Playful design*, segundo Borges et al. (2014), é o uso de partes de jogos como estética, usabilidade, mecânica, utilizados em contextos não-jogos para divertir, causar uma resposta emocional ou chamar a atenção do usuário.

2.3.3 GAMIFICAÇÃO NOS ESTUDOS

A inserção da gamificação nos processos de aprendizado é impulsionada principalmente pelo aumento no engajamento dos estudantes, o que acaba servindo como uma importante medida para sua eficácia. Desta forma, a gamificação utilizada nos estudos almeja contribuir para este processo utilizando os efeitos motivadores dos elementos e técnicas dos jogos digitais e de tabuleiro. Entretanto, não se pode resumir o conceito de gamificação apenas pelos métodos de gratificação aos alunos, o que pode reduzir significativamente seu efeito intencional nos alunos (YMRAN et al., 2018). Na atualidade, a gamificação e os conceitos de *design* de jogos são aplicados, por exemplo, em áreas de treinamento em empresas para bonificações aos colaboradores (YANNAKAKIS; TOGELIUS, 2018).

Prensky (2012) identifica seis elementos considerados chave no projeto de jogos, e quando definidos podem contribuir para a construção de um bom jogo, para ser utilizado nos estudos, sendo eles: metas e objetivos; regras; competição e desafios; interação; história e narrativa; desfecho e *feedback*.

- **Metas e objetivos**

As metas ou objetivos podem ser divididos em duas categorias, de curto prazo e longo prazo.

Os de curto prazo são utilizadas para auxiliar os jogadores a passar para o próximo nível ou etapa do jogo, eles fazem parte dos objetivos intermediários que representam as ações, tomadas de decisões que o jogador toma adaptando sua estratégia para melhorar o seu desempenho e conseguir passar para a próxima etapa.

Já os objetivos/metasp de longo prazo auxiliam o jogador a alcançar os objetivos para concluir o jogo.

- **Regras**

As regras fazem parte do principal componente dos jogos, é por meio delas que os jogadores encontram quais são suas limitações e o que é permitido ou não. Elas devem ser definidas no início do projeto, na fase de criação do jogo, e a partir delas o jogo pode ser desenvolvido.

- **Competição e desafios**

A competição é de fundamental importância nos jogos para torná-los mais competitivos e desafiadores, a disputa é um dos principais elementos para criar certas sensações ou condições nos jogadores, como tensão por uma rivalidade acirrada contra um adversário no jogo.

Já os desafios estão ligados aos objetivos dos jogos e representam as adversidades que os jogadores têm que solucionar.

”Um bom jogo deve ser fácil de aprender e difícil de dominar, isso significa que os desafios de um bom jogo não devem residir em dificuldade para aprender ou mesmo entender seu funcionamento, ou interface” (SILVA et al., 2010).

- **Interação**

Existem duas formas básicas de interação do jogador. A primeira é como o jogador influencia o meio onde ele se encontra. A segunda é como um jogador

pode influenciar os outros jogadores. Segundo Prensky (2007), as interações podem se classificadas das seguintes formas:

- Jogador individual contra o jogo: o jogador não possui nenhuma interação com outro jogador;
- Diversos Jogadores individuais contra o jogo: cada jogador interage somente com o jogo e recebe os resultados dos outros jogadores;
- Jogador contra jogador: cada um interage diretamente contra o seu oponente, como exemplo tem se os jogos de xadrez, damas, luta 1x1;
- Competição parcial: vários jogadores interagindo com um jogador normal;
- Partida multilateral: os participantes interagem com os outros de forma individual;
- Jogo em equipe: os jogadores interagem juntos, com uma mesma meta e cada ato de um dos jogadores, influência sobre os demais;
- Desafio entre grupos: um conjunto de jogadores enfrentar outros.

- **História, narrativa**

A História é uma cadeia de eventos essenciais delineados de forma extensa sem detalhes específicos. A história foca nas dificuldades e objetivos do jogo, assim como nos desafios que os jogadores enfrentaram.

A narrativa é o projeto e a adaptação de como a história é descrita. Ela não é a apresentação do que acontece na história, mas a maneira de gerar uma reação emocional no jogador mediante de causas empolgantes proporcionados pela sua elaboração e medidas inovadoras tomadas.

- **Feedback e desfecho**

Já a *feedback* ou respostas condicionadas no jogo possibilitam a comunicação dos objetivos intermediários ao jogador e o retardo delas pode levar a perda em sensações repassadas pelo jogo. Uma indicação comumente mencionada é mostrar ao jogador se ele/ela está “ganhando” ou “perdendo”.

O frequente *feedback* busca reter a concentração do jogador, de forma que o jogo está interagindo e necessitando de interpretações avaliações e resultados por parte do jogador (PRENSKY, 2007), (ALDRICH, 2005).

O desfecho ou final de jogo envolve um estado, ou condição no qual o jogador se encontrará ao completar os objetivos finais do jogo.

A não delimitação ou falta de compreensão nas condições de fim de jogo podem levar à decepção por parte do jogador.

2.4 ELEMENTOS DO PROJETO DE JOGOS, UTILIZADOS NO PROG-POLY

Os elementos importantes no projeto de jogos, descritos na seção anterior, utilizadas no Prog-poly são:

- **Metas:** Prog-poly possui a meta que os jogadores adquiram e criem empresas de Tecnologia, fazendo que elas prosperem e as tornem o mais rico jogador do tabuleiro.
- **Regras:** Elas são descritas no manual e ensinam passo a passo como utilizar o jogo.
- **Competição:** Está presente em várias partes do jogo, no jogar dos dados, os jogadores almejam cair nas casas que não possuem proprietário para não precisar gastar dinheiro, ou torcem para os outros competidores caiam nas suas propriedades para conseguir a receber o máximo possível aumentando assim sua fortuna no jogo.
- **Desafios:** Eles são encontrados nas perguntas sobre ILPC, somente os jogadores que acerta-las pode adquirir propriedades no jogo.
- **Interação:** As interações no Prog-poly é por uma partida multilateral, os participantes interagem com os outros de forma individual.
- **História:** O contexto do jogo é que jogadores exerceram o papel de programadores que no decorrer da competição adquiram empresas de software. Conforme vão evoluindo os jogadores podem obter colaboradores para suas

empresas, até chegarem a uma quantidade máxima de colaboradores, e seus negócios se tornarem matrizes. Os jogadores devem se esforçarem para que seus negócios cresçam e prosperem gerando lucro, os tornando os competidores mais ricos do tabuleiro. Jogando os estudantes podem se familiarizar com os conceitos estudados em IES, como gestão de suas empresas, colaboradores e do dinheiro.

- **Desfecho:** O jogo termina quando um jogador possuir o maior número de propriedades e dinheiro e os outros jogadores falirem ou desistirem.

Neste capítulo foi apresentada definição de jogo e a diferença entre Jogos, Brinquedo e Brincadeira, também foi apresentado a relação entre os jogos e gamificação no processo de estudos. Apresentou também a definição de gamificação e a relação entre os jogos e o processo de obtenção de conhecimento.

3 TAXIONOMIA REVISADA DE BLOOM

Neste capítulo apresenta a Taxionomia revisada de Bloom, assim como a sua aplicabilidade no projeto Prog-poly.

A Taxionomia de Bloom segundo Flinders e Uhrmacher (2012) é uma metodologia de avaliação da aprendizagem. Desenvolvida em 1948, por uma equipe de psicólogos reunidos em uma convenção da Associação Americana de Psicologia (APA), neste congresso que estava sendo realizado em Boston, levantou-se a importância de criar um plano para desenvolver um modelo teórico que ajudasse a interlocução entre examinadores e motivasse a pesquisa sobre os métodos de avaliação, originando um debate entre os envolvidos no processo de avaliar.

Quem se incumbiu de dirigir este projeto foi Bloom (1956) ao qual a taxionomia recebeu seu nome. A primeira versão associava a aquisição de sabedoria em relação à mudança de desempenho. Em 1990 houve um processo de atualização da taxionomia e, em 2001, foi lançada por Lorin Anderson e sua equipe. A nova versão da Taxionomia revisada de Bloom acrescentou o entendimento a ser obtido e o processo utilizado durante a aquisição desta aprendizagem (KRATHWOHL, 2001).

As características da Taxionomia revisada de Bloom podem ser observadas na Figura 2.

A utilização da Taxionomia revisada de Bloom auxilia na definição dos objetivos na educação, na elaboração consciente da metodologia educacional, para gerar hipóteses de pensar, conduzir e tomar decisões. Essa construção é fruto de um método intimamente relacionado à definição do conteúdo, das ações e atividades, e da estrutura a ser adotada por um certo período.

De acordo com Chang et al. (2010), o processo cognitivo na Taxionomia

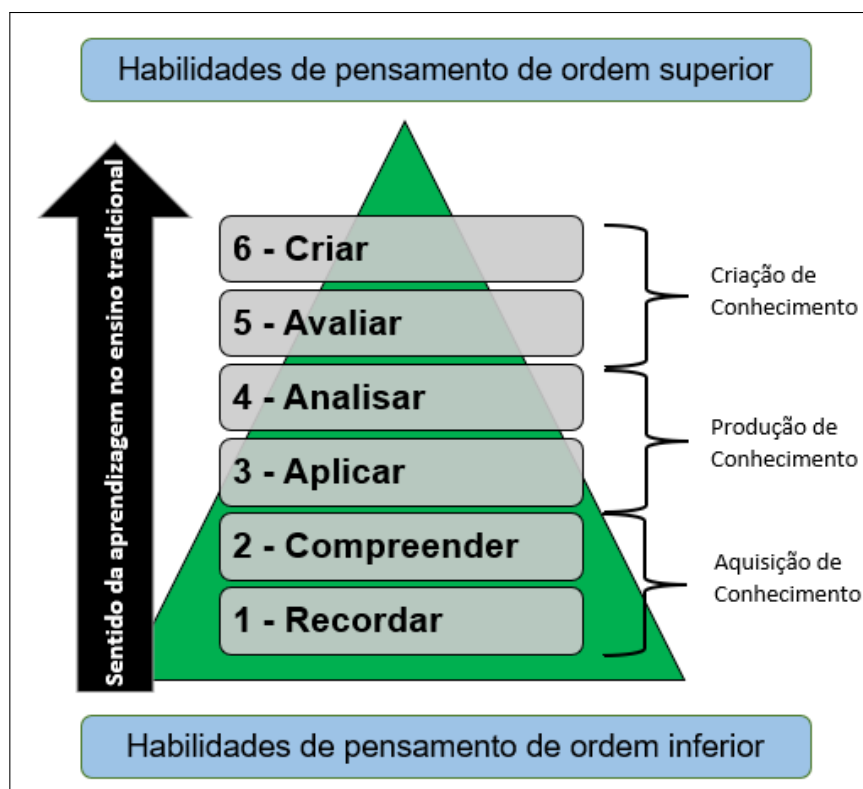


Figura 2: A Taxionomia revisada de Bloom.

Fonte: Adaptado de (SANZOVO; ESCOLA, 2017).

revisada de Bloom possui as seguintes características:

1. Recordar: Significa identificar ideias e conteúdos apresentados anteriormente. Memorizar demanda discernir e distinguir uma certa informação e refazer ou recorda-lá, mais associado à procura por uma dado importante memorizado.
2. Entender: Referente a criar uma ligação entre o novo e o entendimento anteriormente obtido. A informação é conhecida ao passo que o estudante possa reproduzir de acordo com suas “próprias palavras”.
3. Aplicar: Ligado a fazer ou exercer uma ação em um determinado momento em particular e também pode apresentar o uso de uma compreensão de um novo contexto.
4. Analisar: Relativo a separar a informação em componentes importantes e irrelevantes e identificar a ligação presente entre os elementos.

5. Avaliar: Associado a tomada de decisões fundamentadas em métodos e modelos qualitativos e quantitativos em outras palavras de rendimento e eficiência.
6. Criar: Representa apresentar elementos unidos com o propósito de desenvolver uma nova perspectiva, um novo resultado, sistema ou norma, utilizando experiência e capacidades anteriormente adquiridas. Resulta na construção de ideias originais e inéditas, artefatos e regras no processo da compreensão multidisciplinar e da correlação de ideias.

3.1 TAXIONOMIA REVISADA DE BLOOM NO PROG-POLY

Prog-poly possui as seguintes características exibidas na Figura 2, da Taxionomia revisada de Bloom.

- Recordar: Para que os estudantes recordem-se da LP e IES, o jogo fornece itens como o tabuleiro, desenvolvido para os alunos recordarem o ciclo de desenvolvimento de programas, também recordar das situações reais que podem ocorrer em uma empresa de software encontradas nas cartas de sorte e revés.
- Entender: Por meio das regras os jogadores adquirem o entendimento do funcionamento de Prog-poly e, conforme for jogando, os alunos podem criar uma ligação de entendimento anteriormente obtido em cada partida com as habilidades e competências relacionadas ILPC e IES.
- Aplicar: Utilizar de estratégia para saber o que deve adquirir no jogo e quando é importante manter de dinheiro para não ter que vender suas propriedades. Os estudantes podem aplicar as habilidades e conhecimentos adquiridos com o jogo e utilizarem na ILPC e IES.
- Analisar: Por meio da análise do tabuleiro os jogadores perceberam o ciclo que possui o desenvolvimento de um programa, por exemplo, o carregamento de bibliotecas, criação de variáveis, tipos, funções, registros, etc. Somando a isto os jogadores podem usar o SADJP para armazenar os dados das

partidas e exportar para uma planilha do Excel, desta forma os jogadores podem armazenar e analisar sua melhora conforme continuem jogando.

- Avaliar: Utilizando o SADJP os jogadores podem acompanhar seu progresso no jogo, verificando quais questões acertaram ou erraram, identificando o conteúdo que tiveram maior dificuldade para estudar.
- Criar: Utilizando o aprendizado adquirido com o jogo os alunos poderão utilizar estas habilidades e conhecimentos adquiridos para o desenvolvimento de programas na Linguagem C.

Neste capítulo foi apresentado a Taxionomia revisada de Bloom e as características da taxionomia utilizadas no Prog-poly.

4 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo apresenta a fundamentação teórica para desenvolver a dissertação. E explica o protocolo de pesquisa utilizados para encontrar os trabalhos relacionados. Também apresenta os jogos que foram desenvolvidos para auxiliar nos estudos de ES e LP, nas versão não computacionais como os jogos de tabuleiros e cartas e nas versões computacionais como software e websites.

A metodologia utilizada para pesquisar os trabalhos relacionados foi a Revisão Sistemática de Literatura (RSL). Foram desenvolvidas questões da RSL sobre as áreas de ES e LP. Também foram criados critérios de inclusão e exclusão para selecionar os trabalhos, todos estes elementos dá-se o nome de protocolo de pesquisa (KITCHENHAM, 2004).

4.1 OBJETIVO DA RSL

O objetivo deste estudo é identificar e analisar métodos de gamificação a partir do utilização de jogos com o propósito de entender sua aplicação e objetivos de aprendizagem no contexto auxiliar nos estudos de LP e ES, nos últimos 21 anos, no ensino superior?

Para atingir esse objetivo, estabelecemos as seguintes questões de pesquisa:

4.1.1 QUESTÕES DA RSL SOBRE LP

1. Quais jogos foram desenvolvidos para auxiliar nos estudos LP nos últimos 21 anos no ensino superior?

2. Que categorias de jogos foram propostos ou usados para auxiliar nos estudos LP nos últimos 21 anos no ensino superior? Quais são computacionais e quais não são?
3. Quais jogos foram desenvolvidos baseados no Monopoly para auxiliar nos estudos LP nos últimos 21 anos no ensino superior?

4.1.2 QUESTÕES DA RSL SOBRE ES

1. Quais jogos foram desenvolvidos para auxiliar nos estudos ES nos últimos 21 anos no ensino superior?
2. Que categorias de jogos foram propostos ou usados para auxiliar nos estudos ES nos últimos 21 anos no ensino superior? Quais são computacionais e quais não são?
3. Quais jogos foram desenvolvidos baseados no Monopoly para auxiliar nos estudos ES nos últimos 21 anos no ensino superior?

A formulação das questões foram para identificar o cenário de produção científica na área de jogos educacionais no ensino de LP e ES. Mediante este estudo foi verificado as pesquisas na área de jogos voltados à educação, obtendo um prospecto da inserção destes materiais como processo de apoio ao auxiliar nos estudos de LP e ES.

Realizou-se a busca por trabalhos publicados em português e inglês, nas bases de dados:

1. Elsevier;
2. (IEEE) Xplorer;
3. (ACM) Digital library;
4. Scopus;
5. Google Scholar

6. dimensions.ai
7. engineeringvillage
8. MicrosoftResearch
9. Academia.edu
10. ResearchGate
11. DisneyResearch

4.2 SELEÇÃO DOS ESTUDOS - CRITÉRIOS DE EXCLUSÃO E INCLUSÃO

Os artigos selecionados foram publicados no período de 2000 a 2021 e cujos títulos e resumos foram encontrados as palavras das strings de buscas. Em uma primeira análise foram selecionados artigos pesquisando pelo título e pela leitura dos resumos. Esta seleção teve como objetivo responder às questões de pesquisa.

4.2.1 STRINGS DE BUSCAS

As strings de buscas utilizadas foram:

- String de Busca em Inglês: ("teaching"OR "learning"OR "training"OR "education") AND ("Programming Language"OR "Software Engineering") AND ("game"OR "gamification"OR "monopoly").
- String de Busca em Português: ("ensinar"OR "estudo"OR "treinar"OR "educação") AND ("Linguagem de Programação"OR "Engenharia de Software") ("jogo"OR "jogos"OR "gamificação").

4.2.2 CRITÉRIOS DE EXCLUSÃO

1. Trabalhos para auxiliar no estudo ES ou LP, mas não utilizam jogos ou gamificação;
2. Trabalhos duplicados;

3. Trabalhos com publicações incompletas;
4. Trabalhos que abordam jogos ou gamificação, mas não pertencem à área ES ou LP;
5. Trabalhos publicados em anos diferentes do período escolhido;
6. Trabalhos que utilizam jogos ou gamificação, mas não são utilizados para auxiliar nos estudos;
7. Trabalhos que utilizam jogos ou gamificação, utilizados para auxiliar nos estudos de programação ou ES ou LP, para crianças;
8. Trabalhos que utilizam jogos ou gamificação, utilizados para auxiliar nos estudos de ES ou LP, mas são pagos.
9. Trabalhos que utilizam jogos ou gamificação, utilizados para auxiliar nos estudos de ES ou LP, nas versões computacionais descontinuados ou que estão fora do ar por mais de 1 ano.

4.2.3 CRITÉRIOS DE INCLUSÃO

1. Trabalhos com acesso gratuito ou que possuem acesso liberado por meio de do convênio com a UTFPR;
2. Trabalhos que utilizam jogos ou gamificação para auxiliar no estudo de ES ou LP, utilizados no ensino superior?;
3. Trabalhos que utilizam jogos ou gamificação baseados no Monopoly, utilizados nos estudos de ES ou LP, utilizados no ensino superior;
4. Trabalhos que utilizam jogos ou gamificação e estão em evidência sendo utilizados para a estudos de ES ou LP.

4.3 EXTRAÇÃO DOS DADOS

Após a primeira seleção e de realizar a leitura de todos os títulos e resumo foram aplicados os critérios de inclusão e exclusão todos os artigos foram lidos.

Foram selecionados os trabalhos que possuíam relação com este trabalho. Foram sintetizados na Seção 4.5, Trabalhos Relacionados, Página 40, que está dividido em três partes, os trabalhos desenvolvidos utilizando jogos não computacionais, trabalhos utilizando jogos de tabuleiro baseados no Monopoly e jogos computacionais.

4.4 RESULTADOS

Para realizar a RSL foram criadas questões de pesquisa. As respostas encontradas na RSL para essas questões foram divididas em 2 partes, a primeira área do conhecimento de LP e a segunda com base área do conhecimento de ES:

4.4.1 RESPOSTAS DA RSL SOBRE LP

1. Quais jogos foram desenvolvidos para auxiliar nos estudos LP nos últimos 21 anos no ensino superior?
 - 1.1. C-Jump, (SINGH et al., 2007).
 - 1.2. Wu's Castle, (EAGLE; BARNES, 2008).
 - 1.3. *EleMental: The Recurrence*, (CHAFFIN et al., 2009).
 - 1.4. Scratch, (RESNICK et al., 2009).
 - 1.5. *Bomberman game: Game-based digital learning system assists and motivates C programming language learners*, (CHANG et al., 2010).
 - 1.6. Checkio, (GILL, 2013).
 - 1.7. CodinGame, (TACOURI; NAGOWAH, 2021).
 - 1.8. Pex4fun, (TILLMANN et al., 2012).
 - 1.9. CodeCombat, (SEVERGNINI; SOARES, 2019).
 - 1.10. Code Hunt, (TILLMANN et al., 2014).
 - 1.11. Google Grasshopper, (DECUYPERE, 2019).
2. Que categorias de jogos foram propostos ou usados para auxiliar nos estudos LP nos últimos 21 anos no ensino superior? Quais são computacionais e quais não são?

2.1. Nos jogos desenvolvidos em versões não computacionais as categorias utilizadas são:

2.1.1. Jogo de tabuleiro: C-Jump, (SINGH et al., 2007).

2.2. Já nas versões computacionais as categorias dos jogos são:

2.2.1. Software: Wu's Castle, (EAGLE; BARNES, 2008).

2.2.2. Software compilador: *EleMental: The Recurrence*, (CHAFFIN et al., 2009).

2.2.3. Website: Scratch, (RESNICK et al., 2009).

2.2.4. Software: *Bombberman game: Game-based digital learning system assists and motivates C programming language learners*, (CHANG et al., 2010).

2.2.5. Website: Checkio, (GILL, 2013).

2.2.6. Website: CodinGame, (TACOURI; NAGOWAH, 2021).

2.2.7. Website: Pex4fun, (TILLMANN et al., 2012), atualmente não esta funcionando.

2.2.8. Website: CodeCombat, (SEVERGNINI; SOARES, 2019).

2.2.9. Aplicativo da Microsoft Store: Code Hunt, (TILLMANN et al., 2014), atualmente não esta funcionando.

2.2.10. Website: Google Grasshopper, (DECUYPERE, 2019).

3. Quais jogos foram desenvolvidos para auxiliar nos estudos de LP foram baseados no Monopoly utilizados no ensino superior nos últimos 21 anos?

Não foram encontrados jogos baseados no Monopoly para auxiliar nos estudos de Linguagem de programação.

4.4.2 RESPOSTAS DA RSL SOBRE ES

1. Quais jogos foram desenvolvidos para auxiliar nos estudos de ES no ensino superior nos últimos 21 anos?

1.1. THE MIS GAME, (MARTIN, 2000);

1.2. *Problems and Programmers*, (BAKER et al., 2003);

- 1.3. SimuES, (MAGNO et al., 2006);
 - 1.4. SimVBSE, (JAIN; BOEHM, 2006);
 - 1.5. Project-o-poly (PoP), (BUGLIONE, 2007);
 - 1.6. RE-O-Poly, (SMITH; GOTEL, 2008);
 - 1.7. SimuES-W, (MONSALVE et al., 2010);
 - 1.8. *Mission to mars Release or iteration planning*, (KRUCHTEN; KING, 2011);
 - 1.9. DELIVER! (WANGENHEIM et al., 2012);
 - 1.10. RealEstate, (TIWARI et al., 2012);
 - 1.11. GameES, (MEDEIROS; REIS, 2013);
 - 1.12. eRiskGame, (OLIVEIRA et al., 2013);
 - 1.13. *StrateJect: An Interactive Game for Project Management Experiential Learning*, (TELUKUNTA SRIKANTH KUMAR KOTA et al., 2014);
 - 1.14. ARMI, (RAMINGWONG; RAMINGWONG, 2014);
 - 1.15. ARMI 2.0, (SONCHAN; RAMINGWONG, 2015);
 - 1.16. *The monopoly game to teach ERi*c-intentional requirements engineering*, (OLIVEIRA et al., 2015).
2. Que categorias de jogos foram propostos ou usados para auxiliar nos estudos ES nos últimos 21 anos no ensino superior? Quais são computacionais e quais não são?
- 2.1. Nos jogos desenvolvidos em versões não computacionais as tecnologias utilizadas são:
 - 2.1.1. Jogo de tabuleiro, THE MIS GAME, (MARTIN, 2000);
 - 2.1.2. Jogo de cartas, *Problems and Programmers*, (BAKER et al., 2003);
 - 2.1.3. Jogo de cartas, SimuES, (MAGNO et al., 2006);
 - 2.1.4. Jogo de tabuleiro, Project-o-poly (PoP), (BUGLIONE, 2007);
 - 2.1.5. Jogo de tabuleiro, RE-O-Poly, (SMITH; GOTEL, 2008);
 - 2.1.6. Jogo de tabuleiro, *Mission to mars Release or iteration planning*, (KRUCHTEN; KING, 2011);

- 2.1.7. Jogo de tabuleiro, DELIVER! (WANGENHEIM et al., 2012);
- 2.1.8. Jogo de tabuleiro, ARMI, (RAMINGWONG; RAMINGWONG, 2014);
- 2.1.9. Jogo de tabuleiro, *The monopoly game to teach ERI*c-intentional requirements engineering*, (OLIVEIRA et al., 2015).
- 2.1.1. Já nas versões computacionais as tecnologias utilizadas são:
 - 2.1.1.1. Software SimulES-W, (MONSALVE et al., 2010), é a evolução da versão SimulES;
 - 2.1.1.2. Software, SimVBSE, (JAIN; BOEHM, 2006);
 - 2.1.1.3. Software, RealEstate, (TIWARI et al., 2012);
 - 2.1.1.4. Website, GameES, (MEDEIROS; REIS, 2013);
 - 2.1.1.5. Website, eRiskGame, (OLIVEIRA et al., 2013);
 - 2.1.1.6. Software, *StrateJect: An Interactive Game for Project Management Experiential Learning*, (TELUKUNTA SRIKANTH KUMAR KOTA et al., 2014);
 - 2.1.1.7. Software, ARMI 2.0, (SONCHAN; RAMINGWONG, 2015), é a evolução da versão ARMI;
- 3. Quais jogos baseados no Monopoly auxiliar nos estudos ES nos últimos 21 anos no ensino superior?
 - 3.1. Project-o-poly (PoP), (BUGLIONE, 2007);
 - 3.2. RE-O-Poly, (SMITH; GOTEL, 2008);
 - 3.3. *Mission to mars Release or iteration planning*, (KRUCHTEN; KING, 2011);
 - 3.4. *StrateJect: An Interactive Game for Project Management Experiential Learning*, (TELUKUNTA SRIKANTH KUMAR KOTA et al., 2014);
 - 3.5. RealEstate, (TIWARI et al., 2012);
 - 3.6. *The monopoly game to teach ERI*c-intentional requirements engineering*, (OLIVEIRA et al., 2015).

A relação completo dos resultados estão disponíveis nos trabalhos selecionados, disponíveis na Seção 4.5, Página 40. O Quadro 1, Página 48, exibe um comparativo sobre os trabalhos encontrados na RSL.

4.5 TRABALHOS RELACIONADOS

Os jogos educativos segundo Silva et al. (2014) são ferramentas de estudo, eficientes, dado que podem proporcionar uma forma de estudar e motivar os alunos neste processo.

Muitos jogos de tabuleiro para auxiliar no estudo foram desenvolvidos, mas esta RSL buscou especificamente jogos utilizados no estudo de LP e ES, nos últimos 21 anos, utilizados no ensino superior. Conforme a pesquisa realizada, os seguintes resultados foram descobertos:

4.5.1 JOGOS NÃO COMPUTACIONAIS

Jogos analógicos para ajudar no estudo foram encontrados, mas esses não são baseados no monopoly.

4.5.1.1 SOBRE ÁREA DO CONHECIMENTO LP

- **C-Jump**: é um jogo de tabuleiro não computacional, que ensina Fundamentos da Linguagem de Programação. O objetivo do jogo é encontrar a maneira mais eficiente de "esquiar", por uma montanha. Portanto, os jogadores devem tomar decisões com base na sintaxe de programação comum para percorrer um determinado caminho. Movendo-se pelo tabuleiro, inserindo *loops*, ramificando-se sob instruções condicionais e *switch*, os jogadores ganham experiência física de um programa completo. O primeiro jogador a mover todos os esquiadores além da linha de chegada é o vencedor (SINGH et al., 2007).

4.5.1.2 SOBRE ÁREA DO CONHECIMENTO ES

- **ARMI: A risk management incorporation**: é um jogo de cartas, não computacional, desenvolvido para estudar gerenciamento de riscos, sendo uma área importante em Engenharia de software. Os jogadores seguem princípios de gerenciamento de risco e visam completar os ciclos economizando suas despesas (RAMINGWONG; RAMINGWONG, 2014).

- **Problems and Programmers:** é um jogo de cartas educacional, não computacional, direcionado para ajudar a ensinar engenharia de software. Baseia-se na observação de que os alunos, em um curso típico de engenharia de software, ganham pouca experiência prática em questões relativas ao processo de software. O problema subjacente é o tempo: qualquer curso enfrenta a restrição prática de apenas poder envolver os alunos em, no máximo, alguns pequenos projetos de desenvolvimento de software. Este jogo supera essa limitação fornecendo uma simulação do processo de software (BAKER et al., 2003).
- **SimULES:** é jogo educacional de cartas, que ensina Engenharia de Software, inspirado no Problems and Programmers (PnP). Nele pode ser jogado entre 4 e 8 jogadores, o objetivo de SimULES é que os jogadores disputem uma partida para ver quem consegue terminar um projeto de software primeiro. O vencedor será o jogador quem finalizar o projeto primeiro. Por meio de deste jogo, seus participantes, aprendem conceitos de computação e engenharia de software (MAGNO et al., 2006).
- **THE MIS GAME:** é um jogo de tabuleiro, não computacional, faz parte da área de ensino Engenharia de Software. O jogo ensina Gerenciamento de Projetos que simula o desenvolvimento de projeto de software sob uma perspectiva de gerenciamento. E tem como elementos-chave:
 1. Uma placa com tokens representando o progresso;
 2. Artefatos que o jogador pode possuir (por exemplo, dinheiro);
 3. Eventos que acontecem e os dados, que impulsionam o progresso.

Os jogadores recebem um certo orçamento no início sendo obrigados a adquirir e implantar recursos para desenvolver um portfólio de sistemas. Então, o jogador se concentra no desenvolvimento do projeto de software (MARTIN, 2000).
- **DELIVER!:** É um jogo de tabuleiro, não computacional, para ensinar Gestão de Valor Agregado para monitorar e controlar um projeto de software. Ensina Engenharia de Software. Visto que o jogo começa, os pares de jogadores avançam na execução do projeto rolando um dado, além disso, os eventos de

risco podem acontecer. A cada rodada completada todos os pares de jogadores têm que pagar o salário semanal para suas equipes de projeto. Ao terminar uma fase de projeto, os pares de jogadores devem monitorar o progresso do projeto calculando os indicadores de Valor Agregado e podem alterar o plano do projeto (por exemplo, demitindo ou contratando novos membros da equipe). O Vencedor do jogo é o par de jogadores, que primeira entrega o produto ao cliente chegando na entrega do produto de campo sem ficar sem recursos financeiros durante a execução (WANGENHEIM et al., 2012).

4.5.2 JOGOS DE TABULEIRO BASEADOS NO MONOPOLY

Relacionados com o estudos de LP e ES no ensino superior utilizando jogos de tabuleiro baseados no Monopoly nos últimos 21 anos. Foram encontrados apenas jogos que auxiliam nos estudos de ES. Nenhum jogo de tabuleiro baseado no monopoly que auxiliam nos estudos de LP foi encontrado.

4.5.2.1 SOBRE ÁREA DO CONHECIMENTO LP

Não foram encontrados jogos de tabuleiro baseados no Monopoly para auxiliar nos estudos de LP, no ensino superior, nos últimos 21 anos!

4.5.2.2 SOBRE ÁREA DO CONHECIMENTO ES

- **Project-o-poly (PoP)**: É um jogo de tabuleiro, baseado no Monopoly, ensina Engenharia de Software, o objetivo do jogo é obter o maior lucro possível alugando, comprando e vendendo os projetos localizados ao redor do tabuleiro, até se tornar o mais rico Gerente de Projetos (PM) (BUGLIONE, 2007).
- **Mission to mars Release or iteration planning**: é um jogo de tabuleiro, baseado no monopoly, não computacional. Missão a Marte ensina Engenharia de Software, o jogo ilustra o processo de planejamento no desenvolvimento iterativo de software; reúne conceitos como: iteração (*sprint*), *backlog*, *story cards* e *storypoints*, velocidade (produtividade), impacto de defeitos, dívida técnica e riscos. O jogo é um jogo de tabuleiro de baixo custo e estilo Monopoly,

jogado em grupos de 2 a 4 alunos, onde alguns fatores como incerteza na estimativa, velocidade real e ocorrência de defeitos são simulados por um lance de dados adicionadas para estimular a discussão sobre a estratégia a seguir e como mitigar os riscos (KRUCHTEN; KING, 2011).

- ***StrateJect: An Interactive Game for Project Management Experiential Learning***: É um jogo computacional, baseado no monopoly, que ensina técnicas de gerenciamento de projetos e padrões disponíveis no (PMBOK) *Project Management Body of Knowledge*. Os jogadores são gerentes de projeto e monitoram seu desempenho com outros membros, o jogo lida com as interações dos usuários e ensina gerenciamento de projetos (TELUKUNTA SRIKANTH KUMAR KOTA et al., 2014).
- ***The monopoly game to teach ERI*c-intentional requirements engineering***: É um jogo de tabuleiro, não computacional, baseado no monopoly. Este jogo ensina a elicitaco de requisitos. Utilizando o Mtodo "ERi*c" que ajuda em todo o processo de elicitaco de requisitos e tambm a reengenharia (OLIVEIRA et al., 2015).
- ***RealEstate: Selecting requirement elicitation techniques for software projects***: Este jogo de computacional, baseado no monopoly, chamado **RealEstate** ensina elicitaco de requisitos. Ele fornece vrios recursos que podemos ver no jogo Monopoly. Ele pode ser utilizado por apenas um usurio (TIWARI et al., 2012).
- ***RE-O-Poly: Gameplay to Introduce and Reinforce Requirements Engineering Practices***: Este jogo de tabuleiro, no computacional, baseado no monopoly, para apresentar a estudantes e organizaoes as boas prticas de Engenharia de Requisitos (ER). Apresentar aos estudantes a conscientizao e a adoo de prticas de ER podem ser facilitadas por meio de de uma jogabilidade simples, de baixo custo e criativa como parte de um programa educacional ou de treinamento (SMITH; GOTEL, 2008).

4.5.3 JOGOS COMPUTACIONAIS

A maioria dos jogos de tabuleiro para auxiliar no estudo, encontrados nos últimos 21 anos estão relacionados com Engenharia de Software, mas para ajudar no estudo de programação em sua maioria os jogos são computacionais.

4.5.3.1 SOBRE ÁREA DO CONHECIMENTO LP

- **Wu's Castle:** É um jogo computacional, que ensina LP, por meio de matrizes e loops. Os jogadores interagem de duas maneiras: manipulando *arrays* alterando os parâmetros do *loop* e caminhando fisicamente pelo personagem por meio da execução do *loop*. Depois de uma introdução à história e interface do jogo, o jogador percorre uma representação visual de *loops for*, *while* e *do-while*. O jogador então manipula um *array* unidimensional definindo os parâmetros em um *loop for*. O jogador repete esse processo com uma análise de *loop* aninhada e manipulação de matriz bidimensional (EAGLE; BARNES, 2008).
- **EleMental: The Recurrence:** É um jogo computacional que ensina LP. O jogo é um compilador que permite os alunos escrever código e realizar visualizações interativas para aprender sobre recursão por meio da busca em profundidade de uma árvore binária. É um jogo de Mata-Mata colocando os jogadores um contra o outro em uma batalha que leva tudo. Os jogadores podem manipular o terreno, o vento, a água e as temperaturas enquanto lutam pela supremacia contra seus oponentes aplicando a programação. Jogo de estilo 3D FPS que roda tanto no PC quanto no Xbox 360 (CHAFFIN et al., 2009).
- **Bomberman game: Game-based digital learning system assists and motivates C programming language learners:** É um jogo computacional baseado no jogo Bomberman para ensinar a linguagem de programação C (CHANG et al., 2010).
- **Checkio:** É um jogo computacional, disponível por meio de website. Onde os jogadores, participam de um mundo criado pela comunidade de programadores, onde os programadores podem se reunir para desenvolver novos níveis para outros testarem. O jogo é gratuito e ensina Python e TypeScript. O jogo começa

com missões simples e conforme o jogador vai aprendendo o nível de dificuldade vai aumentando. O público alvo não é para leigos em programação, e para pessoas que já possuem conhecimento de programação e querem apreender Python e TypeScript (GILL, 2013).

- **CodeCombat:** É um jogo computacional, disponível por meio de website. Categorizado como jogo sério de aventura educacional voltado para o aprendizado de LP de computadores. Ao contrário de plataformas como o Scratch e Code.org, que geram código arrastando e soltando blocos, os programas feitos no CodeCombat são escritos pelos alunos em linguagens de programação reais como Javascript, Python, C++, CoffeeScript, Java, como geralmente é um ambiente de programação real (SEVERGNINI; SOARES, 2019).
- **CodinGame:** É um jogo computacional, que ensina LP *online*, adequado para todas as faixas etárias. Disponível em 26 linguagens de programação: Bash, C, C++, C#, Clojure, D, Dart, F#, Go, Groovy, Haskell, Java, JavaScript, Kotlin, Lua, Objective-C, OCaml, Pascal, Perl, PHP, Python (v3), Ruby, Rust, Scala, Swift, TypeScript, and Visual Basic .NET. CodinGame tem uma história de fundo que contém informações como iniciar o jogo, formato de saída esperado e nomes de variáveis a serem usados no código para resolvê-lo. Casos de teste também fazem parte do jogo para avaliar com eficiência como a solução funciona em várias configurações de missão e os jogadores podem visualizar como seus códigos se comportam no jogo. Os jogadores também podem interagir e conversar uns com os outros usando o quadro de mensagens. CodinGame possui vários níveis de dificuldade. O nível fácil, ajudam os usuários a aprender conceitos básicos de programação e jogos. A dificuldade média se concentram mais em estruturas de dados complexas. Em jogos com dificuldade difícil, as missões consistem em algoritmos complexos, como localização de caminho e rastreamento de volta. Existe também o nível de dificuldade muito difícil, neste nível exploram-se conceitos ainda mais complexos, como criptografia (TACOURI; NAGOWAH, 2021).
- **Pex4fun:** É um jogo computacional, disponibilizado em uma plataforma *online* de

código fechado, utilizada para ensinar LP C# e F#. O professor pode utilizar em sala de aula para criar especificação em formato de um programa, os alunos implementam a solução de modo a ter o mesmo resultado que o programa elaborado pelo professor, os jogadores analisam os casos de testes gerados e consideram uma técnica baseada em execução simbólica dinâmica para chegar nos resultados, atualmente o Website está fora do ar (TILLMANN et al., 2012).

- **Code Hunt:** É um jogo computacional, *online* cujo objetivo é ensinar Linguagem de programação, o jogo ensina as linguagens Java e C#. Os desafios são separados por setores como: laços de repetição, aritmética ou condicionais, sendo o código base onde o aluno deve implementar a solução. Dentro de cada um destes, há um nivelamento, ou seja, a cada nível a dificuldade vai aumentando. É o sucessor de Pex4fun, disponível na Microsoft Store, mas atualmente não está funcionando (TILLMANN et al., 2014).
- **Scratch:** É um jogo computacional, disponível por meio de de website. Scratch é uma linguagem de programação de computador gratuita. Lançado em 2007, no Media Lab do MIT, pelo grupo *Lifelong Kindergarten*. Utiliza programação visual e multimídia baseado em *Squeak*, "blocos de montar". Em particular, as atividades incentivam a exploração de conceitos-chave de pensamento computacional (sequência, loops, paralelismo, eventos, condicionais, operadores, dados) e práticas-chave de pensamento computacional (experimentar e iterar, testar e depurar, reutilizar e remixar, abstrair e modularizar). Oferece uma *interface* intuitiva e muito fácil de compreender (RESNICK et al., 2009). Segundo Salazar et al. (2015), Scratch pode ser utilizado no ensino superior para introduzir conceitos e motivar os alunos.

"Scratch não é suficiente para aprendizagem de programação em disciplinas de graduação, mas que sua utilização na introdução dos conceitos, em conjunto com linguagens de programação tradicionais, pode aumentar a motivação do aluno e consequentemente favorecer seu aprendizado."(SALAZAR et al., 2015).

- **Google Grasshopper:** É um jogo computacional, disponível por meio de de website. Para ensinar Linguagem de programação JavaScript. Foi lançado em 2017 é um aplicativo para celular desenvolvido na incubadora de projetos

experimentais do Google, a Área 120, por uma equipe de programadores da companhia (DECUYPERE, 2019).

4.5.3.2 SOBRE ÁREA DO CONHECIMENTO ES

- **ARMI 2.0: An online risk management simulation:** Esta é uma versão computadorizada do do jogo ARMI. ARMI 2.0 foi desenvolvido para gerenciamento de riscos, sendo uma área importante em Engenharia de software. O jogo possui melhorias em várias áreas em comparação com a versão anterior, como cálculo automatizado de custos, aleatoriedade controlada, bem como banco de dados de risco aprimorado. (SONCHAN; RAMINGWONG, 2015).
- **eRiskGame: Learning Risk Management in Software Projects with a Serious Game Based on Intelligent Agents and Fuzzy Systems:** O jogo é uma ferramenta de apoio ao ensino em cursos de engenharia de software. Esta é uma versão computacional, utilizada via navegador *web*. O eRiskGame é utilizado no aprendizado de gerenciamento de riscos em projetos de software. Ele simula um ambiente onde muitas empresas, cada uma representada por um jogador que desempenha o papel de gerente de projeto de software, proporciona ao usuário riscos relacionados à atividade real de gerenciamento de projetos de software, como mudanças nos projetos, saída de trabalhadores da equipe, contratação de novos trabalhadores, problemas no desempenho dos trabalhadores, além de apresentar propostas para obter novos recursos no mercado. O sistema é modelado usando agentes inteligentes que operam baseados em regras do sistemas fuzzy (OLIVEIRA et al., 2013).
- **GameES:** é um jogo computacional *online*, multiusuário, ele utiliza o sistema de perguntas e respostas, com objetivo de auxiliar na aprendizagem da Engenharia de Software (MEDEIROS; REIS, 2013).
- **SimuIES-W:** é a evolução do SimuES em sua versão computacional, ensina Engenharia de Software, por meio de um jogo de cartas. Este jogo permite que o aluno encarrega-se de diferentes papéis em um projeto de construção de

software, permitindo que ele experimente as tarefas e decisões usuais em um ambiente de produção de software (MONSALVE et al., 2010).

- **SimVBSE:** É um jogo computacional que ensina o valor da engenharia de software, por meio de do conceito de "fatores críticos de sucesso". O objetivo deste jogo é agregar esse valor por meio de de fundamentos utilizados para estudar os princípios e práticas de engenharia de software existentes (JAIN; BOEHM, 2006).

4.5.4 COMPARATIVO DOS TRABALHOS RELACIONADOS

O Quadro 1, faz um comparativo dos trabalhos relacionados, mostrando quais são: analógicos, computacionais, baseados no Monopoly, qual categoria, por exemplo, se é um jogo de tabuleiro ou de cartas, qual idioma, se auxiliam nos estudos de ES ou LP, e o ano de sua publicação.

Nome	Utilizado nos estudos	Baseado no Monopoly	Jogos não computacionais	Jogos Computacionais	Categoria	Idioma	Ano
<i>THE MIS GAME</i>	Engenharia de Software		X		Tabuleiro	Inglês	2000
<i>Problems and Programmers</i>	Engenharia de Software		X		Cartas	Inglês	2003
<i>SimuES</i>	Engenharia de Software		X		Cartas	Português(BR)	2006
<i>SimVBSE</i>	Engenharia de Software			X	Software	Inglês	2006
<i>Project-o-poly (PoP)</i>	Engenharia de Software	X	X		Tabuleiro	Inglês	2007
<i>C-Jump</i>	Linguagem de programação		X		Tabuleiro	Inglês	2007
<i>Scratch</i>	Linguagem de programação			X	Website	Inglês	2007
<i>RE-O-Poly</i>	Engenharia de Software	X	X		Tabuleiro	Inglês	2008
<i>Wu's Castle</i>	Linguagem de programação			X	Software	Inglês	2008
<i>EleMental: The Recurrence</i>	Linguagem de programação			X	Software compilador	Inglês	2009
<i>SimuES-W</i>	Engenharia de Software			X	Software	Inglês	2010
<i>Bombberman game: Game-based digital learning system assists and motivates C programming language learners</i>	Linguagem de programação			X	Software	Inglês	2010
<i>Mission to mars Release or iteration planning</i>	Engenharia de Software	X	X		Tabuleiro	Inglês	2011
<i>Checkio</i>	Linguagem de programação			X	Website	Inglês	2011
<i>DELIVER!</i>	Engenharia de Software		X		Tabuleiro	Inglês	2012
<i>RealEstate</i>	Engenharia de Software	X		X	Software	Inglês	2012
<i>CodinGame</i>	Linguagem de programação			X	Website	Inglês	2012
<i>Pex4fun</i>	Linguagem de programação			X	Website	Inglês	2012
<i>GameES</i>	Engenharia de Software			X	Website	Português(BR)	2013
<i>eRiskGame</i>	Engenharia de Software			X	Website	Inglês	2013
<i>StrateJect: An Interactive Game for Project Management Experiential Learning</i>	Engenharia de Software	X		X	Software	Inglês	2014
<i>ARMI</i>	Engenharia de Software		X		Tabuleiro	Inglês	2014
<i>CodeCombat</i>	Linguagem de programação			X	Website	Inglês/Português(BR)	2014
<i>Code Hunt</i>	Linguagem de programação			X	<i>Aplicativo da Microsoft Store</i>	Inglês	2014
<i>ARMI 2.0</i>	Engenharia de Software			X	Software	Inglês	2015
<i>The monopoly game to teach ERIS*c-intentional requirements engineering</i>	Engenharia de Software	X	X		Tabuleiro	Inglês	2015
<i>Google Grasshopper</i>	Linguagem de programação			X	Website	Inglês	2017

Quadro 1: Quadro comparativa dos trabalhos relacionados.

Fonte: Autoria própria.

5 PROG-POLY

Neste capítulo apresenta o jogo Prog-poly, suas características, seu funcionamento, os componentes e como foi elaborado a estratégia para ajudar os alunos a desenvolverem as habilidades e competências relacionadas IES e ILPC.

O Jogo de Tabuleiro Prog-poly, foi inspirado na mecânica do jogo Monopoly criado em 1935 por (DARROW, 1935). Varias versões do jogo foram desenvolvidas, como a versão do Star Wars, Harry Potter, Google, etc.

Como demonstrado na seção 4.5.2, página 42, diversos jogos para o aprendizado baseados no Monopoly foram desenvolvidos, em sua maioria para o aprendizado de Engenharia de Software, mas especificamente para o auxiliar no estudo de ILPC nenhuma versão foi encontrada, Prog-poly foi desenvolvido para auxiliar no estudo de ILPC.

Com objetivo de ser utilizado como uma ferramenta de estudo para os alunos, permitindo a integração com tecnologias e recursos didáticos existentes, ajudando a estimular cada vez mais os alunos nos estudos, utilizando gamificação para tornar aulas menos cansativas e mais motivadoras.

Somente a mecânica do jogo é baseada no monopoly, o restante foi desenvolvido do zero, como o tabuleiro e as peças utilizadas pelos jogadores. As peças para se locomoverem pelo tabuleiro foram impressas em uma Impressora 3D.

Foram criadas perguntas sobre ILPC para os participantes responderem. Uma regra criada com base nas perguntas é que os jogadores só podem adquirir propriedades no caso de acertarem a resposta. Essas questões foram escolhidas sobre o tema Linguagem de Programação C, e utilizando elas no Prog-poly, os alunos são ajudados a desenvolverem as habilidades e competências relacionadas ILPC.

5.1 FUNCIONAMENTO DO PROG-POLY

Os jogadores exercem o papel de programadores, que devem adquirir as empresas de desenvolvimento de software que estão disponíveis no tabuleiro. A quantidade de participantes que podem participar são de 2 à 6 pessoas, para uma partida com uma duração de 2 à 3 horas. O ideal é jogarem no máximo 4 jogadores. O tempo das partidas esta diretamente, relacionado com a quantidade dos jogadores, quanto maior a quantidade de participantes, maior será a duração da partida. O objetivo do jogo é obter o maior lucro possível alugando, comprando e vendendo as empresas, localizados ao redor do tabuleiro. O jogador precisa responder perguntas sobre ILPC e somente se acertar pode adquirir as propriedades, caso possua dinheiro suficiente. O tabuleiro também possui o fator sorte sendo assim o jogador pode ganhar ou perder dinheiro, avançar ou retroceder dependendo das cartas que tirar, ou revés que retirar, poderá até parar na cadeia por algumas jogadas. Será o vencedor o jogador que conseguir acumular o maior número de propriedades e dinheiro.

5.2 PARTES QUE COMPÕEM O PROG-POLYT

Nas próximas seções será apresentado as partes que compõem o jogo Prog-poly, as regra, o tabuleiro, colaboradores, companhias, empresas, fator sorte, etc.

5.2.1 REGRAS

5.2.1.1 INSTRUÇÕES PARA INICIAR UMA PARTIDA DE JOGAR PROG-POLY

1. Para iniciar o joga é necessário reunir de 2 à 4 pessoas;
2. Posicionar o tabuleiro e as cartas em uma mesa com espaço adequado e cadeiras para todos os participantes;
3. Quando os jogadores estiverem prontos, eles(as) escolhem os seus peões, colocando-os no ponto de partida, para jogarem os 2 dados, quem tirar o maior número começa, os outros jogadores continuarão depois dele seguindo o sentido horário;

4. O jogo deve começar pela casa escrito início, e seguir a seta no sentido horário.
5. É muito importante embaralhar as cartas de Sorte e Revés, e as Perguntas sobre ILPC e coloca-los de cabeça para baixo no local indicado, no centro do tabuleiro;
6. Em seguida, é hora de distribuir o "dinheiro", cada jogador terá: 10 notas de R\$ 1, 10 de R\$ 5, 10 de R\$ 10, 10 de R\$ 20, 5 de R\$ 50, 5 de R\$ 100 e 1 nota de R\$ 500, cada jogador começa com R\$ 1610, o restante volta para o banco, as outras cartas como de colaboradores, empresas, companhias também.
7. Somente uma pessoa pode ser o banqueiro, e se quiser jogar terá que tomar cuidado para não misturar as notas recebidas com as do banco.

5.2.1.2 FUNCIONAMENTO DO JOGO

1. Cada um joga os 2 dados em sua vez, percorre o número de casas determinado pelo resultado e cumpre as ordens da casa em que cair.
2. A cada volta completa, ao passar pelo início ganha R\$ 200 como honorários.
3. Ao cair em uma empresa ou companhia, poderá comprá-los do banqueiro, mas antes precisa responder a uma pergunta, verificar as regras das perguntas abaixo, caso acerte a pergunta pode comprá-la pagando o preço indicado no tabuleiro.

Nesta seção foram mostradas partes das regras, o conteúdo integral pode ser lido no **Apêndice C**, página 110. As regras são essenciais para ensinar aos jogadores o funcionamento e como jogar o Prog-poly, é por meio delas que os jogadores encontram quais suas limitações e o que é permitido ou não no jogo.

5.2.2 TABULEIRO

O tabuleiro, Figura 3, foi desenvolvido pensando no ciclo de estudo de ILPC, as casas do tabuleiro foram dispostas da seguinte forma: compiladores, carregamento e inicialização das bibliotecas, variáveis, tipos de dados, entrada de dados, saída de dados, operadores lógicos, estrutura condicional, estrutura de repetição, funções,

110 EMPRESA VARIÁVEIS NÚMEROS INTEIROS	90 EMPRESA BIBLIOTECAS PERSONALIZADAS	80 EMPRESA BIBLIOTECAS PADRÕES	COMPANHIA DE PROPAGANDAS ONLINE	70 EMPRESA SOFTWARE .EXE	60 EMPRESA DE ANALISE LÉXICA, SINTÁTICA E SEMÂNTICA	INICIO ←
TIPOS DE DADOS PRIMITIVOS	INICIALIZAÇÃO	INICIALIZAÇÃO	120	COMPILADORES	COMPILADORES	RECEBA R\$ 200 QUANDO PASSAR

Figura 4: Parte onde começa o jogo de Tabuleiro Prog-poly.

Fonte: Autoria própria.

O jogo foi criado para fazer uma analogia com empresas reais, e quando eles começam a crescer necessitam de mais colaboradores, eles representam a equipe responsável pelo desenvolvimento do software, assim utilizar os princípios de IEG para montar uma equipe de desenvolvimento.

”O software profissional, o que é usado por alguém além do seu desenvolvedor, é normalmente criado por equipes, em vez de indivíduos. Ele é mantido e alterado durante sua vida.”(SOMMERVILLE, 2011)

Para isso, o jogador necessita ter conquistado um grupo de propriedades da mesma cor. Somente após fazer isto, o jogador poderá, então, acrescentar colaboradores, pagando ao banqueiro os preços indicados nos títulos de propriedades. Os jogadores podem acrescentar no máximo 4 colaboradores por empresa, somente após isto terá a opção de se tornar uma Empresa Matriz. Uma regra importante do jogo é que os jogadores só podem acrescentar colaboradores, após ter um colaborador em cada empresa. O mesmo vale para o segundo, terceiro e quarto colaborador.

Quando cair em empresas ou propriedades de outro dono, os jogadores têm que pagar aluguel e taxas correspondentes. Caso não possua nenhum colaborador só será paga a diária de um colaborador.

5.2.4 COMPANHIAS

As companhias proporcionam aos jogadores os princípios de IEG em relação aos custos indiretos no desenvolvimento de software. Segundo Carlos et al. (2021) os custos indiretos, ”são os gastos com materiais, mão de obra e gastos gerais

que atuam indiretamente na produção dos produtos ou prestação dos serviços.”As companhias do jogo foram elaboradas com base nas companhias de prestação de serviços que comumente são utilizadas pelas empresas no seu dia a dia, como pode ser visto na Figura 5, em que são exibidas duas companhias uma de propaganda *online* e outra de energia elétrica, cada vez que o jogador cai na companhia tem que pagar o valor estipulado, multiplicado pelo número do dado que ele tirou. O Prog-poly possui 7 companhias, são elas: companhia de propaganda *online*, companhia de energia elétrica, companhia provedora de internet, companhia de backup em nuvem, companhia locação de computadores e eletrônicos, companhia de banco de dados, companhia de hospedagem de sites e e-mail.



	
COMPANHIA DE PROPAGANDAS	COMPANHIA DE ENERGIA ELÉTRICA
Taxas a cobrar:	Taxas a cobrar:
Pontos dos dados multiplicados por: 40	Pontos dos dados multiplicados por: 50
HIPOTECA 75	HIPOTECA 100

Figura 5: Cartas das companhias do Prog-poly.

Fonte: Autoria própria.

5.2.5 EMPRESAS

Cada empresa faz parte das casas situadas no tabuleiro. Já o nome das empresas foi criado para os alunos lembrarem da ILPC, os nomes das empresas começando pelos compiladores, carregamento e inicialização das bibliotecas, variáveis, como pode ser visto na Figura 6, e vai até à introdução a orientação a objetos posiciona no final do tabuleiro como mostrado na Figura 3. As empresas são criadas em grupos divididos por categoria, diferenciadas por cores, como mostra a Figura 6.

No jogo somente quando o jogador adquire um grupo de empresas pode começar a adicionar os colaboradores e caso atinja o limite de 4, pode torná-las em empresas matrizes, como pode ser visto na Figura 7. Na parte inferior aparece o custo para adicionar colaboradores e também o valor da hipoteca. A hipoteca é uma opção

90	80
EMPRESA BIBLIOTECAS PERSONALIZADAS	EMPRESA BIBLIOTECAS PADRÕES
INICIALIZAÇÃO	INICIALIZAÇÃO

Figura 6: Cartas das empresas do Prog-poly.

Fonte: Autoria própria.

do jogador tem quando não possui mais dinheiro e necessita pagar algum valor a outro jogador ele pode hipotecar sua propriedade.

Sommerville (2011), descreve em seu livro, sobre quando uma empresa de software começa ela possui baixa expectativas e geralmente começam sem utilizar os métodos de engenharia de software, desenvolvendo software com baixa qualidade e fora de prazo e custos.

”Muitas empresas foram forçadas a desenvolver softwares à medida que seus produtos e serviços evoluíram. Elas não usam métodos de engenharia de software no dia a dia. Conseqüentemente, seu software é frequentemente mais caro e menos confiável do que deveria ser. Precisamos de educação e treinamento em engenharia de software para solucionar esses problemas.”, (SOMMERVILLE, 2011).

INICIALIZAÇÃO		INICIALIZAÇÃO	
EMPRESA BIBLIOTECAS PADRÕES		EMPRESA BIBLIOTECAS PERSONALIZADAS	
DIARIA COLAB.	6	DIARIA COLAB.	9
1 COLABORADOR	12	1 COLABORADOR	18
2 COLABORADORES	24	2 COLABORADORES	36
3 COLABORADORES	48	3 COLABORADORES	72
4 COLABORADORES	72	4 COLABORADORES	108
EMPRESA MATRIZ	108	EMPRESA MATRIZ	162
CADA COLAB.	100	CADA COLAB.	100
HIPOTECA	45	HIPOTECA	68

Figura 7: Cartas das empresas do Prog-poly, com seus valores.

Fonte: Autoria própria.

5.2.6 FATOR SORTE

As cartas de sorte ou revês foram desenvolvidas para proporcionar algumas situações reais que podem acontecer em uma empresa de desenvolvimento de

software. Desta forma os estudantes podem se familiarizar com os conceitos estudados em IES. Segundo Sommerville (2011), problemas como entrega fora de prazo, não conformidade do software entregue, problemas de custos, confiabilidade são características que existem no desenvolvimento de software e isto é utilizado nas cartas de sorte ou revês.

SORTE	SORTE
Tirou primeiro lugar no campeonato de qualidade de Software! Receba 50.	Sua empresa ficou entre as 50 melhores empresas de Software do Brasil para se trabalhar, receba 75.

Figura 8: Cartas de Sorte.

Fonte: Autoria própria.

REVÉS	REVÉS
Recursos do cliente não estarão disponíveis como planejado. Consequência falta de orçamento para honrar compromissos, pague 35.	Um vírus de computador infectou o ambiente de desenvolvimento do projeto. Consequência atrasos no prazo de entrega pague 45.

Figura 9: Cartas de revês.

Fonte: Autoria própria.

5.2.6.1 SORTE

Exemplo de situações que podem ocorrer nas cartas de sorte são: entrega do software dentro do prazo, custo. Caso o jogador retire cartas de sorte como as mostradas na Figura 8, ele ganha dinheiro.

5.2.6.2 REVÉS

Exemplo de situações que podem ocorrer nas cartas de revés são: entrega do software com atraso, excedendo os custos, não conformidade com os desejos dos clientes.

Já a opção de revés o jogador perder dinheiro como a cartas mostradas na Figura 9.

5.2.7 PEÕES

Os peões foram impressos utilizando uma impressora 3D. Os arquivos digitais foram baixados do website Thingiverse¹, e todos com o mesmo categoria de licença de utilização *Creative Commons* (CC) que permitem a cópia e compartilhamento dos arquivos.

Os seguintes arquivos foram baixados e impressos em 3D para ser utilizados como peões:

1. *Eiffel tower*²;
2. *Camaro*³;
3. *Top Hat*⁴;
4. Conjunto de peças para o Monopoly⁵.

5.2.8 PRISÃO

Existem várias cartas da revês, uma delas pode levar até o jogador para a cadeia do jogo por algumas rodadas. Os jogadores podem ir para a prisão no Prog-

¹ Pesquisa realizada no website <https://www.thingiverse.com/> no dia 11 de janeiro de 2019.

² Arquivos baixados no website <https://www.thingiverse.com/thing:912478> no dia 11 de janeiro de 2019.

³ Arquivos baixados no website <https://www.thingiverse.com/thing:589974> no dia 11 de janeiro de 2019.

⁴ Arquivos baixados no website <https://www.thingiverse.com/thing:2453990> no dia 11 de janeiro de 2019.

⁵ Arquivos baixados no website <https://www.thingiverse.com/thing:20349> no dia 11 de janeiro de 2019.

5.4.1 PERGUNTAS

As perguntas são de grande importância no jogo, é por meio delas que os jogadores desenvolvam as habilidades e competências relacionadas a Introdução à Linguagem de Programação C, as perguntas estão no Apêndice B, página 89.

5.4.2 O QUE O JOGO AUXILIA A ESTUDAR

O foco principal do jogo é auxiliar os alunos no desenvolvimento das habilidades e competências relacionadas a IES e ILPC.

As cartas de sorte ou revês proporcionam aos jogadores, situações reais que podem ocorrer em uma empresa de desenvolvimento de software. Ajudando desta forma os estudantes a se familiarizar com os conceitos estudados em IES. Segundo Sommerville (2011) o estudo de engenharia de software é de grande importância! Pois, ensina os problemas que podem decorrer no desenvolvimento de software, ao analisar os números disponibilizados pelo *Standish Group International* e apresentados na tabela 1, Wojewoda e Hastie (2015), é possível constatar que entre 2011 à 2015. O ano em que os projetos de software obtiveram melhor resultado foi de 2013 com 41% dos softwares desenvolvidos com sucesso. O relatório mais recente Portman (2021) constatou que em 2020 apenas 31% dos projetos de software em todo o mundo, foram concluídos com sucesso.

Tabela 1: RESOLUÇÃO TRADICIONAL PARA TODOS OS PROJETOS DE SOFTWARE DE 2011 À 2015

	2011	2012	2013	2014	2015
Sucesso	39%	37%	41%	36%	36%
Mudaram	39%	46%	40%	47%	45%
Falharam	22%	17%	19%	17%	19%

Fonte: *Chaos Report*, 2015, (WOJEWODA; HASTIE, 2015).

"A engenharia de software é, portanto, uma tecnologia de importância crítica para o futuro da humanidade. Devemos continuar a educar engenheiros de software e a desenvolver a disciplina para podermos criar sistemas de software mais complexos. É claro que ainda há problemas com os projetos de software. Às vezes, o software ainda atrasa e custa mais do que o esperado. No entanto, não devemos deixar que esses problemas

ofusquem os verdadeiros sucessos na engenharia de software e os métodos e as tecnologias impressionantes de engenharia de software que foram desenvolvidos.”, (SOMMERVILLE, 2011).

O jogo auxilia os alunos no desenvolvimento das habilidades e competências relacionadas a ILPC

A importância do ensino de introdução a LP é de grande importância, segundo Barbosa et al. (2014).

”As dificuldades enfrentadas pelos alunos nas disciplinas introdutórias de programação causam um grande número de reprovações, ocasionando atraso dos estudos e conseqüentemente aumento na evasão do curso. Os cursos da área de exatas, mais particularmente os cursos de matemática e computação, tem apresentado altos índices de reprovação e evasão.”, (BARBOSA et al., 2014).

Como mostrados nos estudos de Bosse e Gerosa (2015), o índice de reprovação nestas disciplinas pode chegar até 50%.

Prog-poly por meio das perguntas e do tabuleiro, auxiliar os alunos no desenvolvimento das habilidades e competências relacionadas a ILPC.

Somando a isto, o jogo também ajuda os jogadores no desenvolvimento das habilidades e competências sobre as finanças mostrando como é importante fazer uma boa gestão do dinheiro. Planejando e investindo corretamente, como estar com os seus pagamentos em dia no jogo, caso contrario terá que hipotecar ou vender suas propriedades.

O jogador que escolher o papel de banqueiro, exerce a função de líder do grupo, organizando o jogo e fazendo as relações funcionarem mais tranquilamente. Além disso, para jogar os participantes precisam interagir, negociar e aprender a ouvir o outro, Prog-poly auxiliar no desenvolvimento da socialização.

Além de exercitar a estratégia e o pensamento analítico, escolhendo corretamente quais propriedades comprar e vender, a rodada certa para fazer um investimento, escolher sua carteira de empresas e companhias.

5.4.3 COMO TERMINA O JOGO

O jogo se encerra quando restar apenas um jogador e os outros jogadores desistam ou declararem falência. Caso seja necessário encerrar uma partida, somam-se os dinheiros e propriedades, o vencedor é o jogador que possuir mais propriedades e dinheiro.

5.4.4 DO QUE O JOGO É COMPOSTO

Prog-poly foi desenvolvido utilizando vários elementos, e estes são:

- 1 Manual de regras;
- 32 cartas de colaboradores;
- 12 cartas de empresas;
- 12 cartas de empresas matrizes;
- 29 cartas de títulos de propriedades;
- 18 cartas de sorte;
- 18 cartas de revés;
- 6 peões;
- 2 dados de 6 lados;
- 1 tabuleiro;
- Dinheiro: possui 560 notas, sendo 80 notas de cada valor R\$ 1, R\$ 5, R\$ 10, R\$ 20, R\$ 50, R\$ 100, R\$ 500.

5.5 PROG-POLY ESTÁ DISPONÍVEL GRATUITAMENTE *ONLINE*

O jogo é gratuito e está disponível integralmente *online* via *GitHub*⁶. O jogo é protegido pelas leis de direitos autorais CC BY-NC-SA Atribuição-NãoComercial-

⁶Disponível no endereço <https://github.com/rovanni/Prog-poly>

Compartilhalgal e qualquer pessoa pode baixar e utilizar e modificar atribuindo o crédito apropriado ao autor.

"GitHub é um site de codificação social que usa o Git3 como seu controle de revisão distribuído e de sistema de gerenciamento de código-fonte. Ele implementa uma rede social onde os desenvolvedores podem transmitir suas atividades para outras pessoas interessadas e que se inscreveram nelas. Atualmente, o GitHub hospeda mais de três milhões de projetos mantidos por mais de um milhão de desenvolvedores registrados."(THUNG et al., 2013).

Neste capítulo foi apresentado o Jogo Prog-poly, seu funcionamento, as partes que compõem o jogo e como Prog-poly ajuda os alunos a adquirir habilidades e competências relacionadas a IES e ILPC. No próximo capítulo será apresentado o Software de Análise de Desempenho dos Jogadores do Prog-Poly.

6 SOFTWARE DE ANÁLISE DE DESEMPENHO DOS JOGADORES DO PROG-POLY

Neste capítulo apresenta o Software de Análise de Desempenho dos Jogadores do Prog-Poly, (SADJP), a plataforma de desenvolvimento utilizada, seu funcionamento e o registro no INPI.

SADJP foi desenvolvido para os jogadores de Prog-poly armazenarem os dados das partidas e assim conseguir analisar seu desempenho durante cada uma das partidas. Possibilitando que os participantes acompanhem seu progresso no jogo, verificando quais questões acertaram ou erraram, identificando o conteúdo que tiveram maior dificuldade, para estudar. Outra vantagem do SADJP é que os dados podem ser exportados para uma planilha do Excel, desta forma os jogadores podem armazenar e analisar sua melhora conforme continuar jogando.

6.1 PLATAFORMA DE DESENVOLVIMENTO

O Software foi desenvolvido utilizando o Microsoft Access, é um sistema de gerenciamento de banco de dados da Microsoft que faz parte do pacote Microsoft Office Professional. Esta plataforma possibilita a modelagem e o desenvolvimento do banco de dados, e também através dela foi possível desenvolver a interface que será utilizada pelos usuários. Para facilitar o entendimento do projeto do banco de dados do software foi utilizado o modelo entidade relacionamento também conhecido como Diagrama Entidade Relacionamento (DER), desenvolvido por Peter Chen, Chen (1976) com o DER é possível demonstrar como as tabelas, "entidades", se relacionam no banco de dados, possibilitando também identificar os atributos, os campos que são chaves primarias e a cardinalidade entre as tabelas. A Figura 11 mostra o DER do banco de dados.

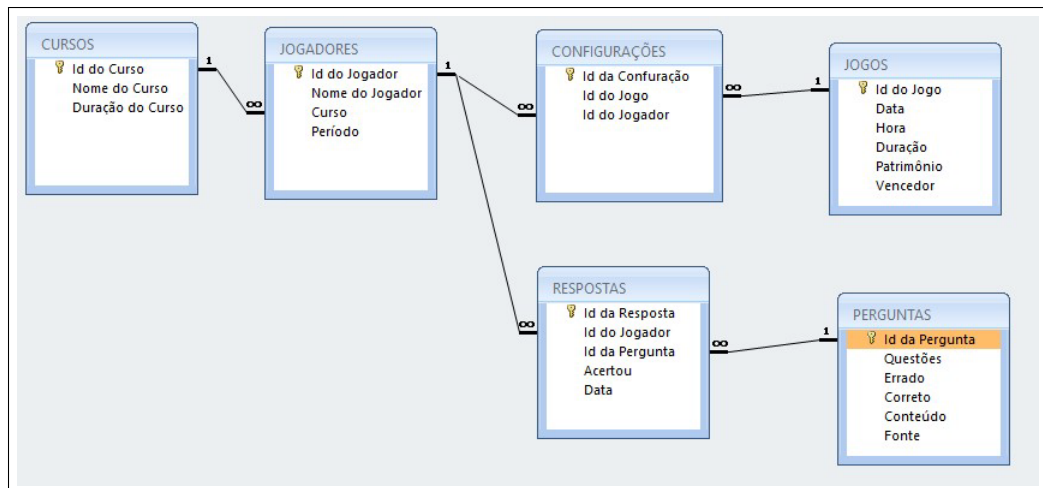


Figura 11: DER do Software de Desempenho dos Jogadores do Prog-Poly

Fonte: Autoria própria.

6.2 FUNCIONAMENTO DO SOFTWARE

Na tela inicial representada pela Figura 12, exibe o menu principal, contendo as opções Cursos, Jogadores, Perguntas, Jogos, Respostas.

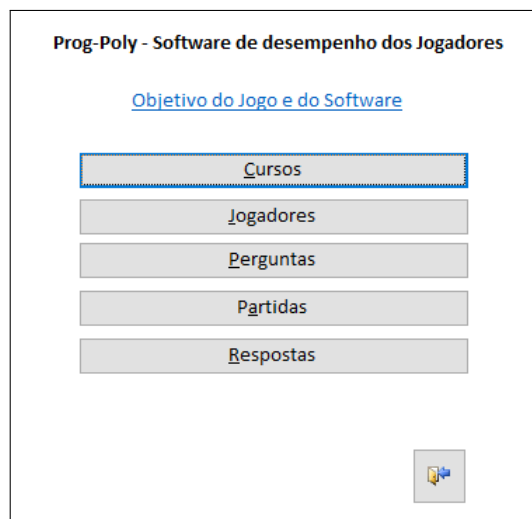


Figura 12: Tela do Menu principal do software.

Fonte: Autoria própria.

Com o software é possível cadastrar: curso, jogador, perguntas do jogo, as partidas, e as respostas dos jogadores.

Na tela representada pela Figura 13 onde é cadastrado o curso do jogador.

Figura 13: Tela de cadastro dos cursos.

Fonte: Autoria própria.

A tela representada pela Figura 14 é cadastrado o jogador.

Figura 14: Tela de cadastro dos jogadores.

Fonte: Autoria própria.

A tela representada pela Figura 15 é utilizada para cadastrar as perguntas do jogo.

Figura 15: Tela de cadastro das perguntas.

Fonte: Autoria própria.

A tela representada pela Figura 16 é utilizada para cadastrar as partidas jogadas.

PARTIDAS

Id do Jogo: 1

Data: 22/08/2021

Hora: 10:50

Duração Prevista/Realizada: 2

Vencedor: Luciano

Patrimônio: R\$ 10.000,00

CONFIGURAÇÕES

Id da Configuração	Id do Jogo	Id do Jogador
3	1	Luciano Rovanni do Nascimento
4	1	Luciano Rovanni do Nascimento
7	1	Luciano Rovanni do Nascimento
* (Novo)	1	

Registro: 1 de 3 Sem Filtro Pesquisar

Figura 16: Tela de cadastro das partidas.

Fonte: Autoria própria.

A tela representada pela Figura 17 é utilizada para cadastrar as respostas feitas pelos jogadores.

RESPOSTAS

Id da Resposta: 2

Id do Jogador: Luciano Rovanni do Nascimento

Id da Pergunta: 5

Acertou:

Figura 17: Tela de cadastro das respostas dos jogadores.

Fonte: Autoria própria.

6.3 REGISTRO INPI

O software foi registrado no Instituto Nacional da Propriedade Industrial (INPI), possui Certificado de Registro de Programa de Computador, processo N^o: BR512021002037-5, expedido em 31/08/2021.

6.4 MANUAL

Foi elaborado um manual para ajudar os usuários na utilização do software, está disponível no **Apêndice D**, paginá 119.

6.5 O SOFTWARE ESTÁ DISPONÍVEL GRATUITAMENTE *ONLINE*

O SADJP e seu manual de funcionamento estão disponíveis *online* via GitHub¹. O SADJP é protegido pelas leis de direitos autorais CC BY-NC-SA Atribuição-NãoComercial-Compartilhalgal e qualquer pessoa pode baixar e utilizar e modificar atribuindo o crédito apropriado ao autor.

Nesse capítulo, foi descrito o Software de Análise de Desempenho dos Jogadores do Prog-Poly, seu banco de dados, os principais elementos. O próximo capítulo serão apresentados os métodos e procedimentos para verificar a viabilidade do Prog-poly.

¹Disponível no endereço <https://github.com/rovanni/Prog-poly/Software>

7 MÉTODOS E PROCEDIMENTO PARA VALIDAR PROG-POLY

Neste capítulo é explicado os métodos e procedimento para validar Prog-poly, também é demonstrado o protocolo do quase-experimento, a hipótese validada, a configuração do ambiente, como foi a aplicação do quase-experimento, de que forma foi realizada a coleta dos dados, a análise dos resultados e as melhorias feitas no jogo.

Para validar Prog-poly foi realizado um quase-experimento, que aconteceu em uma sala de aula reservada da Universidade Tecnológica Federal do Paraná (UTFPR), Campus Cornélio Procópio, oferecendo assim maior nível de controle.

7.1 EXPERIMENTO

O experimento segundo Wohlin et al. (2012) tem o preenchimento do protocolo experimental. Tem como o foco avaliar os níveis de conhecimentos que o jogo pode proporcionar, evoluir o jogo, coletando as informações dos participantes e os ambientes envolvidos. Experimentando e manipulam variáveis potencialmente confusas.

7.1.1 PROTOCOLO EXPERIMENTAL

Para verificar a eficiência do jogo será utilizado o método de pesquisa Experimental, pois segundo Wohlin et al. (2012), Travassos et al. (2002), que possui as seguintes etapas: configuração do ambiente; aplicação do experimento; coleta dos dados; avaliação dos resultados.

O processo de experimentação pode ser dividido nas seguintes atividades Wohlin et al. (2012):

1. Define-se o experimento em termos de problema, objetivo e metas.
2. Planejamento - Determina-se o projeto do experimento e a instrumentação. As possíveis ameaças para validação dos resultados do experimento são avaliadas.
3. Operação - Coletam-se medidas
4. Análise e interpretação - As dados coletados na fase anterior são analisados e avaliados.
5. Apresentação e empacotamento - Os resultados são apresentados e as informações do experimento são disponibilizadas para futuras replicações.

7.2 QUASE-EXPERIMENTO

O quase-experimento foi utilizado para testar e validar o Prog-poly, e verificar se o mesmo ajuda os alunos a desenvolverem habilidades e competências relacionadas a ILPC.

Quase-experimento já são utilizados desde a década de 70, de acordo com Selltiz et al. (1976) o termo foi utilizado a primeira vez no livro Campbell e Stanley (2015) em 1963, uma forma de realizar uma pesquisa com um número menor de variáveis, mas possibilitando observar "o que ocorre, quando ocorre e a quem ocorre e a decidir o que e quando medir" (SELLTIZ et al., 1976).

"Quase-experimentos são delineamentos de pesquisa que não têm distribuição aleatória dos sujeitos pelos tratamentos, nem grupos de controle. Ao invés disso, a comparação entre as condições de tratamento e não-tratamento deve sempre ser feita com grupos não equivalentes ou com os mesmos sujeitos antes do tratamento. Se você não puder distribuir aleatoriamente pessoas ou grupos pelas condições experimentais, perderá a capacidade de controlar o que ocorre a quem. Entretanto, você ainda poderá observar o que ocorre, quando ocorre e a quem ocorre e a decidir o que e quando medir, poderá planejar um dentre vários quase-experimentos. Você terá perdido o controle por não usar a distribuição aleatória, mas ainda assim poderá realizar uma pesquisa e analisar relações de causa-efeito sem um experimento." (SELLTIZ et al., 1976).

"Os estudos quase-experimentais usam o chamado pré-teste. Isto significa que existe uma recolha de dados antes da introdução do tratamento, no

sentido de se traçar o ponto de partida, antes de qualquer tratamento. Em seguida, a experiência real é feita, ou seja, a amostra é sujeita ao tratamento e posteriormente é feito um pós-teste e a respectiva recolha de dados". (SOARES, 2021).

Além disso, o quase-experimento procura obter dos participantes por meio dos *feedbacks* dos jogadores e via o preenchimento do questionário validar o jogo e encontrar possíveis problemas e melhorias para serem realizadas no jogo.

7.3 HIPÓTESES

Para validar a eficiência do Prog-poly foi criada a seguinte hipótese:

- H1 - Prog-poly ajudar os alunos a desenvolverem habilidades e competências relacionadas a ILPC?

O quase-experimento foi realizado com o objetivo de verificar a viabilidade do trabalho e responder a hipótese H1.

Para coletar os dados da hipótese H1, foi desenvolvido um questionário com a seguinte questão: É possível ensinar programação, utilizando o jogo de tabuleiro Prog-poly?

A estratégia de pesquisa quantitativa será utilizada por meio de questionários para obtenção das informações para validar ou refutar a hipótese de pesquisa H1.

7.4 PROTOCOLO DO QUASE-EXPERIMENTO

Nesta seção são descritas as atividades para a execução do quase-experimento e possam proporcionar aos pesquisadores, meios para replicarem o quase-experimento deste trabalho.

7.4.1 TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Os convidados antes de participarem do quase-experimento, foram informados sobre o "TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO"

(TCLE), explicando aos participantes sobre os procedimentos da pesquisa, tratamento de possíveis riscos e desconfortos, benefícios e custos, confidencialidade da pesquisa, esclarecendo sobre a participação voluntária, informando o pesquisador e responsável. Após assinarem o formulário *on-line* TCLE, foi iniciado o quase-experimento sobre o jogo. O TCLE está disponível no **Apêndice A**.

7.4.2 CONFIGURAÇÃO DO AMBIENTE

A configuração do ambiente Figura 18, foi realizada da seguinte forma: o quase-experimento foi realizado em um ambiente acadêmico, na sala de aula da Universidade Tecnológica Federal do Paraná (UTFPR), Campus Cornélio Procópio. Foram separadas 2 mesas para colocar o tabuleiro e as cartas do jogo e 1 mesa para deixar com o dinheiro utilizado pelo jogador que faz papel de banqueiro do jogo. Foram utilizadas 4 cadeiras para cada jogador.



Figura 18: Configuração do ambiente.

Fonte: Autoria própria.

7.4.3 FORMAÇÃO DOS PARTICIPANTES

Alunos do Mestrado em Engenharia de Software, do Programa de Pós-Graduação em Informática (PPGI) da UTFPR. Foi requisitado ao Professor Dr. José Augusto Fabri voluntários para testarem o jogo, e o mesmo propôs fazer com os discentes de sua disciplina de Gerenciamento de Projetos de Software, que fazem parte do Mestrado em Engenharia de Software, participantes do Programa de Pós-Graduação em Informática (PPGI).

7.4.4 QUANTIDADE DE PARTICIPANTES

Participaram do quase-experimento 4 alunos que se voluntariaram para testar o jogo, como pode ser visto na Figura 19.



Figura 19: Participantes do quase-experimento.

Fonte: Autoria própria.

7.4.5 OCORRÊNCIA TEMPORAL

O quase-experimento teve a duração de 120 minutos, e foi realizado no mês de março de 2019.

7.4.6 COLETA DOS DADOS

A coleta dos dados do quase-experimento foi feita utilizando o formulário *online* do *Google Forms* que é uma alternativa amplamente utilizada, facilita a criação dos questionários, à distribuição da pesquisa aos entrevistados a organização e análise dos dados coletados. As vantagens associadas à utilização do formulário eletrônico são várias quando comparado a versão convencional. O *Google Forms* auxilia na análise dos dados, gerando tabelas e gráficos, facilitando assim a tarefa de quantificar os dados. Após o quase-experimento, os participantes responderam um questionário com perguntas sobre o Prog-poly, com questões sobre qualidade da construção do jogo, mecânica, dificuldade do jogo e sobre o fator sorte.

Os dados coletados pelo preenchimento do formulário eletrônico após a conclusão do quase-experimento podem ser visto integralmente no **Anexo A**, página 132.

Após a avaliação das respostas obtidas, os seguintes resultados foram encontrados a partir de perguntas como a mostrada na Figura 20, que quantificam as respostas com níveis.

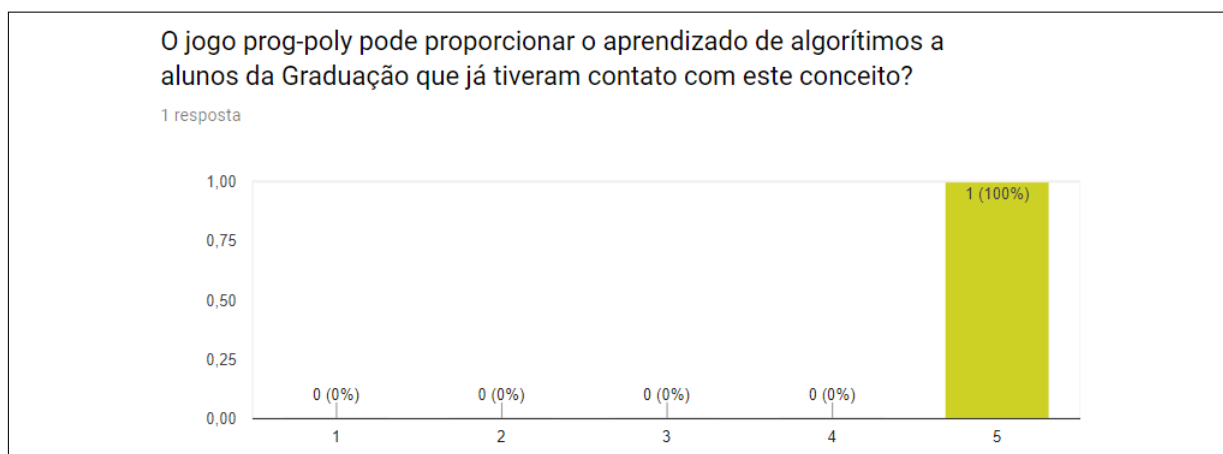


Figura 20: Gráfico com os resultados das respostas.

Fonte: Autoria própria.

Outra questão importante foi é "Com relação ao jogo: Quais dos itens abaixo ajudam a entender conceitos de algoritmos?", esta questão possuía múltiplas escolhas e os resultados podem ser observados no Figura 22 que exibe um gráfico das

O jogo prog-poly pode proporcionar o aprendizado de algoritmos a alunos da ^{*} Graduação que já tiveram contato com este conceito?

1 2 3 4 5

Pouco aprendido Muito aprendido

Figura 21: Pergunta com opções de respostas.

Fonte: Autoria própria.

respostas, onde todos os participantes, ou seja, 100% respondeu que as perguntas ajudam no estudo de algoritmos e 33% que o tabuleiro ensina.

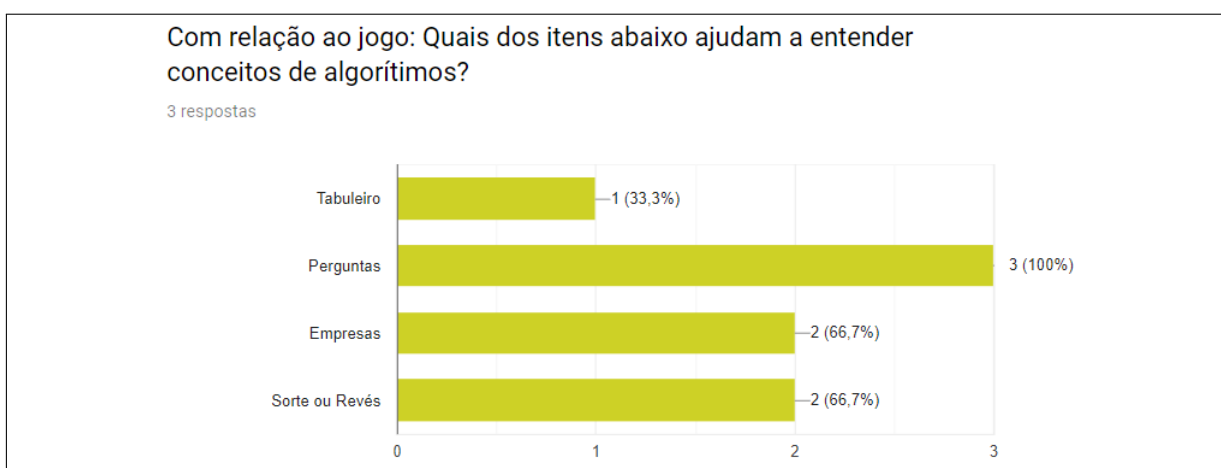


Figura 22: Gráfico com os resultados das respostas.

Fonte: Autoria própria.

7.4.7 ANÁLISE DOS RESULTADOS

Após analisar as respostas do questionário, após o quase-experimento, respondido pelos alunos, como pode ser observado no **Anexo A**, foi possível validar o jogo e observar que Prog-poly ajuda os alunos a desenvolverem as habilidades e competências relacionadas à ILPC.

Os resultados mostraram que as seguintes partes do jogo podem ser melhoradas:

1. Regras

2. Mecânica do jogo
3. Adicionar novas perguntas
4. Adicionar mais cartas de sorte e revés

Todas essas melhorias já foram efetuadas como pode ser observado na seção 7.4.8, e o jogo se encontra pronto para o próximo experimento. Um aspecto importante analisado nos depoimentos dos participantes, foi que o uso do Prog-poly, proporcionou aos alunos, maior engajamento e motivação no estudo de ILPC. Sendo estes fatores de grande relevância para desenvolvimento do jogo e de fundamental importância para ajudar os alunos a desenvolverem as habilidades e competências relacionadas a ILPC.

Por fim como resultado foi desenvolvido um Jogo de Tabuleiro para ser utilizado para que os alunos desenvolvam as habilidades e competências de ILPC. E como foi sugestionado pelos participantes ele pode ser usado em qualquer curso em que se ministre a LP.

7.4.8 MELHORIAS REALIZADAS APÓS QUASE-EXPERIMENTO

Os participantes além de dar suas opiniões na partida, deixaram observações e sugestões como pode ser visto no final das perguntas no B, utilizadas para fazer as melhorias após Quase-experimento. As seguintes melhorias foram realizadas:

1. Foram melhorados as Regras, seguindo as sugestões dos participantes;
2. Mecânica do jogo, foi removida a parte sobre pagamento de impostos;
3. Foi adicionado novas perguntas sobre Linguagem de Programação;
4. Foi acrescentado novas cartas de sorte e revés;
5. Foi desenvolvido o Software de Análise de Desempenho dos Jogadores do Prog-Poly, para os jogadores poderem registrar os dados de suas partidas e analisar posteriormente seu desempenho nos estudos;

6. Além de nova versão do tabuleiro no formato hexagonal possibilitando aumentar as casas do tabuleiro, adicionando casas com informações introdutórias sobre Orientação a Objetos;
7. Utilizar o tabuleiro para alunos do nível superior.

Neste capítulo foi apresentado os métodos e procedimento para verificar a viabilidade do Prog-poly, foi apresentado o quase-experimento utilizado, a hipóteses validadas, a configuração do ambiente, a aplicação do quase-experimento, como foi realizado a coleta dos dados, análise dos resultados. No próximo capítulo será apresentado a conclusão do trabalho e no capítulo 9 os trabalhos futuros.

8 CONCLUSÃO

A ideia do jogo Prog-poly surgiu por meio das aulas da disciplina de Gerenciamento de Projetos de Software, ministrada pelo Professor Dr. José Augusto Fabri, onde foi desenvolvido um jogo de tabuleiro para auxiliar no ensino de Gestão de Projetos de Software. Através do *feedback* do professor foi sugerido o desenvolvimento de uma versão para auxiliar no ensino de ILPC.

Como resultado da revisão sistemática buscando jogos de tabuleiro para auxiliar nos estudos de LP e ES, vários trabalhos foram encontrados, mas baseados no Monopoly criado por Darrow (1935) para auxiliar nos estudos foram identificados os seguintes trabalhos Project-o-poly (PoP), Buglione (2007), RE-O-Poly Smith e Gotel (2008), *Mission to mars Release or iteration planning* Kruchten e King (2011), *StrateJect: An Interactive Game for Project Management Experiential Learning*, Telukunta Srikanth Kumar Kota et al. (2014), RealEstate Tiwari et al. (2012), *The monopoly game to teach ERi*c-intentional requirements engineering* Oliveira et al. (2015). Todas estas versões são voltadas para auxiliar nos estudos de ES, está é a primeira versão desenvolvida com o objetivo auxiliar nos estudos ILPC no ensino superior nos últimos 21 anos.

Os resultados apresentados neste trabalho sugerem a factibilidade da utilização do jogo de tabuleiro Prog-poly, com intenção de ser usado como ferramenta para que os alunos desenvolvam as habilidades e competências relacionadas à IES e ILPC por meio de perguntas e gamificação. O jogo tem como objetivo tornar os alunos mais engajados no processo de estudo e além disso aplicar conhecimentos de IES e ILPC.

Com o desenvolvimento do Prog-poly espera-se atenuar os fatos relatados como negativos pelos discentes e, conseqüentemente, permitir aos alunos que

desenvolvam as habilidades e competências relacionadas à IES e ILPC. A partir do quase-experimento realizado, buscou comparar os resultados de estudos de ILPC e interesse dos alunos entre o Prog-poly e o método clássico de estudos.

Os resultados obtidos com quase-experimento mostram que 95% dos alunos responderam que o Prog-poly motiva estudo de ILPC, 85% responderam que o jogo permitir aos alunos que desenvolvam as habilidades e competências relacionadas ILPC, 80% julgaram que o jogo apresenta uma forma de estudo menos cansativa do que os métodos tradicionais de estudo. Essa evidência demonstra que o jogo de tabuleiro pode ajudar nos estudos LP, além de tornar as aulas mais motivadoras.

Outra contribuição deste trabalho foi o desenvolvimento de um Software de Análise de Desempenho dos Jogadores do Prog-Poly. Com ele os participantes podem armazenar os dados das partidas e analisar seu desempenho, podendo assim identificar suas dificuldades.

O Trabalho futuro sugerido é o desenvolvimento de uma versão computacional do jogo de tabuleiro Prog-poly, implementar um aplicativo ou um site que funcione nos celulares atuais, podendo ser utilizado obedecendo aos rigores do distanciamento e medidas de segurança impostas pela pandemia do Coronavírus (COVID-19).

REFERÊNCIAS

ALDRICH, C. **Learning by doing: A comprehensive guide to simulations, computer games, and pedagogy in e-learning and other educational experiences**. [S.l.]: John Wiley & Sons, 2005.

BAKER, A.; NAVARRO, E. O.; HOEK, A. V. D. Problems and programmers: An educational software engineering card game. In: . [S.l.: s.n.], 2003. ISSN 02705257.

BARBOSA, A. de A.; FERREIRA, D. . S.; COSTA, E. B. Influência da linguagem no ensino introdutório de programação. In: . [s.n.], 2014. v. 25, p. 612–621. ISSN 2316-6533. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/2991%5Cnhttp://www.br-ie.org/pub/index.php/sbie/article/download/2991/2502>>.

BARNES, T.; POWELL, E.; CHAFFIN, A.; LIPFORD, H. Game2learn : Improving the motivation of cs1 students. **GDCSE '08 Proceedings of the 3rd international conference on Game development in computer science education**, 2008.

BERTOLDO, J. V.; RUSCHEL, M. A. de M. **Jogos, Brinquedo e Brincadeira – Uma Revisão Conceitual**. 2011.

BLOOM, B. A taxonomy of cognitive objectives. **A Taxonomy of Cognitive Objectives**, 1956.

BORGES, S. D. S.; DURELLI, V. H.; REIS, H. M.; ISOTANI, S. A systematic mapping on gamification applied to education. In: . [S.l.]: Association for Computing Machinery, 2014. p. 216–222. ISBN 9781450324694.

BOSSE, Y.; GEROSA, M. A. Reprovações e trancamentos nas disciplinas de introdução à programação da universidade de são paulo: Um estudo preliminar. In: . [S.l.: s.n.], 2015.

BUGLIONE, L. Project-o-poly. **Giocare per Apprendere. Il gioco come opportunità nelle Learning Organizations. Persone & Conoscenze**, p. 43–47, 2007. Disponível em: <<http://www.semq.eu/leng/proimplog1.htm>>.

BZUNECK, J. A. **A motivação do aluno: aspectos introdutórios**. 2004.

CAMPBELL, D. T.; STANLEY, J. C. **Experimental and quasi-experimental designs for research**. [S.l.]: Ravenio books, 2015.

CARLOS, J.; OLIVEIRA, N. D.; DA, R.; PEREIRA, S. **FORMAÇÃO DOS CUSTOS NA PRESTAÇÃO DE SERVIÇOS EM UMA EMPRESA DESENVOLVEDORA DE SOFTWARE LOCALIZADA EM JOÃO PESSOA-PB**. 2021.

CARVALHO, C. V. D. Aprendizagem baseada em jogos game-based learning. 2015. Disponível em: <<http://www.timemesh.eu>>.

CHAFFIN, A.; DORAN, K.; HICKS, D.; BARNES, T. Experimental evaluation of teaching recursion in a video game. **Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games - Sandbox '09**, v. 1, p. 79, 2009. ISSN 10797114. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1581073.1581086>>.

CHANG, W.; CHOU, Y.; CHEN, K.; CHANG, W.-C.; CHOU, Y.-M.; CHEN, K.-C. Game-based digital learning system assists and motivates c programming language learners. **International Conference on Networked Computing and ...**, p. 704–709, 2010. Disponível em: <<https://ieeexplore.ieee.org/document/5573266>>.

CHEN, P. P. S. The entity-relationship model—toward a unified view of data. **ACM Transactions on Database Systems (TODS)**, v. 1, 1976. ISSN 15574644.

CRISTIANO, A.; FÁBIO, B.; PARREIRA, J.; CUNHA, G. B. D.; MACEDO, R. T. **LINGUAGEM DE PROGRAMAÇÃO I LICENCIATURA EM COMPUTAÇÃO**. [S.l.: s.n.], 2019. ISBN 9788583412465.

DARROW, C. B. **Board game apparatus**. [S.l.]: Google Patents, 1935. US Patent 2,026,082.

DECUYPERE, M. Researching educational apps: ecologies, technologies, subjectivities and learning regimes. **Learning, Media and Technology**, v. 44, 2019. ISSN 17439892.

DETERDING, S.; DIXON, D.; KHALED, R.; NACKE, L. From game design elements to gamefulness: Defining “gamification”. p. 9–15, 2011. ISSN 1450308163.

EAGLE, M.; BARNES, T. Wu’s castle: Teaching arrays and loops in a game. **Proc. 13th Annu. Conf. Innov. Technol. Comput. Sci. Educ. - ITiCSE '08**, p. 245, 2008. ISSN 00978418. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1384271.1384337>>.

FLINDERS, D. J.; UHRMACHER, P. B. **Curriculum and Teaching Dialogue: Vol. 13 # 1 & 2**. Information Age Pub., 2012. 29-44 p. ISBN 9781617356070. Disponível em: <<https://books.google.com.br/books?id=QQMoDwAAQBAJ>>.

GILL, G. Growing checkio. **Journal of Information Technology Education: Discussion Cases**, v. 2, 2013. ISSN 2166-1316.

HOGLE, J. G. Considering games as cognitive tools: In search of effective ‘edutainment’. **University of Georgia Department of Instructional Technology**, p. 1–27, 1996. ISSN 0879860723. Disponível em: <<http://twinpinefarm.com/pdfs/games.pdf>>.

HUIZINGA, J. Homo ludens. **Editora Perspectiva**, v. 4, p. 162, 2000. ISSN 03043479.

JAIN, A.; BOEHM, B. Simvbse: Developing a game for value-based software engineering. In: . [S.l.: s.n.], 2006. v. 2006, p. 103–111. ISBN 0769525571. ISSN 10930175.

KISHIMOTO, T. M. **Jogo, brinquedo, brincadeira e a educação**. [S.l.]: Cortez editora, 2017. ISBN 978-85-249-2570-2.

KITCHENHAM, B. Procedures for performing systematic reviews. **Keele, UK, Keele University**, v. 33, p. 28, 2004. ISSN 13537776.

KRATHWOHL, A. dan. Anderson and krathwohl - understanding the new version of bloom ' s taxonomy the cognitive domain : Anderson and krathwohl - bloom ' s taxonomy revised. **A succinct discussion of the revisions to Bloom's classic cognitive taxonomy by Lorin Anderson and David Krathwohl and how to use them effectively**, v. 41, 2001.

KRUCHTEN, P.; KING, J. Mission to mars: An agile release planning game. **2011 24th IEEE-CS Conference on Software Engineering Education and Training, CSEE and T 2011 - Proceedings**, p. 552, 2011. ISSN 1093-0175.

LAROUSSE, K. Grande dicionário lrousse cultural da língua portuguesa. **Rio de Janeiro: Nova Cultural**, 1999.

MAGNO, E.; FIGUEIREDO, L.; ARACATY, C.; KLESSIS, L.; DIAS, L.; SAMPAIO, J. C.; LEITE, P.; LUCENA, C. J. P. D. Simules: Um jogo para o ensino de engenharia de software. 2006. ISSN 0103-9741.

MARTIN, A. The design and evolution of a simulation/game for teaching information systems development. **Simul. Gaming**, v. 31, p. 445–463, 2000. ISSN 10468781.

MEDEIROS, T. J.; REIS, T. Ensino de programação utilizando jogos digitais : uma revisão sistemática da literatura. **Renote**, v. 11, p. 1–10, 2013.

MONSALVE, E. S.; WERNECK, V. B. M.; LEITE, J. C. S. do P. **SimulES-W: Um Jogo para o Ensino de Engenharia de Software**. 2010. Disponível em: <www.inf.puc-rio.br/~julio>.

MOREIRA, G. L.; HOLANDA, W.; COUTINHO, J. C. da S.; CHAGAS, F. S. Desafios na aprendizagem de programação introdutória em cursos de ti da ufersa , campus pau dos ferros : um estudo exploratório. In: . [S.l.: s.n.], 2018. ISSN 25267574.

NETO, A.; SILVA, A. P. da; BITTENCOURT, I. I.; SILVA, A. P. da; BITTENCOURT, I. I.; SILVA, A. P. da; BITTENCOURT, I. I.; SILVA, A. P. da; BITTENCOURT, I. I.; SILVA, A. P. da; BITTENCOURT, I. I. Uma análise do impacto da utilização de técnicas de gamificação como estratégia didática no aprendizado dos alunos. p. 667, 2015. ISSN 2316-6533. Disponível em: <<http://br-ie.org/pub/index.php/sbie/article/view/5336>>.

OLIVEIRA, A. D. P. A.; WERNECK, V. M. B.; LEITE, J. C. S. D. P.; CYSNEIROS, L. M. The monopoly game to teach eri*c-intentional requirements engineering. In: . [S.l.: s.n.], 2015. v. 1370. ISSN 16130073.

OLIVEIRA, C. D. D.; CINTRA, M. E.; NETO, F. M. M. Learning risk management in software projects with a serious game based on intelligent agents and fuzzy systems. In: . [S.l.: s.n.], 2013. v. 32.

PORTMAN, H. **Review Standish Group – CHAOS 2020: Beyond Infinity**. 2021.

PRENSKY, M. Digital natives, digital immigrants. **On the Horizon**, 2001. ISSN 00396109.

PRENSKY, M. **Digital Game-based Learning**. 2007.

PRENSKY, M. **Aprendizagem Baseada Em Jogos Digitais**. [S.l.]: Senac São Paulo, 2012. 576 p. ISBN 9788539602711.

RAABE, A. L. A.; SILVA, J. M. C. da S. Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. **XXV Congresso da Sociedade Brasileira de Computação**, p. 2326–2337, 2005.

RAMINGWONG, S.; RAMINGWONG, L. Armi: A risk management incorporation. In: . [S.l.: s.n.], 2014.

RESNICK, M.; MALONEY, J.; MONROY, A.; RUSK, N.; EASTMOND, E.; BRENNAN, K.; MILLNER, A.; ROSENBAUM, E.; SILVER, J.; SILVERMAN, B.; KAFAL, Y. Scratch: Programming for all. 2009. ISSN 00010782.

SALAZAR, R.; ODAKURA, V.; BARVINSKI, C. Scratch no ensino superior: motivação. In: . [S.l.: s.n.], 2015. v. 1.

SANTANA, P.; FORTES, D.; PORTO, R. Jogos digitais: A utilização no processo ensino aprendizagem. **Revista Científica da FASETE**, 2016. ISSN 1098-6596.

SANZOVO, N. S.; ESCOLA, J. J. J. Proposta didático-metodológica no enfrentamento da baixa aprendizagem na área do cálculo. **Revista de Estudios e Investigación en Psicología y Educación**, Universidade da Coruna, p. 176–181, 12 2017. ISSN 2386-7418.

SELLTIZ; WRIGHTSMAN; COOK. Texto 6 : Delineamentos quase-experimentais. **Métodos de pesquisa nas relações sociais**, 1976.

SENA, S. D.; SCHMIEGELOW, S. S.; PRADO, G. M. B. C. do; SOUSA, R. P. L. de; FIALHO, F. A. P. Aprendizagem baseada em jogos digitais: a contribuição dos jogos epistêmicos na geração de novos conhecimentos. **RENOTE**, v. 14, 2016. ISSN 1679-1916.

SEVERGNINI, L.; SOARES, E. O serious game codecombat e o professor como mediadores da aprendizagem do pensamento computacional. In: . [S.l.: s.n.], 2019. ISSN 2316-6533.

SILVA, A. C. et al. Jogo educacional para apoiar o ensino de técnicas para elaboração de testes de unidade. Universidade do Vale do Itajaí, 2010.

SILVA, T. R. da; MEDEIROS, T. J.; ARANHA, E. H. S.; SILVA, T. da; MEDEIROS, T. J.; ARANHA, E. H. S. Jogos digitais para ensino e aprendizagem de programação: uma revisão sistemática da literatura. **Anais do Simpósio Brasileiro de**, v. 25, p. 692–701, 2014. ISSN 2316-6533. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/3000>>.

SINGH, J.; DORAIRAJ, S. K.; WOODS, P. Learning computer programming using a board game: Case study on c-jump. **Masters thesis, Multimedia University**, 2007.

SMITH, C.; STRICK, L. **Dificuldades de aprendizagem de A a Z: um guia completo para pais e educadores**. [S.l.]: Penso Editora, 2001. 332 p. ISBN 9788536312521.

SMITH, R.; GOTEL, O. Gameplay to introduce and reinforce requirements engineering practices. In: . [S.l.: s.n.], 2008.

SOARES, D. Reflexão crítica sobre estudos quasi-experimentais. 2021. Disponível em: <<https://www.researchgate.net/publication/349831240>>.

SOARES, F. A. L.; CARVALHO, R. B. Proposta de um portal educacional para estudantes de programação de computadores. **Abakós**, v. 5, 2017. ISSN 2316-9451.

SOMMERVILLE, I. **Engenharia de software**. [S.l.]: Pearson Prentice Hall, 2011. ISBN 9788579361081.

SONCHAN, P.; RAMINGWONG, S. Armi 2.0: An online risk management simulation. In: . [S.l.: s.n.], 2015.

TACOURI, H.; NAGOWAH, L. Code saga – a mobile serious game for learning programming. In: . IEEE, 2021. p. 190–195. ISBN 978-1-6654-2035-8. Disponível em: <<https://ieeexplore.ieee.org/document/9628484/>>.

TELUKUNTA SRIKANTH KUMAR KOTA, S. P. H. S. T. R. S.; TELUKUNTA, S.; SRIKANTH, M.; KOTA, K.; POTTI, S.; SHASHANK, M.; REDDY, M.; TELUKUNTA, T. R. S.; OTHERS. Strateject: An interactive game for project management experiential learning. **Project Management Institute (PMI)PMPI**, v. 1, p. 1–15, 2014.

THUNG, F.; BISSYANDÉ, T. F.; LO, D.; JIANG, L. Network structure of social coding in github. In: . [S.l.: s.n.], 2013. p. 323–326. ISBN 9780769549484. ISSN 15345351.

TILLMANN, N.; HALLEUX, J. D.; XIE, T.; BISHOP, J. Pex4fun: Teaching and learning computer science via social gaming. In: . [S.l.: s.n.], 2012. ISBN 9780769546933. ISSN 1093-0175.

TILLMANN, N.; HALLEUX, J. de; XIE, T.; BISHOP, J. Code hunt. In: . [S.l.: s.n.], 2014. ISBN 9781450326698.

TIWARI, S.; RATHORE, S. S.; GUPTA, A. Selecting requirement elicitation techniques for software projects. In: . [S.l.: s.n.], 2012.

TRAVASSOS, G.; GUROV, D.; AMARAL, E. **Introdução à Engenharia de Software Experimental**. 2002. 52 p.

TSAI, T. H.; LIN, H. C.; HUANG, K. C. Digital game-based learning on digital archives: A case study of taiwanese classical poems. In: . [S.l.: s.n.], 2012.

VIANNA, Y.; VIANNA, M.; MEDINA, B.; TANAKA, S. **Gamification, Inc. Como reinventar empresas a partir de jogos**. [S.l.: s.n.], 2013. ISSN 1098-6596.

WANGENHEIM, C. G. V.; SAVI, R.; BORGATTO, A. F. Deliver! - an educational game for teaching earned value management in computing courses. **Information and Software Technology**, v. 54, 2012. ISSN 09505849.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. **Experimentation in software engineering**. [S.l.: s.n.], 2012. 1-236 p. ISSN 00652458. ISBN 9783642290442.

WOJEWODA, S.; HASTIE, S. Chaos report. **The Standish Group International, Inc**, p. 1–13, 2015. Disponível em: <https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf>.

YANNAKAKIS, G. N.; TOGELIUS, J. **Artificial intelligence and games**. [S.l.: s.n.], 2018.

YMRAN, F.; AKEEM, O.; YI, S. Gamification design in a history e-learning context. In: . [S.l.: s.n.], 2018.

ZICHERMANN, G.; CUNNINGHAM, C. **Gamification by design: Implementing game mechanics in web and mobile apps**. [S.l.]: "O'Reilly Media, Inc.", 2011. ISBN 9350234548.

APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

A seguir sera apresentado o "TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO", que os participantes assinaram antes de participaram do quase-experimento do jogo Prog-poly.

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Prezado Sr(a).

Como parte de uma pesquisa de mestrado, este estudo visa investigar como o jogo “Prog-poly” auxilia no ensino da algoritmos. Os jogos motivam os participantes a trabalharem em equipe e tomar decisões. Desta forma, você está sendo convidado a participar de uma pesquisa que estudará os resultados de utilização desse jogo como parte do aprendizado de algoritmos. O objetivo da pesquisa é avaliar se a abordagem é adequada (aplicável) e identificar pontos que precisam ser aperfeiçoados. Sua participação na pesquisa é muito importante e voluntária.

1) Procedimento

Na primeira etapa pedimos que você reúna algumas pessoas para jogar Prog-poly, seguindo as regras do mesmo.

Na segunda etapa você será convidado a responder o questionário com perguntas relacionadas a dinâmica do Jogo e conteúdo abordado no Jogo.

O tempo para finalizar o Jogo, é variável. Já a segunda etapa, tem uma duração prevista de 15 minutos.

2) Tratamento de possíveis riscos e desconfortos

Serão tomadas todas as providências durante a coleta de dados de forma a garantir a sua privacidade e seu anonimato. Os dados coletados durante o estudo destinam-se estritamente as atividades de pesquisa relacionadas à abordagem, não sendo utilizados em qualquer forma de avaliação profissional ou pessoal.

3) Benefícios e Custos

Espera-se que a participação neste estudo lhe seja benéfica, visto que você terá contato com conhecimentos em desenvolvimento de jogos e algoritmos.

Este estudo também contribuirá com resultados importantes para a pesquisa de modo geral nas áreas de Engenharia de Software, principalmente na área de algoritmos.

Você não terá nenhum gasto ou ônus com a sua participação no estudo e também não receberá nenhuma espécie de reembolso ou gratificação devido à participação na pesquisa.

4) Confidencialidade da Pesquisa

Toda informação coletada neste estudo é confidencial e seu nome e o da sua organização não serão identificados de modo algum, a não ser em caso de autorização explícita para esse fim.

5) Participação

Sua participação neste estudo é muito importante e voluntária. Você tem o direito de não querer participar ou de sair deste estudo a qualquer momento, sem penalidades.

Os pesquisadores responsáveis pelo estudo e o Professor orientador poderão fornecer qualquer esclarecimento sobre o mesmo, assim como tirar dúvidas, bastando entrar em contato.

PESQUISADOR RESPONSÁVEL

Luciano Rovanni do Nascimento – rovanni@gmail.com

Aluno do Programa de Pós-Graduação em Informática - PPGI, da Universidade Tecnológica

Federal do Paraná, Câmpus Cornélio Procópio (UTFPR-CP)

Professor Orientador: Prof. Dr. José Augusto Fabri – fabri@utfpr.edu.br (UTFPR-CP)

Muito obrigado!

***Obrigatório**

1. E-mail *

2. Informe seu nome completo *

3. Informe seu local de trabalho *

4. Você concorda com este Termo de Consentimento Livre e Esclarecido? *

Marcar apenas uma oval.

Sim

Não

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

APÊNDICE B – PERGUNTAS

A seguir serão apresentados as perguntas sobre programação que serão utilizadas no jogo Prog-poly.

PERGUNTAS

Sobre Compiladores:

1. Um compilador é um programa de computador (ou um grupo de programas) que, a partir de um código-fonte, escrito em uma linguagem máquina, cria um programa semanticamente equivalente, porém escrito em outra linguagem de alto nível, código objeto. Fonte: “Compilador” (2022).
a) Errado (X)
b) Certo ()
2. Classicamente, um compilador traduz um programa de uma linguagem textual facilmente entendida por um ser humano para uma linguagem de máquina, específica para um processador e sistema operacional. Fonte: “Compilador” (2022).
a) Errado ()
b) Certo (X)
3. Um compilador traduz um programa de uma linguagem de máquina para uma linguagem textual facilmente entendida por um ser humano. Fonte: “Compilador” (2022).
a) Errado (X)
b) Certo ()
4. Com relação a linguagens de programação e compiladores, julgue os itens subsequentes. Um algoritmo computacional escrito em linguagem de programação pode ser completamente executado sem gerar nenhuma saída. Fonte: Concursos (2022)..
a) Errado (X)
b) Certo ()
5. Linguagem compilada é uma linguagem de programação em que o código-fonte, nessa linguagem, é executado diretamente pelo sistema operacional ou pelo processador, após ser traduzido através de um processo chamado compilação, usando um programa de computador chamado compilador, para uma linguagem de baixo nível, como linguagem de montagem “Assembly” ou código de máquina. Fonte: “Linguagem compilada” (2022).
a) Errado ()
b) Certo (X)
6. Linguagem interpretada é uma linguagem de programação em que o código-fonte, nessa linguagem, é executado diretamente pelo sistema operacional ou pelo processador, após passar pelo interpretador é gerado um arquivo executável. Fonte: “Linguagem interpretada” (2022).
a) Errado (X)
b) Certo ()
7. Linguagem interpretada é uma linguagem de programação em que o código-fonte nessa linguagem é executado por um programa de computador

chamado interpretador, executado em seguida pelo sistema operacional ou processador. Exemplo de linguagens interpretadas (ASP, JavaScript, PHP, Python). Fonte: "Linguagem interpretada" (2022).

- a) Errado ()
- b) Certo (X)

8. Linguagem compilada gera um arquivo executável, que pode ser executado diretamente pelo sistema operacional. Existem linguagens que não geram código executável, como as linguagens interpretadas. Exemplo de linguagens compiladas (C, C++, COBOL, Delphi, Visual Basic). Fonte: "Linguagem compilada" (2022).

- a) Errado ()
- b) Certo (X)

9. Acerca das características das principais linguagens de programação, julgue o item seguinte.

Linguagem de programação de alto nível consiste em um conjunto de símbolos, palavras e regras próximo da linguagem humana, e que é fácil de escrever, depurar e manter. Fonte: "Questões de Concursos" (2022).

- a) Errado ()
- b) Certo (X)

10. Em programação de computadores, uma palavra reservada é uma palavra que, em algumas linguagens de programação, não pode ser utilizada como um identificador por ser reservada para uso da gramática da linguagem. Por exemplo, na linguagem de programação Java, um programador não pode utilizar uma variável com o nome (**For**), pois esta palavra é reservada para construção de loops. Ela é uma "palavra-chave", e por ser de uso restrito, é também uma "palavra reservada". (Em algumas linguagens, como C ou Java, os termos têm o mesmo significado). Fonte: "Palavra reservada" (2022).

- a) Errado ()
- b) Certo (X)

11. *Case-sensitive* é um anglicismo que se refere a uma categoria de análise tipográfica da informática. Em língua portuguesa, significa algo como "sensível à caixa das letras" ou "sensível a maiúsculas e minúsculas". Diz-se que um software é *case-sensitive* ou possui "*case-sensitivity*" quando ele consegue analisar uma cadeia de caracteres, avaliar a existência de caixa alta e caixa baixa e comportar-se de diferentes maneiras em função disso. Fonte: "Case-sensitive" (2022).

- a) Errado ()
- b) Certo (X)

12. Diz-se que um software não é *case-sensitive* ou não possui "*case-sensitivity*" quando ele não consegue analisar uma cadeia de caracteres, e avaliar a existência de caixa alta e caixa baixa e comportar-se de diferentes maneiras em função disso. Fonte: "Case-sensitive" (2022).

- a) Errado (X)
- b) Certo ()

Sobre Bibliotecas

13. Na ciência da computação, biblioteca é uma coleção de subprogramas utilizados no desenvolvimento de *software*. Bibliotecas contêm código e dados auxiliares, que provêm serviços a programas independentes, o que permite o compartilhamento e a alteração de código e dados de forma modular. Fonte: “Biblioteca (computação)” (2022).
- a) Errado ()
 - b) Certo (X)
14. Alguns executáveis são tanto programas independentes quanto bibliotecas, mas a maioria das bibliotecas não é executáveis. Fonte:
- a) Errado ()
 - b) Certo (X)
15. A maior parte dos sistemas operacionais modernos provê bibliotecas que implementam a maioria dos serviços do sistema, que transformaram em comodidades os serviços que uma aplicação moderna espera providos pelo sistema operacional. Assim sendo, a maior parte do código utilizado em aplicações modernas é fornecido por estas bibliotecas. Fonte: “Biblioteca (computação)” (2022).
- a) Errado ()
 - b) Certo (X)
16. A maior parte dos sistemas operacionais modernos não provê bibliotecas para implementam a maioria dos serviços do sistema. Assim sendo a maior parte do código utilizado em aplicações modernas não é fornecido por estas bibliotecas. Fonte: “Biblioteca (computação)” (2022).
- a) Errado (X)
 - b) Certo ()
17. Quando um programa é desenvolvido utilizando uma linguagem de programação pode usar um conjunto de funções pré-escritas por outros programadores que já resolvem determinados problemas para que os programadores não precisem em todo programa “reinventar a roda”, a esse conjunto de funções damos o nome de? Fonte: “Biblioteca (computação)” (2022).
- a) Variável ()
 - b) *String* ()
 - c) Biblioteca (X)
 - d) *Char* ()
 - e) *Integer* ()

Sobre variáveis

18. Sobre variáveis escolha a opção correta.
Na programação, uma variável é um objeto (uma posição, frequentemente localizada na memória) capaz de reter e representar um valor ou expressão. Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

19. Sobre variáveis escolha a opção correta.

Variável do tipo Inteiro: armazena números que não possuem casas decimais, apenas a parte inteira e podem ser positivos ou negativos (ex: 1, 2, 67, 999). Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

20. Variável do tipo Inteiro: armazena caracteres (ex: A, B, C, a, b) e símbolos como (ex: @, \$, %). Fonte: “Variável (programação)” (2022).

- a) Errado (X)
- b) Certo ()

21. Sobre variáveis escolha a opção correta.

Variável do tipo Real: números que possuem casas decimais e também podem ser positivos, negativos e inteiros (ex: 1, 2.3, 5.7, 78) para usar casas decimais é necessário usar PONTO ao invés de VÍRGULA. Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

22. Sobre variáveis escolha a opção correta.

Variável do tipo *Char* “Caractere”: nesse tipo, apenas um caractere pode ser armazenado, pode ser uma letra, número ou símbolo (dependendo da linguagem), conforme o padrão ASCII (ex: A, B, 7, J) os números serão interpretados como caractere então não terão compatibilidade com inteiros e reais. Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

23. Sobre variáveis escolha a opção correta.

Variável do tipo *String*: são espaços para você armazenar um conjunto de caracteres “palavras”, frases e qualquer coisa que utilize vários caracteres (ex: “Olá mundo”). Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

24. Uma variável é um espaço de memória reservado para armazenar um certo tipo de dado, que pode ser um número, caractere, símbolo. Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

25. Variável tem um nome para referenciar o seu conteúdo. Fonte: “Variável (programação)” (2022).
- a) Errado ()
 - b) Certo (X)
26. Sobre variáveis escolha a opção correta. Fonte: “Variável (programação)” (2022).
- a) () O valor de uma variável é o conjunto da(s) lista(s) de controle acessadas por ela.
 - b) () O valor de uma variável é o conteúdo do(s) caractere(s) alfanuméricos de sua denominação.
 - c) () O identificador de uma variável é o conteúdo da(s) célula(s) de sequenciamento associada(s) a ela.
 - d) (X) O valor de uma variável é o conteúdo da(s) célula(s) de memória associada(s) a ela. (Correta)
 - e) () O valor de uma variável de controle é o identificador de uma estrutura de repetição.

Sobre Vetores e Matrizes

27. Vetor (*array* uni-dimensional) é uma variável que armazena várias variáveis do mesmo tipo. Fonte: De Oliveira Alves (2013a)
- a) Errado ()
 - b) Certo (X)
28. Os Vetores são matrizes de uma só dimensão. Fonte: De Oliveira Alves (2013a)
- a) Errado ()
 - b) Certo (X)
29. Os Vetores são *array* de múltiplas dimensões. Fonte: De Oliveira Alves (2013a)
- a) Errado (X)
 - b) Certo ()
30. Uma matriz é uma coleção de variáveis de mesmo tipo, acessíveis com um único nome e armazenados contiguamente na memória. A individualização de cada variável de um vetor é feita através do uso de índices. Fonte: De Oliveira Alves (2013a)
- a) Errado ()
 - b) Certo (X)

Sobre Operadores

Operadores Aritméticos

31. Operadores aritméticos são aqueles que estudamos na escola, aquelas funções básicas de somar (+), subtrair (-), multiplicar (*), dividir (/), operador de módulo ou resto da divisão (%). Fonte: Eduardo (2012 e De Oliveira Alves (2013a)

- a) Errado ()
- b) Certo (X)

32. Operadores aritméticos são representados por: (se, então, senão)? Fonte: Eduardo (2012 e De Oliveira Alves (2013a)

- a) Errado (X)
- b) Certo ()

Operadores Relacionais

33. Os operadores relacionais comparam dois valores e retornam um valor booleano (*true* ou *false*). Os operadores são Igual a (==), Diferente de (!=), Maior que (>), Menor que (<), Maior do que ou igual a (>=), Menor do que ou igual a (<=). Fonte: Corob-msft (2022).

- a) Errado (X)
- b) Certo ()

34. Os operadores relacionais comparam dois valores e retornam um valor booleano (*true* ou *false*). O exemplo a seguir ajuda a ilustrar o uso destes operadores? Fonte: Corob-msft (2022).

int a = 10, b = 25, c = 50, d = 100;

Esta operação (a == d) tem como resultado TRUE?

- a) Errado (X)
- b) Certo ()

35. Os operadores relacionais comparam dois valores e retornam um valor booleano (*true* ou *false*). O exemplo a seguir ajuda a ilustrar o uso destes operadores. Fonte: Corob-msft (2022).

int a = 10, b = 25, c = 50, d = 100;

Esta operação (a == d) tem como resultado FALSE?

- a) Errado ()
- b) Certo (X)

Operadores Lógicos

36. O operador lógico E (*AND*) somente retorna verdadeiro se todas as condições de entrada forem verdadeiras. Por exemplo, (3 > 2) é uma expressão lógica que possui valor verdadeiro. Por outro lado, (4 < 1) é uma expressão lógica que possui valor falso. Analise o "E" (&&) do exercício abaixo com código na "linguagem C". Fonte: Eduardo (2019)

$((3 < 2) \&\& (2 == 2))$: resultado **FALSO**?

- a) Errado ()
- b) Certo (X)

37. O operador lógico E (AND) somente retorna verdadeiro se todas as condições de entrada forem verdadeiras. Por exemplo, $(3 > 2)$ é uma expressão lógica que possui valor verdadeiro. Por outro lado, $(4 < 1)$ é uma expressão lógica que possui valor falso. Analise o “E” (**&&**) do exercício abaixo com código na “linguagem C”. Fonte: Eduardo (2019)

$((3 < 2) \&\& (2 == 2))$: resultado **VERDADEIRO**?

- a) Errado (X)
- b) Certo ()

38. O operador lógico OU (OR) somente retorna falso se todas as condições de entrada forem falsas. Por exemplo, $(3 > 2)$ é uma expressão lógica que possui valor verdadeiro. Por outro lado, $(4 < 1)$ é uma expressão lógica que possui valor falso. Analise o “OU” (**||**) do exercício abaixo com código na “linguagem C”. Fonte: Eduardo (2019)

$((2 > 1) || (3 < 7))$: resultado **VERDADEIRO**?

- a) Errado ()
- b) Certo (X)

39. O operador lógico OU (OR) somente retorna falso se todas as condições de entrada forem falsas. Por exemplo, $(3 > 2)$ é uma expressão lógica que possui valor verdadeiro. Por outro lado, $(4 < 1)$ é uma expressão lógica que possui valor falso. Analise o “OU” (**||**) do exercício abaixo com código na “linguagem C”. Fonte: Eduardo (2019)

$((2 > 1) || (3 < 7))$: resultado **FALSO**?

- a) Errado (X)
- b) Certo ()

40. O operador lógico NÃO (NOT) inverte a condição de entrada: verdadeiro se torna falso, e falso se torna verdadeiro. Por exemplo, $(3 > 2)$ é uma expressão lógica que possui valor verdadeiro. Fonte: Eduardo (2019)

A operação: $!(3 > 2)$, possui resultado **VERDADEIRO**?

- a) Errado (X)
- b) Certo ()

41. O operador lógico NÃO (NOT) inverte a condição de entrada: verdadeiro se torna falso, e falso se torna verdadeiro. Por exemplo, $(3 > 2)$ é uma expressão lógica que possui valor verdadeiro. Fonte: Eduardo (2019)

$!(3 > 2)$: resultado **FALSO**?

- a) Errado ()
- b) Certo (X)

Estrutura Condicional

42. Os comandos de decisão, como, por exemplo, os comandos *IF*, *THEN*, *ELSE*, não são considerados como comandos de controle de fluxo pelo fato de utilizarem operadores aritméticos. Fonte: Neiva (2022).
- a) Errado (X)
b) Certo ()
43. Em ciência da computação, indentação (reco, neologismo derivado da palavra em inglês *indentation*) é um termo aplicado ao código-fonte de um programa para ressaltar ou definir a estrutura do algoritmo. Na maioria das linguagens de programação, a indentação é empregada visando ressaltar a estrutura do algoritmo, aumentando assim a legibilidade do código. Em algumas linguagens, entretanto, a indentação é obrigatória. Python, Occam e Haskell, por exemplo, utilizam a mudança de indentação para definir a hierarquia dentre blocos de código. Fonte: Neiva (2022).
- a) Errado ()
b) Certo (X)
44. Sobre indentação, enquanto os blocos são delimitados explicitamente em C, Java e PHP por chaves e em Pascal e Fortran por palavras-chave como *then* e *endif*, em Python blocos são delimitados por espaços ou tabulações formando uma indentação visual; não existem símbolos de “abre” e “fecha”. Fonte: “Python/Conceitos básicos/Indentação” (2022).
- a) Errado ()
b) Certo (X)
45. Indentação não deixa o código mais legível, não serve para melhorar a manutenção do código e nem para facilitar a tarefa de encontrar erros na codificação. Fonte: Neitzke (2022).
- a) Errado (X)
b) Certo ()
46. Python requer uma indentação padronizada. Em outras linguagens, como C/C++ ou JavaScript, a indentação não é necessária devido aos delimitadores de blocos, utilizada somente para melhor visualização. Fonte: “Python/Conceitos básicos/Indentação” (2022).
- a) Errado ()
b) Certo (X)
47. Python não requer uma indentação padronizada. Em outras linguagens, como C/C++ ou JavaScript, a indentação é necessária devido aos delimitadores de blocos, utilizada somente para melhor visualização. Fonte: “Python/Conceitos básicos/Indentação” (2022).
- a) Errado (X)
b) Certo ()
48. Estrutura de seleção (expressão condicional ou ainda construção condicional) é, na ciência da computação, uma estrutura de desvio do fluxo de controle presente em linguagens de programação que realiza diferentes

computações ou ações dependendo se a seleção (ou condição) é verdadeira ou falsa, em que a expressão é processada e transformada em um valor booleano. Nas linguagens de programação, usamos as palavras em inglês para expressar uma estrutura de seleção, como *if*, *else if* e *else*. Fonte: “Estrutura de seleção” (2022).

- a) Errado ()
b) Certo (X)

49. Quando o interpretador encontra o identificador "Se", ele espera em seguida uma condição booleana sob forma de expressão relacional (como, por exemplo, $(x > 0)$), que pode ser verdadeira ou falsa. Se a condição for verdadeira, o bloco de código seguido de "Então" é executado. Caso a condição seja falsa, o bloco de código seguido de "Senão" é executado. Em ambos os casos, após a execução do bloco de código, o fluxo do programa é retornado para o ponto indicado por "Fim Se". Note que a parte do "Senão" é opcional e pode ser omitida. Caso omitida, a estrutura é chamada de seleção simples; caso contrário é chamada seleção composta. Fonte: “Estrutura de seleção” (2022).

- a) Errado ()
b) Certo (X)

50. Seleção simples e composta. A condição "se-então (senão)" é uma estrutura de seleção comum em diversas linguagens de programação. Sua estrutura básica é a seguinte: Fonte: “Estrutura de seleção” (2022).

Se (condição) Então
(bloco de código)

Senão
(bloco de código)

Fim Se

- a) Errado ()
b) Certo (X)

51. A estrutura "Senão Se" é formada pela composição de estruturas de seleção, e um exemplo de sua estrutura é o seguinte: Fonte: “Estrutura de seleção” (2022).

Se (condição) Então
(bloco de código)

Senão
Se (condição) Então
(bloco de código)

Fim Se

Fim Se

- a) Errado ()
b) Certo (X)

52. Seleção de múltipla escolha. A seleção de múltipla escolha compara um dado valor a **constantes**, desviando o fluxo de código para o ponto indicado pela primeira constante que é igual à **constante** escolhida. Sua estrutura básica é: Fonte: “Estrutura de seleção” (2022).

Escolha X

Caso V1:
(bloco de código)

Caso V2:
(bloco de código)

Caso contrário:
(bloco de código)

Fim Escolha

a) Errado

b) Certo

Estrutura de repetição

53. Estrutura de repetição permite executar mais de uma vez o mesmo comando ou conjunto de comandos, de acordo com uma condição ou com um contador. Fonte: "Estrutura de seleção" (e Zanelato (2018)

a) Errado

b) Certo

54. Estrutura de repetição permite executar somente uma vez o mesmo comando ou conjunto de comandos, de acordo com uma condição ou com um contador. Fonte: "Estrutura de seleção" (e Zanelato (2018)

a) Errado

b) Certo

55. Estrutura de repetição são utilizadas para repetir ações semelhantes, executadas para todos os elementos de uma lista de dados, ou simplesmente para repetir um mesmo bloco de código até que a condição de parada seja satisfeita. Fonte: "Estrutura de seleção" (e Zanelato (2018)

a) Errado

b) Certo

56. Estrutura de Repetição Enquanto (While)

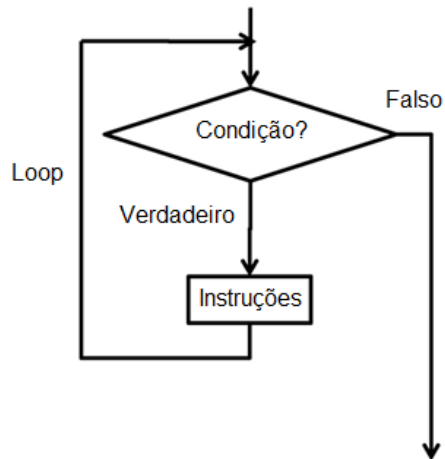
Essa estrutura realiza um teste lógico no início do loop, e cada vez que o teste retorna o valor verdadeiro, os comandos associados ao laço são executados.

Quando o teste retornar falso, o laço é encerrado e o processamento volta para o fluxo principal do programa. Fonte: "Estrutura de seleção" (e Zanelato (2018)

a) Errado

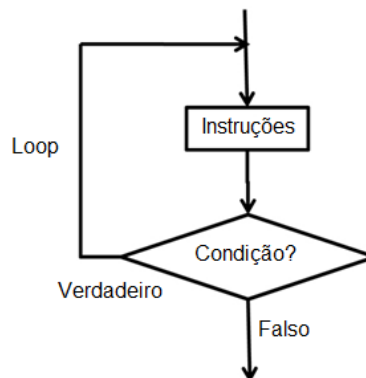
b) Certo

57. A ilustração a seguir mostra o funcionamento de qual estrutura de repetição?
Fonte: Autoria própria.



- a) () For (Para)
- b) () Do While (Faça enquanto)
- c) (X) While (Enquanto)

58. A ilustração a seguir mostra o funcionamento de qual estrutura de repetição?
 Fonte: Autoria própria.



- a) () For (Para)
- b) () While (Enquanto)
- c) (X) Do While (Faça enquanto)

59. O Do while tem quase o mesmo funcionamento que o while, mas neste caso, o bloco de comando estará dentro das chaves do comando DO e a verificação será feita pelo WHILE logo após a execução, a diferença é que com o uso dele teremos os comandos executados ao menos uma única vez. Fonte: "Programação C/C++ - Comandos de Repetição" (2022).

- a) Errado ()
- b) Certo (X)

Sobre Funções

60. As funções (functions), também conhecidas como sub-rotinas, são muito utilizadas em programação. Um dos grandes benefícios é não precisar copiar o código todas as vezes que precisar executar aquela operação, além de deixar a leitura do código mais intuitiva. No exemplo anterior, caso precisássemos descobrir a raiz quadrada de 10 números, bastaria chamar a função que calcula a raiz quadrada 10 vezes. Fonte: De Oliveira Alves (2013b)

- a) Errado ()
- b) Certo (X)

61. Exemplo de Programa em C.

```
int main( )  
{  
    return 0;  
}
```

Essa função main, neste exemplo, é do tipo int

Significa que a função deverá obrigatoriamente retornar um **número inteiro**.

Fonte: Autoria própria.

- a) Errado ()
- b) Certo (X)

Sobre Procedimentos

62. Os procedimentos (procedures) diferem das funções apenas por não retornarem resultado. Enquanto que a função é um algoritmo que será executado e produzirá um resultado final concreto que poderá ser usado por outro algoritmo. Fonte: De Oliveira Alves (2013b)

- a) Errado ()
- b) Certo (X)

Sobre Registros

63. Um registro é uma estrutura de dados heterogênea, ou seja, formada por um conjunto de variáveis (campos) que podem assumir tipos diferentes de dados, inclusive os tipos compostos (vetores, matrizes e registros). Fonte: "Struct - Registros em Linguagem C com struct - Embarcados" (2017)

- a) Errado ()
- b) Certo (X)

64. Um registro é uma estrutura de dados não heterogênea, ou seja, formada por um conjunto de variáveis (campos) que podem assumir tipos iguais de dados. Fonte: "Struct - Registros em Linguagem C com struct - Embarcados" (2017)

- a) Errado (X)
- b) Certo ()

65. Em Linguagem C Registros são chamados de Estruturas e a palavra-chave é Struct. Fonte: “Struct - Registros em Linguagem C com struct - Embarcados” (2017)

Ex: Registros Linguagem C.

```
struct aluno {  
    int codigo;  
    char nome[200];  
    float nota;  
};  
struct aluno aluno_especial, aluno_regular, aluno_ouvinte;
```

- a) Errado ()
- b) Certo (X)

Sobre: Programação Orientada a Objetos.

66. Uma classe é um tipo definido pelo usuário que contém a “receita”, a especificação para os objetos, algo mais ou menos como o tipo inteiro contém o molde para as variáveis declaradas como inteiros. A classe envolve, associa, funções e dados, controlando o acesso a estes, defini-la implica em especificar os seus atributos (dados) e seus métodos (funções). Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

67. Objetos (computacionais) são caracterizados por atributos e métodos. Atributos são as propriedades de um objeto. Métodos são as ações que um objeto pode realizar. Os objetos são características definidas pelas classes. Neles é permitido instanciar objetos da classe para inicializar os atributos e invocar os métodos. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

68. Atributos são as características de um objeto, essas características também são conhecidas como variáveis, utilizando o exemplo dos cães, temos alguns atributos, tais como: cor, peso, altura e nome. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

69. Métodos são as ações que os objetos podem exercer quando solicitados, onde podem interagir e se comunicarem com outros objetos, utilizando o exemplo dos cães, temos alguns exemplos: latir, correr, pular. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

70. Abstração é a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou

acidentais. Em modelagem orientada a objetos, uma classe é uma abstração de entidades existentes no domínio do sistema de software. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

71. Generalizar uma classe, sob o conceito de herança, indica dizer que estamos centralizando as características comuns às classes que precisam herdá-la. As classes que herdam essa classe base, têm como objetivo especializá-la; dando assim um formato específico e característico de seu conteúdo, contudo ao utilizarmos uma classe especializada teremos acesso, tanto a sua estrutura em particular, quanto a estrutura da classe mais genérica que está sendo herdada com a finalidade de aplicação dos conceitos de generalização e especialização. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

72. Polimorfismo é a capacidade de dar a um método ou a um objeto, instância de uma classe, a possibilidade de transformação em outro tipo. Permitir que um objeto assuma "várias formas" ou permitir que seja reescrito através de herança e uso de regras polimórficas, por ser em um momento inicialmente abstrato, faz com que o polimorfismo se torne uma boa prática de programação. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

73. O encapsulamento permite ocultar a complexidade do código. Não é necessário entender como a operação funciona para poder utilizá-la. Um exemplo clássico é o de um motorista conduzindo um veículo. O motorista não precisa entender de mecânica e saber exatamente como o motor funciona internamente para poder dirigir o automóvel. Os detalhes da “implementação” do motor ficam ocultos do motorista. O encapsulamento protege os dados, permitindo o acesso a eles apenas a partir de métodos específicos e autorizados. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

REFERÊNCIAS

Biblioteca (computação). Disponível em: <[https://pt.wikipedia.org/wiki/Biblioteca_\(computa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Biblioteca_(computa%C3%A7%C3%A3o))>. Acesso em: 20/1/2022.

Caelum Escola de Tecnologia Cursos Online. Disponível em: <<https://www.caelum.com.br/apostila-java-orientacao-objetos/orientacao-a-objetos-basica>>. Acesso em: 20/1/2022.

Case-sensitive. Disponível em: <<https://pt.wikipedia.org/wiki/Case-sensitive>>. Acesso em: 20/1/2022a.

Case-sensitive. Disponível em: <<https://pt.wikipedia.org/wiki/Case-sensitive>>. Acesso em: 20/1/2022b.

Compilador. Disponível em: <<https://pt.wikipedia.org/wiki/Compilador>>. Acesso em: 20/1/2022a.

Compilador. Disponível em: <<https://pt.wikipedia.org/wiki/Compilador>>. Acesso em: 20/1/2022b.

Compilador. Disponível em: <<https://pt.wikipedia.org/wiki/Compilador>>. Acesso em: 20/1/2022c.

CONCURSOS, T. Questão 617938 CEBRASPE (CESPE) - Oficial Técnico de Inteligência. Disponível em: <<https://www.tecconcursos.com.br/questoes/617938>>. Acesso em: 20/1/2022.

COROB-MSFT. Operadores relacionais e de igualdade C. Disponível em: <<https://docs.microsoft.com/pt-br/cpp/c-language/c-relational-and-equality-operators>>. Acesso em: 20/1/2022.

EDUARDO. Operadores Aritméticos em Linguagem C. Disponível em: <<http://linguagemc.com.br/operadores-aritmeticos-em-linguagem-c/>>. Acesso em: 20/1/2022.

EDUARDO. Operadores Lógicos em C. Disponível em: <<http://linguagemc.com.br/operadores-logicos-em-c/>>. Acesso em: 20/1/2022.

Estrutura de seleção. Disponível em: <https://pt.wikipedia.org/wiki/Estrutura_de_sele%C3%A7%C3%A3o>. Acesso em: 20/1/2022.

Linguagem compilada. Disponível em: <https://pt.wikipedia.org/wiki/Linguagem_compilada>. Acesso em: 20/1/2022.

Linguagem interpretada. Disponível em: <https://pt.wikipedia.org/wiki/Linguagem_interpretada>. Acesso em: 20/1/2022a.

Linguagem interpretada. Disponível em: <https://pt.wikipedia.org/wiki/Linguagem_interpretada>. Acesso em: 20/1/2022b.

NEITZKE, N. A. Boas práticas de Programação: Organizando o Código e seus Pacotes. Disponível em: <<https://www.devmedia.com.br/boas-praticas-de-programacao/21137>>. Acesso em: 20/1/2022.

NEIVA. Linguagem C: estruturas condicionais. Disponível em: <<https://br.ccm.net/faq/10200-linguagem-c-estruturas-condicionais>>. Acesso em: 20/1/2022.

DE OLIVEIRA ALVES, G. F. O que são Vetores e Matrizes (arrays). Disponível em: <<https://dicasdeprogramacao.com.br/o-que-sao-vetores-e-matrizes-arrays>>. Acesso em: 20/1/2022a.

DE OLIVEIRA ALVES, G. F. O que são Funções e Procedimentos? Disponível em: <<https://dicasdeprogramacao.com.br/o-que-sao-funcoes-e-procedimentos>>. Acesso em: 20/1/2022b.

Palavra reservada. Disponível em: <https://pt.wikipedia.org/wiki/Palavra_reservada>. Acesso em: 20/1/2022.

Programação C/C++ - Comandos de Repetição. Disponível em: <<https://www.inf.pucrs.br/~pinho/Laprol/ComandosDeRepeticao/Repeticao.html>>. Acesso em: 20/1/2022.

Python/Conceitos básicos/Indentação. Disponível em: <https://pt.wikibooks.org/wiki/Python/Conceitos_b%C3%A1sicos/Indent%C3%A7%C3%A3o>. Acesso em: 20/1/2022a.

Python/Conceitos básicos/Indentação. Disponível em: <https://pt.wikibooks.org/wiki/Python/Conceitos_b%C3%A1sicos/Indent%C3%A7%C3%A3o>. Acesso em: 20/1/2022b.

Questões de Concursos. Disponível em: <<https://www.qconcursos.com/questoes-de-concursos/questoes/026552a2-59>>. Acesso em: 20/1/2022.

Struct - Registros em Linguagem C com struct - Embarcados. Disponível em: <<https://www.embarcados.com.br/struct-registros-em-linguagem-c/>>. Acesso em: 20/1/2022.

Variável (programação). Disponível em: <[https://pt.wikipedia.org/wiki/Vari%C3%A1vel_\(programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Vari%C3%A1vel_(programa%C3%A7%C3%A3o))>. Acesso em: 20/1/2022.

ZANELATO, J. Lógica de Programação - estruturas de repetição. Disponível em: <<https://podprogramar.com.br/logica-de-programacao-estruturas-de-repeticao/>>. Acesso em: 20/1/2022.

APÊNDICE C – REGRAS

A seguir serão apresentados as regras utilizadas para jogar Prog-poly.

Regras Prog-poly

Instruções para começar a jogar Prog-poly.

1. Para começar a jogar é necessário reunir de 2 à 4 pessoas;
2. Posicionar o tabuleiro e as cartas em uma mesa com espaço adequado e cadeiras para todos os participantes;
Quando os jogadores estiverem prontos, eles(as) escolhem os seus peões, colocando-os no ponto de partida para jogarem os 2 dados, quem tirar o maior número começa, os outros jogadores continuarão depois dele seguindo o sentido horário;
3. O jogo deve começar pela casa que está escrito início, e seguir a seta no sentido horário.
4. É muito importante embaralhar as cartas de Sorte e Revés, e as Perguntas sobre programação e coloca-los de cabeça para baixo no local indicado, no centro do tabuleiro;
5. Em seguida, é hora de distribuir o "dinheiro" cada jogador terá: 10 notas de R\$ 1, 10 de R\$ 5, 10 de R\$ 10, 10 de R\$ 20, 5 de R\$ 50, 5 de R\$ 100 e 1 nota de R\$ 500, cada jogador começa com R\$ 1610, o restante volta para o banco, as outras cartas como de colaboradores, empresas, companhias também.
6. Somente uma pessoa pode ser o banqueiro, e se quiser jogar terá que tomar cuidado para não misturar as notas recebidas com as do banco.

Funcionamento do jogo.

1. Cada um joga os 2 dados em sua vez, percorre o número de casas determinado pelo resultado e cumpre as ordens da casa em que cair.
2. A cada volta completa, ao passar pelo início ganha R\$ 200 como honorários.
3. Ao cair em uma empresa ou companhia, poderá comprá-los do banqueiro, mas primeira precisa responder a uma pergunta, verificar as regras das perguntas abaixo, caso acerte a pergunta pode comprá-la pagando o preço indicado no tabuleiro.

Perguntas.

As perguntas foram cuidadosamente desenvolvidas para ajudar os jogadores a aprenderem a programação, a utilização delas funciona da seguinte forma:

1. Após jogar os dados e andar com seu peão até o número de casas determinado pelo resultado;
2. Se cair em uma empresa ou companhia, o jogador terá que responder uma pergunta, somente se acertar terá a opção de comprar a empresa ou companhia, se errar não poderá realizar compras;
3. Caso caia em uma casa que não seja empresa ou companhia não deve responder as perguntas;
4. Cada pergunta deve ser lida em voz alta para todos jogadores apreenderem, e deve ser respondida no máximo em 30 segundos, caso ultrapasse o tempo será considerado que errou e não poderá efetuar compras;
5. Todas as respostas estão no gabarito, os jogadores adversários irão conferir se quem respondeu acertou, o jogador que respondeu poderá conferir o gabarito somente depois que responder.
6. Para não ficar repetindo as mesmas perguntas, após responde-las devem ser guardadas e só pode voltar depois de todas as respostas serem utilizadas, para novamente embaralha-las e retornar ao tabuleiro.

Dupla Sorte.

Toda vez que tirar uma dupla nos dados (1 e 1, 2 e 2, 3 e 3 e assim por diante), o jogador poderá jogar novamente. Se tirar uma 2ª dupla poderá jogar novamente. Caso tirar 3ª dupla seguida você irá para a prisão e terá que colocar seu peão na prisão.

Quando cai na prisão do jogo Prog-poly?

Ir para a prisão no Prog-poly acontece de 3 formas:

1. Quando cair na casa "VÁ PARA A PRISÃO";
2. Quando tirar 3 duplas seguidas exemplo: (1 e 1, 2 e 2, 3 e 3 e assim por diante);
3. Ao tirar uma carta de revés que o mande para prisão.

O jogador que estiver na prisão ficaram no máximo por 2ª rodadas. Para sair é necessário:

1. Tenta tirar uma dupla nos dados durante 2ª rodadas, caso tire uma dupla devesse andar o número de casas que tirou nos dados, mas não poderá jogar novamente.
2. Na 3ª rodada poderá pagar R\$ 50 ao banco e sair.
3. Ter uma carta Habeas corpus que tirou no Sorte ou Reves.

Caso não tenha dinheiro para pagar sua saída poderá hipotecar uma propriedade para sair.

Como acrescentar colaboradores e negociar propriedades no Prog-poly.

Um dos grandes segredos de jogar uma boa partida de Prog-poly é construir rápido suas propriedades. Para isso, você precisa ter conquistado um grupo de empresas da mesma cor. Ao fazer isso, você poderá, então, acrescentar colaboradores pagando ao banqueiro os preços indicados nos títulos.

Atenção: para acrescentar colaboradores:

1. Comece acrescentando 1 colaborador em cada empresa da mesma cor, somente após todas as empresas estiverem com 1 colaborador, poderá colocar o segundo, o mesmo vale para o terceiro e quarto colaborador.
2. Pode acrescentar até 4 colaboradores e após isto transformar em uma Empresa Matriz. Lembre-se da regra de ouro: você só pode acrescentar colaboradores depois de ter um colaborador em cada empresa. Quando cair em empresas ou propriedades de outro dono, você tem que pagar aluguel e taxas correspondentes, caso não possua nenhum colaborador só será paga a diária do colaborador.

Vendas e trocas de Empresas e Companhias

Prog-poly permite negociações para compra, venda ou troca de suas empresas ou companhias.

Para poder trocar sua propriedade os jogadores devem esperar pela sua vez de jogar e assim propor a compra, venda ou troca.

A compra, venda ou troca de empresas ou companhias, desde que os jogadores entre em comum acordo.

Caso alguma empresa possua colaboradores ou empresa matriz estes terão que ser vendidos ao banco pela metade do preço antes da compra, venda ou troca da mesma.

Como termina o jogo!

O jogo Prog-poly é, antes de mais nada, um jogo que dá algumas lições importantes sobre as finanças do mundo real. Você tem que estar com os seus pagamentos em dia e no jogo, caso não tenha, você terá que:

- Vender os colaboradores e Empresa Matriz pela metade do preço pago.

- Hipotecar ou vender suas propriedades.

Não conseguiu mesmo assim quitar suas dívidas? Você entra em falência e sai do jogo. Seu dinheiro vai para o seu credor. O jogo termina quando sobrar somente um jogador, tendo os outros declarado falência. Após isto é só somar os valores ganhos com as notas, empresas, propriedades, casas e hotéis e calcular os lucros de sua Vitória.

Como o jogo Prog-poly nos ajuda.

Além de horas de diversão, uma boa partida de Prog-poly vai ajudar você, e a turma de amigos em diversas áreas de conhecimento. O jogo ajudar os alunos, no desenvolvimento das habilidades e competências relacionadas a Introdução a Linguagem de Programação e a Introdução a Engenharia de Software. Utilizando o Prog-poly você irá se familiarizar com os conceitos de Engenharia de Software e para estudar os conceitos da Introdução a Linguagem de Programação C. O jogo também ajuda a adquirir habilidades e competências educação financeira, Prog-poly o ajudara a apreender como é importante planejar e investir corretamente. O banqueiro pode também exercer a função de líder do grupo, organizando o jogo e fazendo as relações funcionarem mais tranquilamente. Além disso, por terem que interagir, negociar e aprender a ouvir o outro, dá para trabalhar também a socialização. Isso sem falar, claro, nas lições de estratégia e pensamento analítico. Escolher que propriedades comprar e vender, a rodada certa para fazer um investimento, escolher seu portfólio de empresas ou companhias e tudo mais que o jogo oferece.

Quantidade total de Cartas.

Dinheiro: possui 560 notas, sendo 80 notas de cada valor R\$ 1, R\$ 5, R\$ 10, R\$ 20, R\$ 50, R\$ 100, R\$ 500. 32 colaboradores, 12 empresas matriz, 29 títulos de propriedades, 18 cartões sorte, 18 cartões revés, 6 cartões pares, 6 cartões ímpares, 6 peões, 2 dados, 1 tabuleiro.

APÊNDICE D – MANUAL DO USUÁRIO

A seguir será apresentado o manual do usuário do Software de Análise de Desempenho dos Jogadores do Prog-Poly.



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
ENGENHARIA DE SOFTWARE

LUCIANO ROVANNI DO NASCIMENTO

Manual do Usuário - Software de Desempenho dos Jogadores do Prog-Poly

CORNÉLIO PROCÓPIO
2021

RESUMO

O aprendizado de linguagem de programação é considerado difícil pelos estudantes dos cursos de Ensino Superior em Computação, os principais motivos para este fato são: abstração, raciocínio lógico, rigor no uso das linguagens de programação. O Jogo tem o objetivo de proporcionar aos estudantes de Linguagem de Programação e Engenharia de Software, maior engajamento e motivação quando estiverem estudando Introdução à Engenharia de Software (IES) e Introdução à Linguagem de Programação C (ILPC). O jogo, denominado Prog-poly, permitirá aos alunos que desenvolvam as habilidades e competências relacionadas à IES e ILPC. Esta ferramenta educacional poderá ser aplicável nas disciplinas que possui o estudo de Engenharia de Software ou Linguagem de Programação C, além de permitir a integração com tecnologias e recursos didáticos já existentes. Para auxiliar os participantes a monitorar seu desempenho durante o jogo, foi desenvolvido o **Software de Desempenho dos Jogadores do Prog-Poly**, ele possibilita que os participantes acompanhem seu desempenho no jogo, possibilitando verificar quais os erros e acertos das questões, podendo estudar o conteúdo que tiveram maior dificuldade. Os dados podem ser exportados para uma planilha do Excel.

Palavras-chave: Lúdico. Linguagem de Programação. Ensino. Jogo. Gamificação. Monopoly.

SUMÁRIO

1	Descrição.....	3
1.1	Menu.....	3
1.2	Cursos.....	4
1.3	Jogadores.....	4
1.4	Perguntas.....	4
1.5	Partidas.....	5
1.6	Respostas.....	6
2	Navegação pelas telas.....	6
3	Exportando a base histórica para uma planilha do Excel.....	7

1 Descrição

Manual do Usuário – Software de Desempenho dos Jogadores do Prog-Poly

O Software foi desenvolvido para os participantes armazenarem os dados das partidas e assim analisar seu desempenho de aprendizagem em cada uma das partidas. Esta é primeira versão do software.

1.1 Menu

Na tela inicial representada pela figura 01, exibe-se o menu principal, contendo as opções **Cursos**, **Jogadores**, **Perguntas**, **Jogos**, **Respostas**.

Prog-Poly - Software de desempenho dos Jogadores

[Objetivo do Jogo e do Software](#)

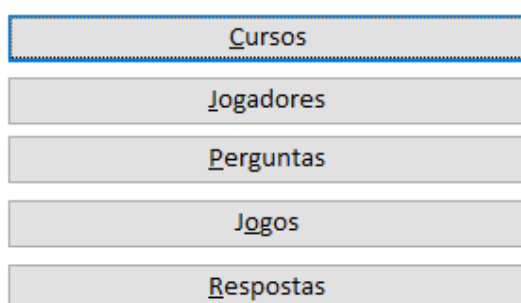
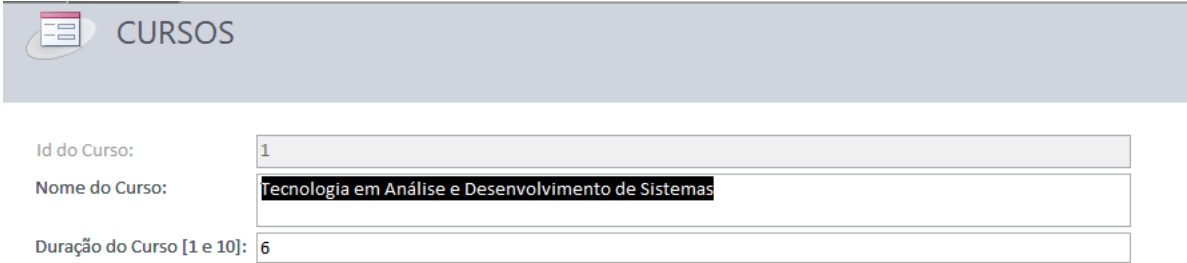


Figura 01: Tela de menu.

1.2 Cursos

Na tela de cursos, primeira opção do menu principal representada na figura 02, é cadastrado o curso dos jogadores, o cadastro é realizado através de dois campos: **Nome do Curso**, **Duração do curso**.

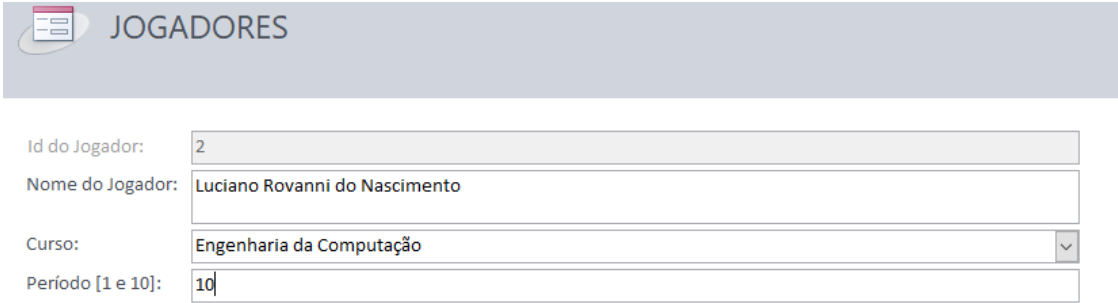


Id do Curso:	1
Nome do Curso:	Tecnologia em Análise e Desenvolvimento de Sistemas
Duração do Curso [1 e 10]:	6

Figura 02: Tela de cadastro dos cursos.

1.3 Jogadores

Na tela de jogadores, segunda opção do menu principal, representada na figura 03, é cadastrados os jogadores, através de dois campos: **Nome do Jogador**, **Curso** onde é selecionado o curso pré-cadastrado seção 1.1, e cadastrado o **Período**.



Id do Jogador:	2
Nome do Jogador:	Luciano Rovanni do Nascimento
Curso:	Engenharia da Computação
Período [1 e 10]:	10

Figura 03: Tela de cadastro dos jogadores.

1.4 Perguntas

Na tela de perguntas, terceira opção do menu principal, representado na figura 04, é cadastrado as perguntas sobre programação utilizadas no jogo, através de cinco campos: **Questões** onde é feito o cadastro da pergunta;

Errado, **Correto**, servem para indicar se é um pergunta com resposta correta ou errada;

Conteúdo referindo ao tema de programação relacionado à pergunta;

Fonte dos dados que foi utilizado para criar a pergunta.

PERGUNTAS

Id da Pergunta: 1

Questões : Um compilador é um programa de computador (ou um grupo de programas) que, a partir de um código fonte escrito em uma linguagem maquina, cria um programa semanticamente equivalente, porém escrito em outra linguagem de alto nível, código objeto.

Errado: 1

Correto:

Conteúdo: Compiladores

Fonte: <https://pt.wikipedia.org/wiki/Compilador#:~:text=Um%20compilador%20%C3%A9%20um%20programa,e m%20outra%20linguagem%2C%20c%C3%B3digo%20objeto.>

Figura 04: Tela de cadastro das perguntas.

1.5 Partidas

Na tela de partidas, quarta opção do menu principal, representada na figura 05, é cadastrado o vencedor do jogo, a duração e o patrimônio, o cadastro é realizado através de 6 campos: **Data**, **Hora**, **Duração Prevista/Realizada**, **Vencedor**, **Patrimônio**. **Configurações** onde é selecionado os jogadores da partida, já cadastrados na tela Jogadores, como pode visto na seção 1.3 deste manual.

PARTIDAS

Id do Jogo: 1

Data: 22/08/2021

Hora: 10:50

Duração Prevista/Realizada: 2

Vencedor: Luciano

Patrimônio: R\$ 10.000,00

CONFIGURAÇÕES

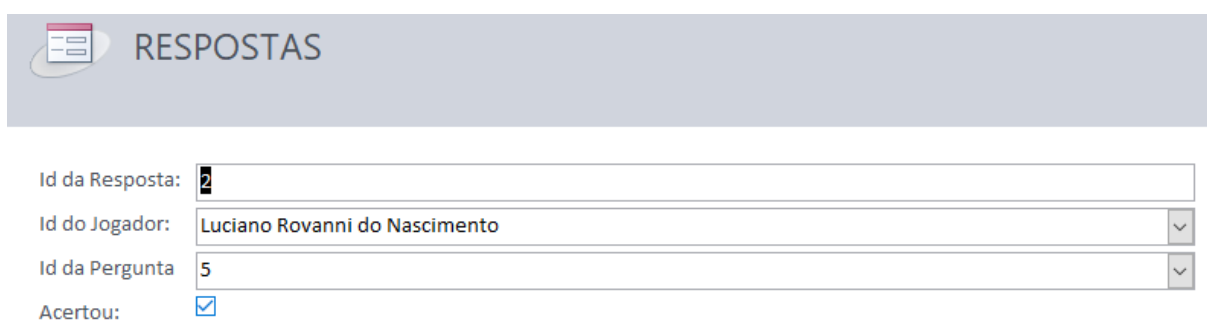
Id da Configuração	Id do Jogo	Id do Jogador
3	1	Luciano Rovanni do Nascimento
4	1	Luciano Rovanni do Nascimento
7	1	Luciano Rovanni do Nascimento
* (Novo)	1	

Registro: 1 de 3 | Sem Filtro | Pesquisar

Figura 05: Tela de cadastro dos jogos.

1.6 Respostas

Na tela de respostas, quinta opção no menu principal, representada na figura 06, é cadastrado cada resposta que o jogador respondeu e se ele acertou. O cadastro é realizado através de quatro campos: **Id da Resposta**, **Id do Jogador**, onde é selecionado um jogador já pré-cadastrado, **Id da Pergunta**, **Acertou**.



RESPOSTAS

Id da Resposta: 2

Id do Jogador: Luciano Rovanni do Nascimento

Id da Pergunta: 5

Acertou:

Figura 06: Tela de cadastro dos jogos.

2 Navegação pelas telas

Barra de navegação das telas




3 de 3 - Exibe qual o registro exibido e o total de registros cadastrados, neste exemplo o registro atual é o terceiro de um total de três registros.

⏪ - Navega para o primeiro registro.

⏩ - Retorna um registro.

▶ - Avança um registro.

⏭ - Avança para o último registro.

 - Criar registro novo, preencher as informações e pressionar a tecla enter para gravar o registro.

- Pesquisa qualquer informação, texto, número, parte do texto, utiliza-se o campo preenchendo com a informação a ser localizada, basta apenas digitar e a busca é iniciada automaticamente.

3 Exportando a base histórica para uma planilha do Excel

Para exportar os dados para uma planilha do Excel, para analisar o desempenho dos jogadores. Siga os seguintes passos:

1. Feche o menu principal;
2. Abra o painel de navegação
3. Selecione na opção consultas, base história para análise de desempenho;
4. Agora clique no na barra de menu dados externos;

5. Clique no botão para exportar para a planilha do Excel.



ANEXO A – RESPOSTAS DO QUASE-EXPERIMENTO

A seguir serão será apresentado as respostas do quase-experimento utilizando o formulário *on-line* do *Google Forms*.

Prog-poly: Ensino de algoritmos através de jogos educacionais baseados no Monopoly

3 respostas

[Publicar análise](#)

Perguntas Sobre o Desenvolvimento do Jogo!

Em qual curso de Graduação você é Formado?

3 respostas

Analise e Desenvolvimento de Sistemas

Sistemas de informação

Engenharia de Computação

Quanto tempo de experiencia você possui em programação?

3 respostas

6 anos

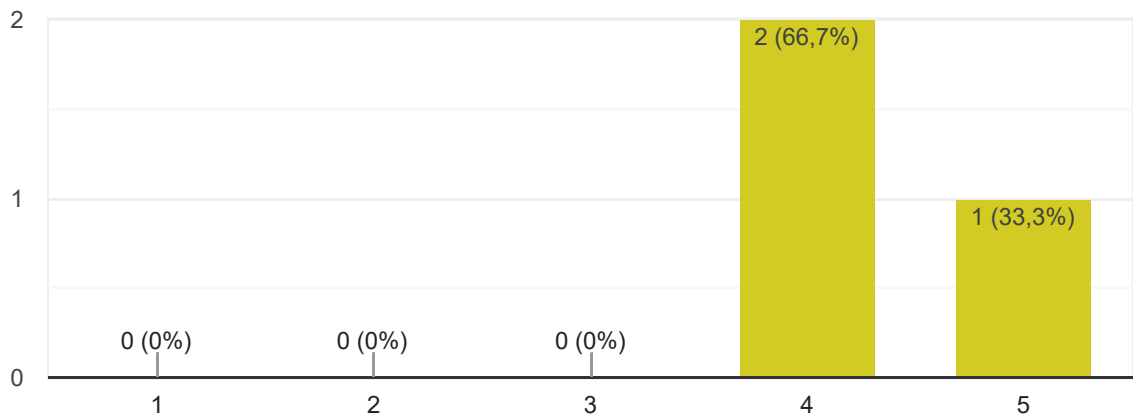
12 anos

1 ano

A qualidade da arte e das ilustrações dos artefatos são:

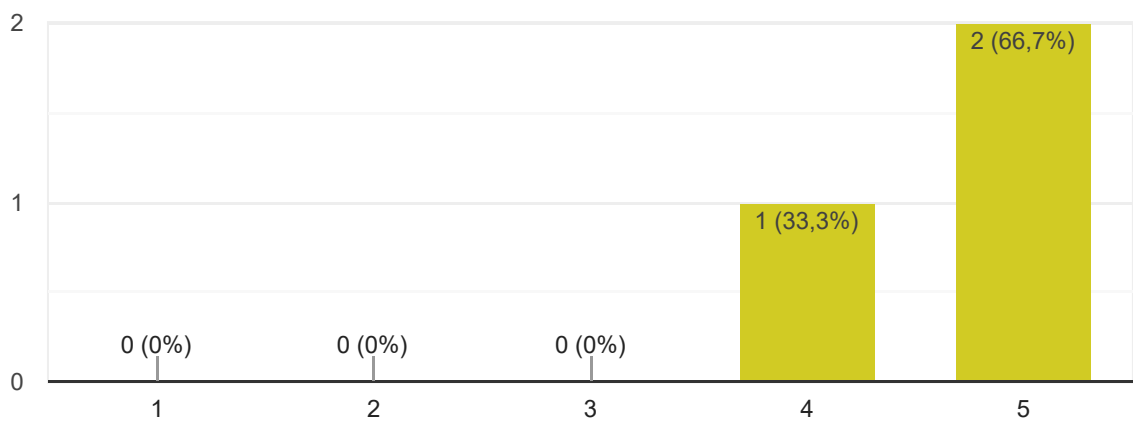


3 respostas



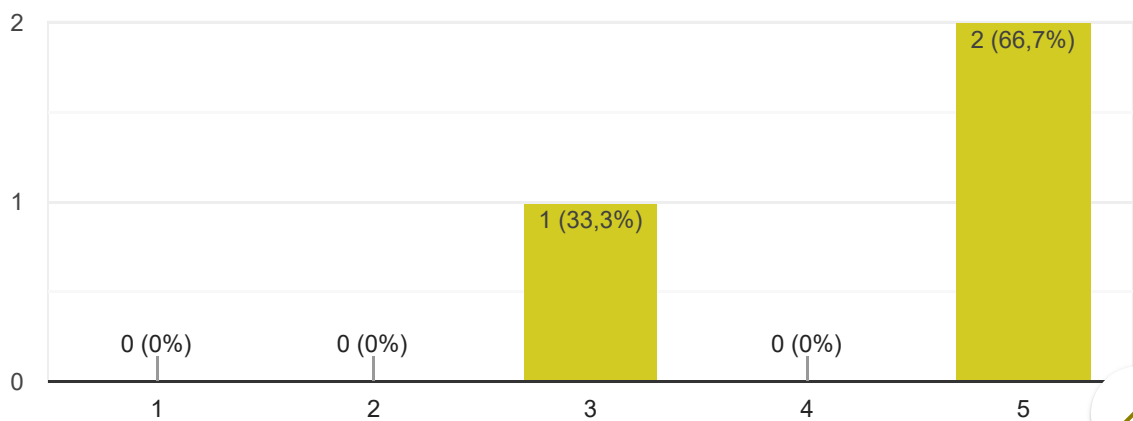
A mecânica do jogo pode ser considerada:

3 respostas



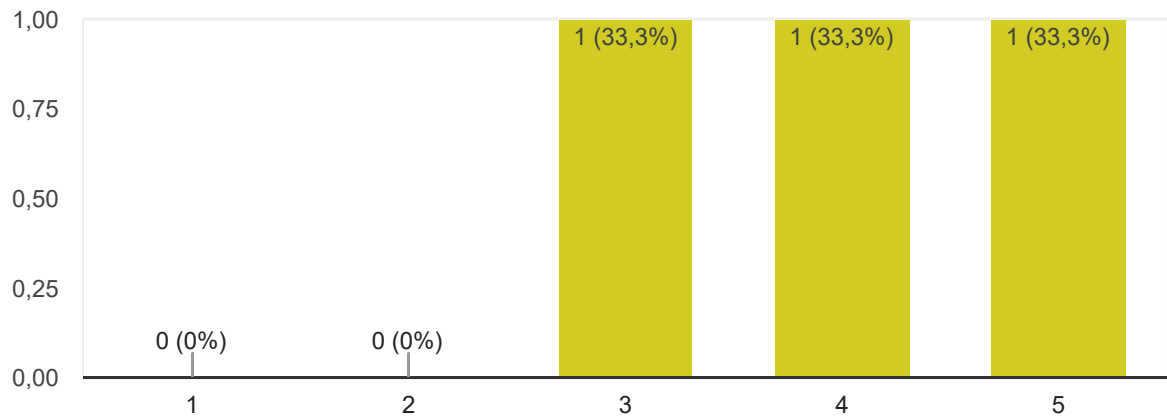
O fator sorte do jogo pode ser considerado:

3 respostas



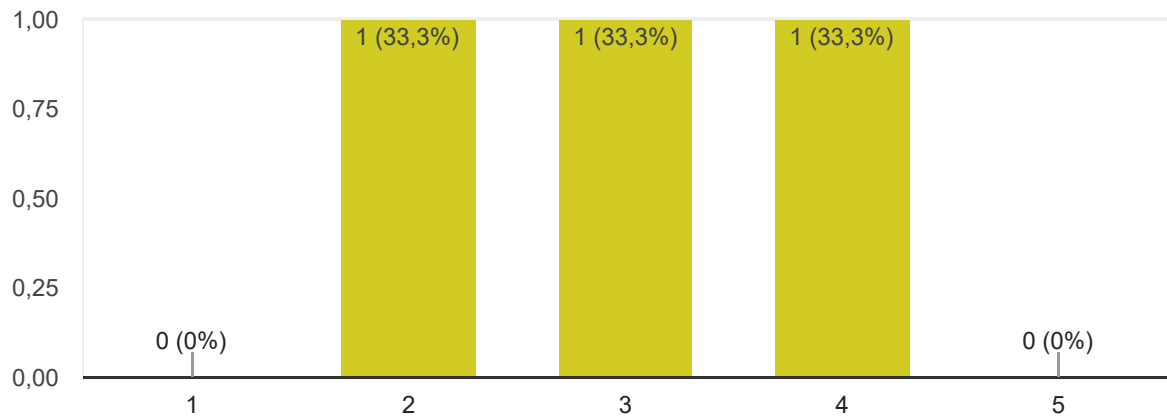
Qual a profundidade do material de algoritmos disponibilizada no jogo?

3 respostas



Qual a dificuldade do jogo?

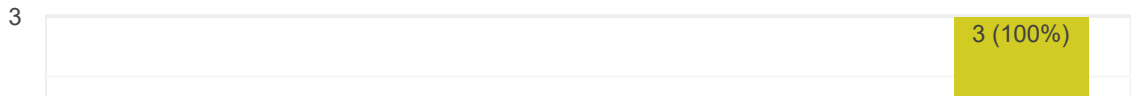
3 respostas



Classifique o grau de interatividade do jogo:

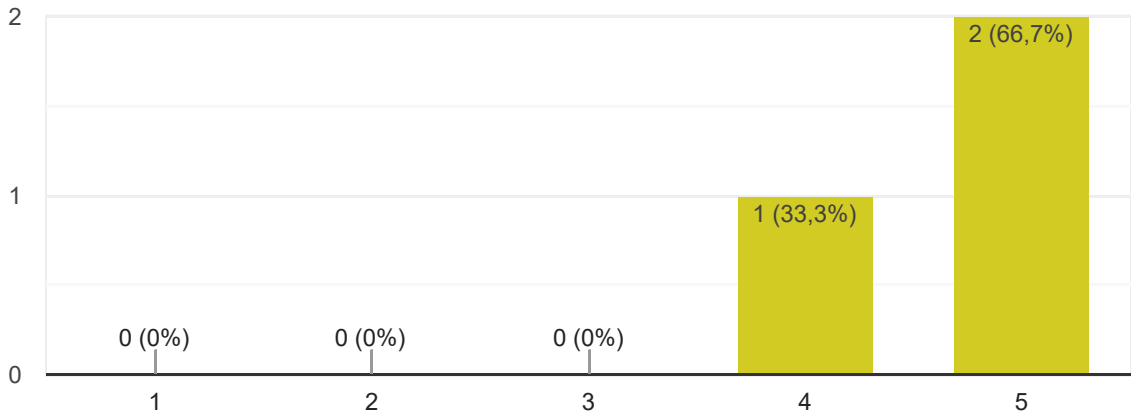
3 respostas





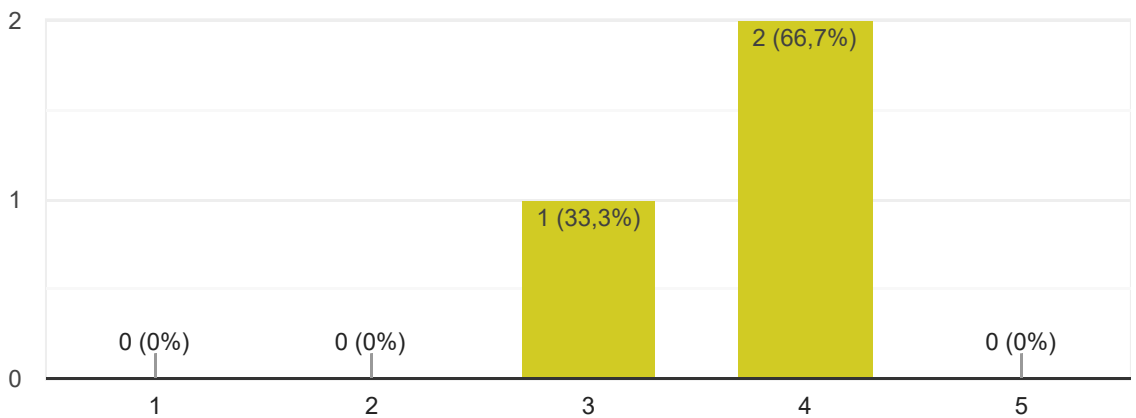
Qual o grau de diversão do jogo?

3 respostas



O manual do jogo pode ser considerado:

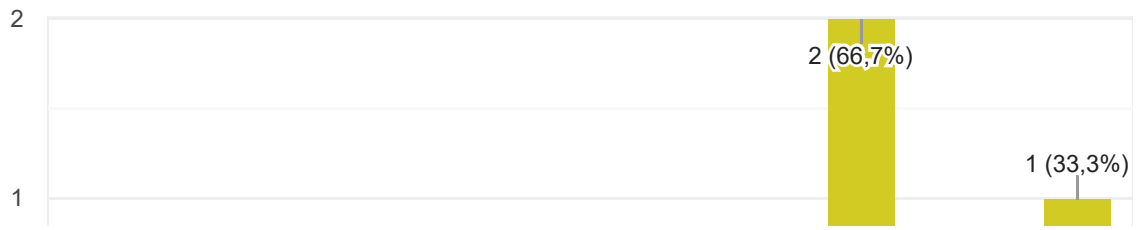
3 respostas



Qual o grau de aprendizado de algoritmo o jogo fornece?

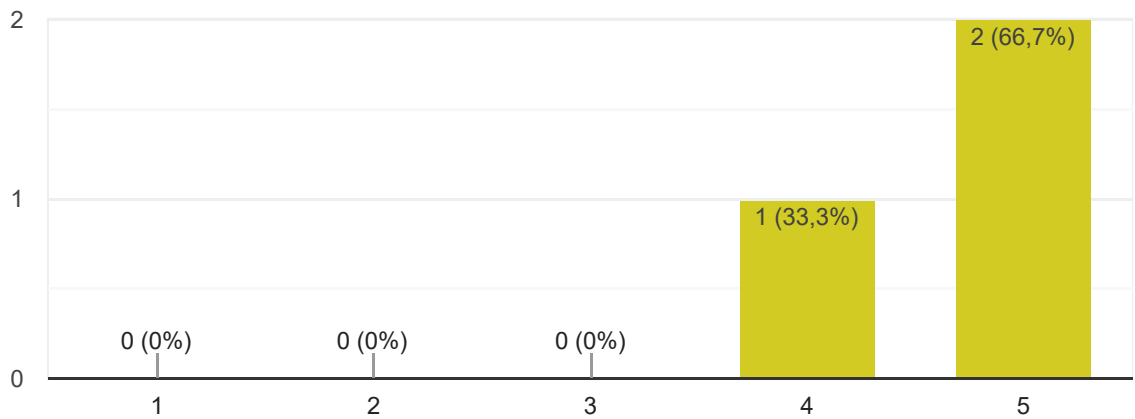
3 respostas





Qual foi o Grau de certeza: de sua resposta na questão acima?

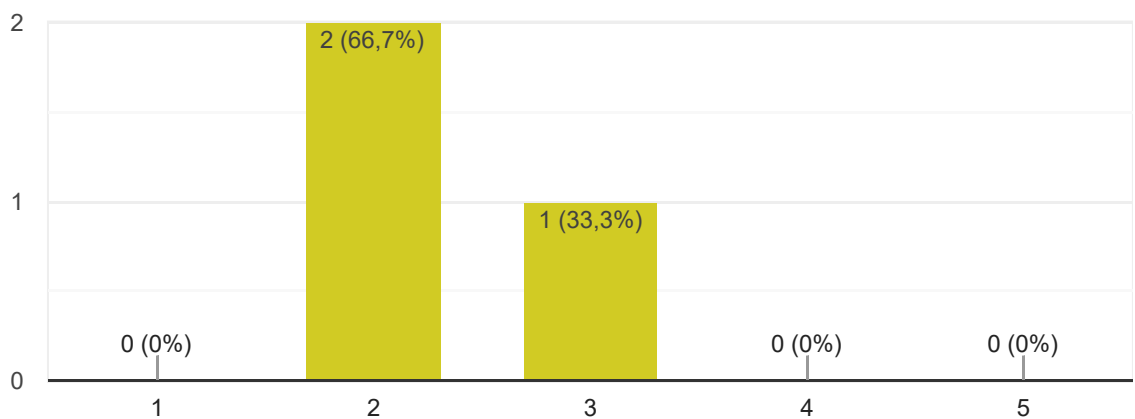
3 respostas



Perguntas relacionadas ao Prog-poly: Ensino de algoritmos.

O jogo prog-poly pode proporcionar o aprendizado de algoritmos a alunos do ensino médio que não tiveram contato com este conceito?

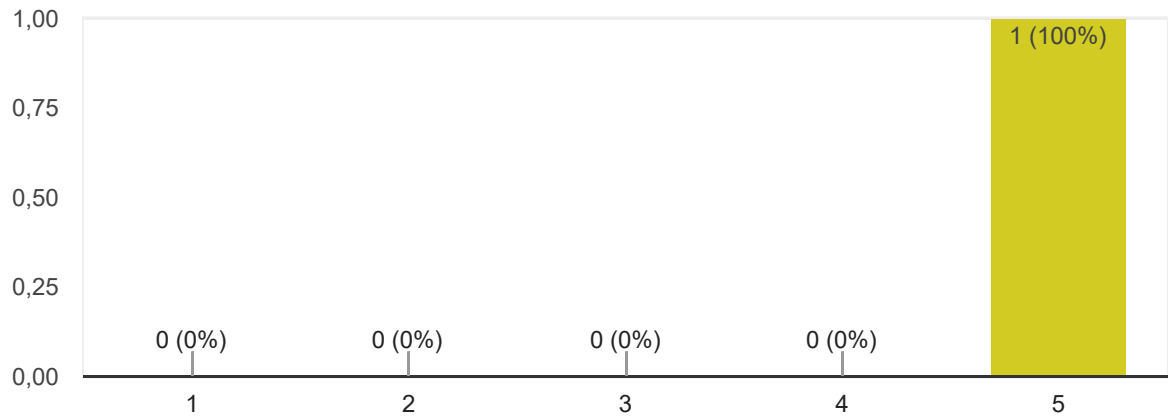
3 respostas



Qual foi o Grau de certeza: de sua resposta na questão acima?

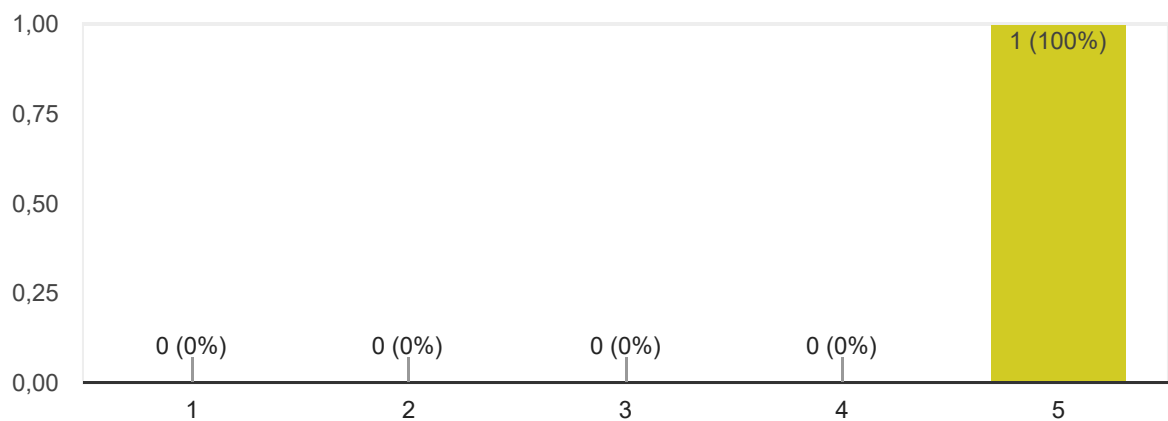
1 resposta





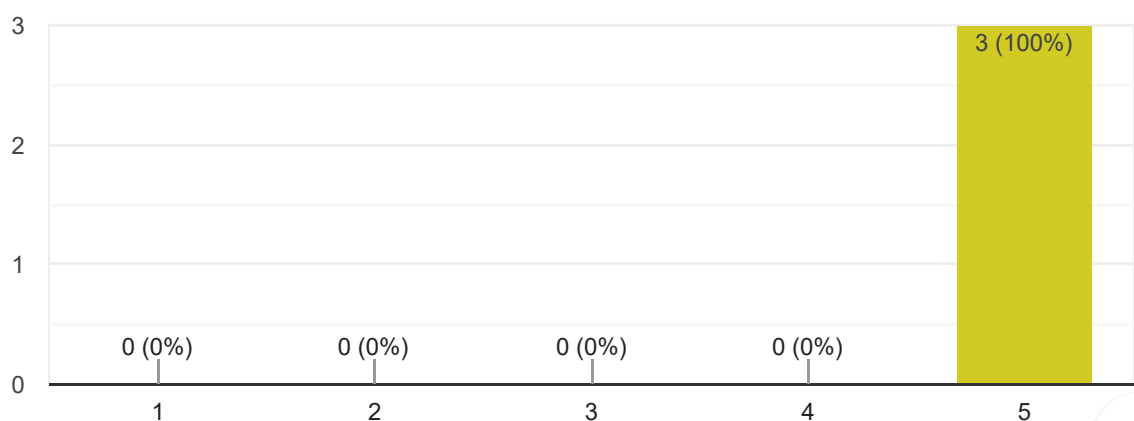
O jogo prog-poly pode proporcionar o aprendizado de algoritmos a alunos da Graduação que já tiveram contato com este conceito?

1 resposta



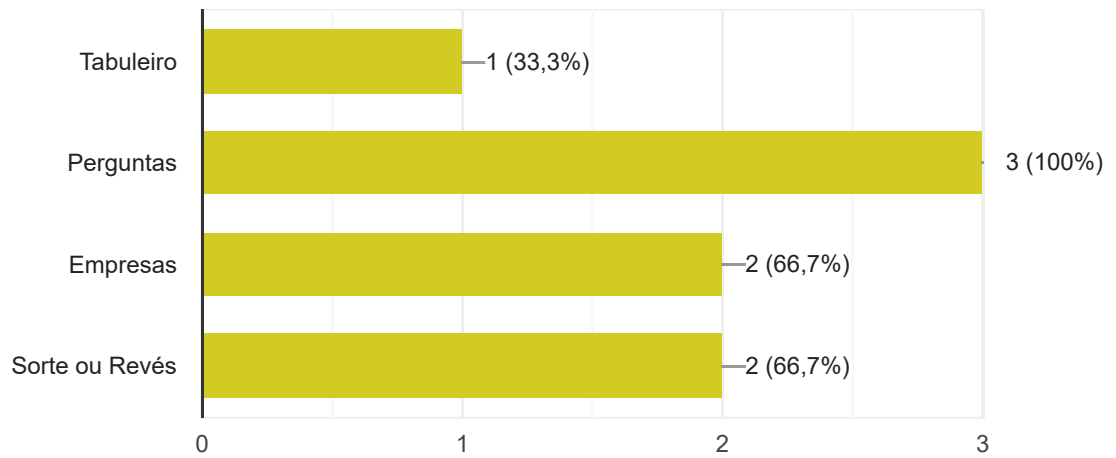
Qual foi o Grau de certeza: de sua resposta na questão acima?

3 respostas



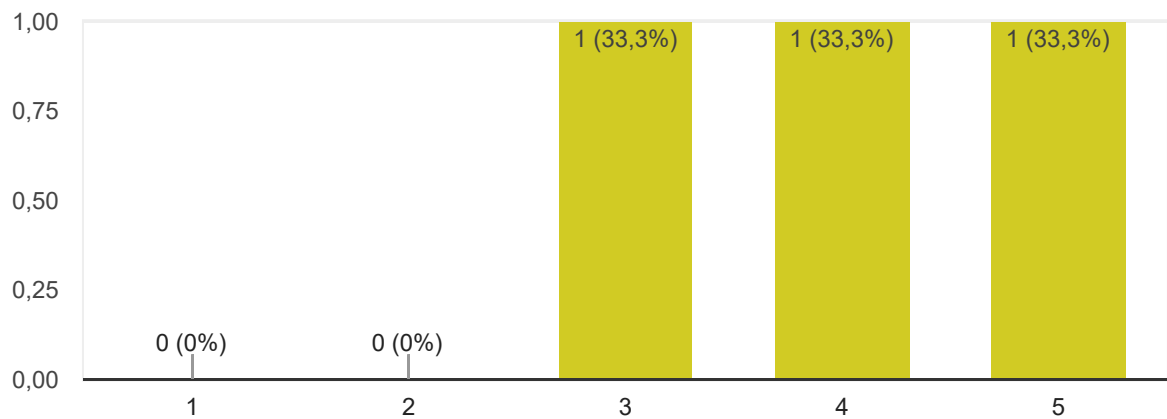
Com relação ao jogo: Quais dos itens abaixo ajudam a entender conceitos de algoritmos?

3 respostas



Qual foi o Grau de certeza: de sua resposta na questão acima?

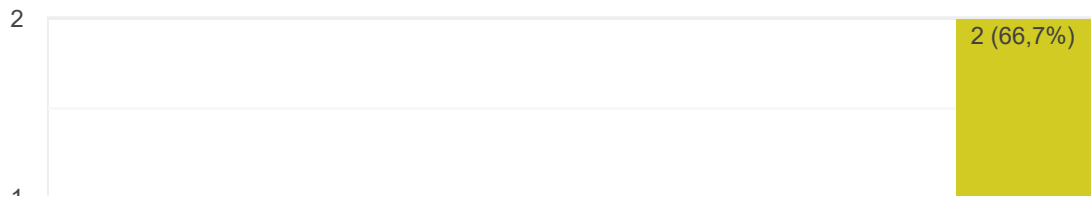
3 respostas



Sobre a distribuição das empresas no tabuleiro: Você obteve conhecimento correlacionando o nome das empresas com os conceitos relacionados a algoritmos e estrutura de dados?

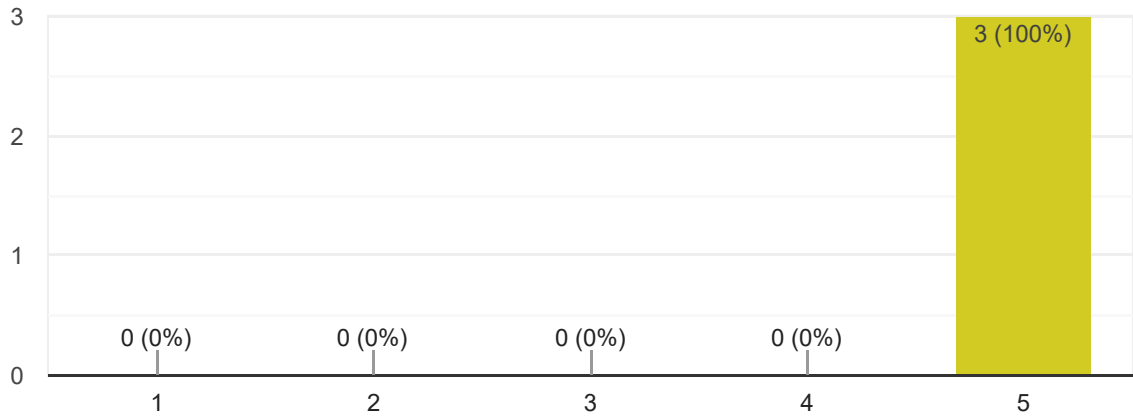
3 respostas





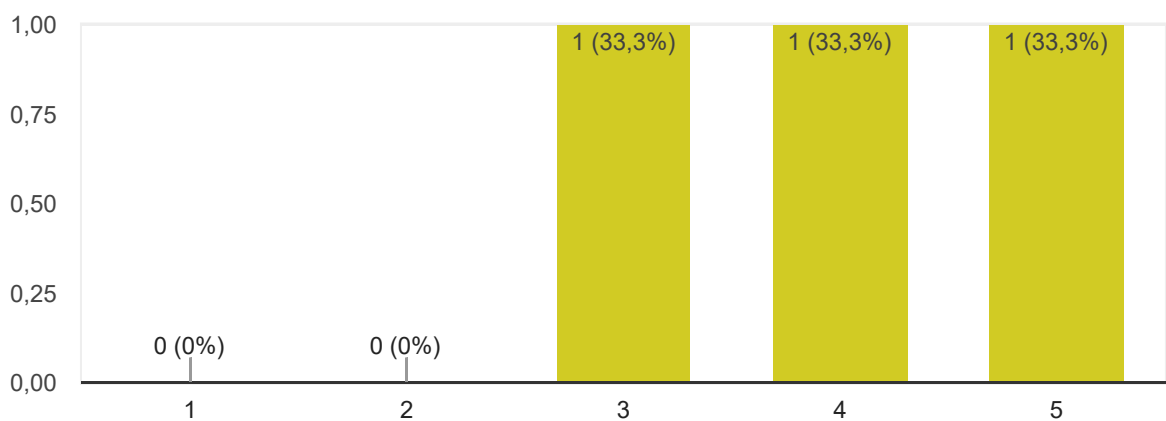
Qual foi o Grau de certeza: de sua resposta na questão acima?

3 respostas



Os cartões que contém as perguntas utilizadas no Prog-poly podem ajudar o aprendizado de conceitos de algoritmos e estrutura de dados?

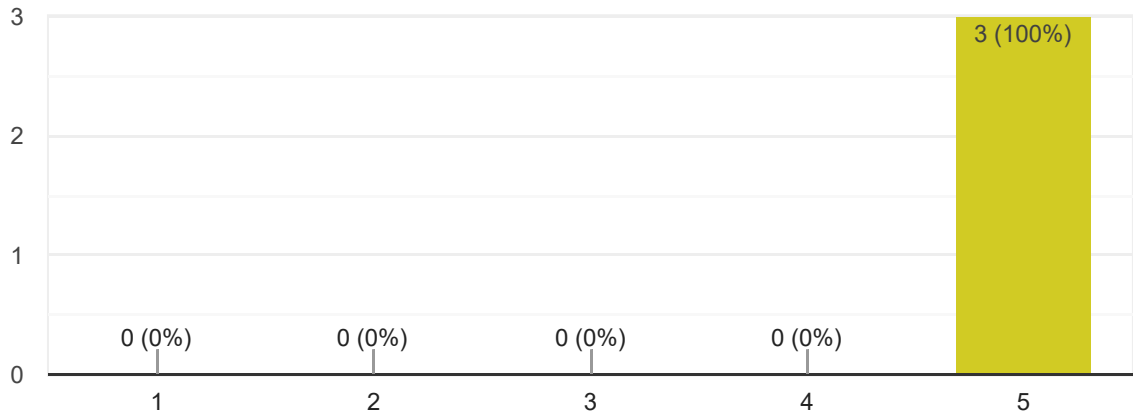
3 respostas



Qual foi o Grau de certeza: de sua resposta na questão acima?

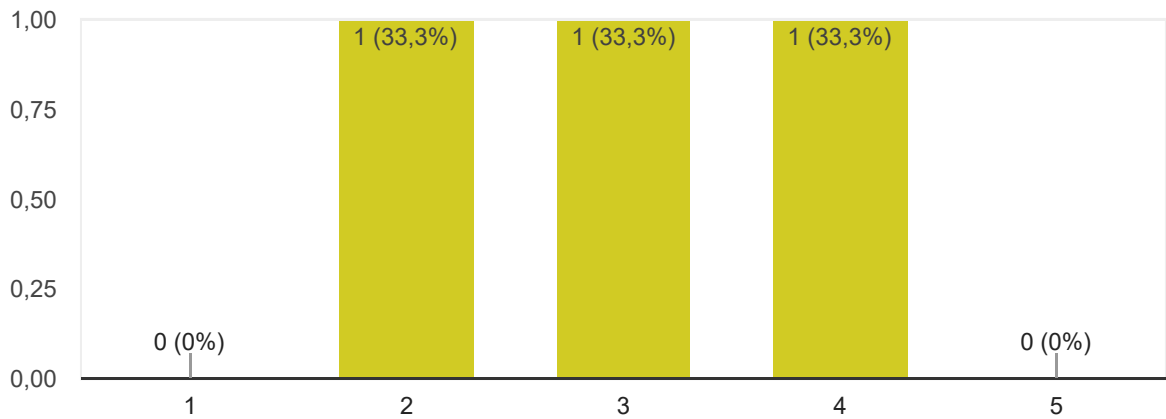
3 respostas





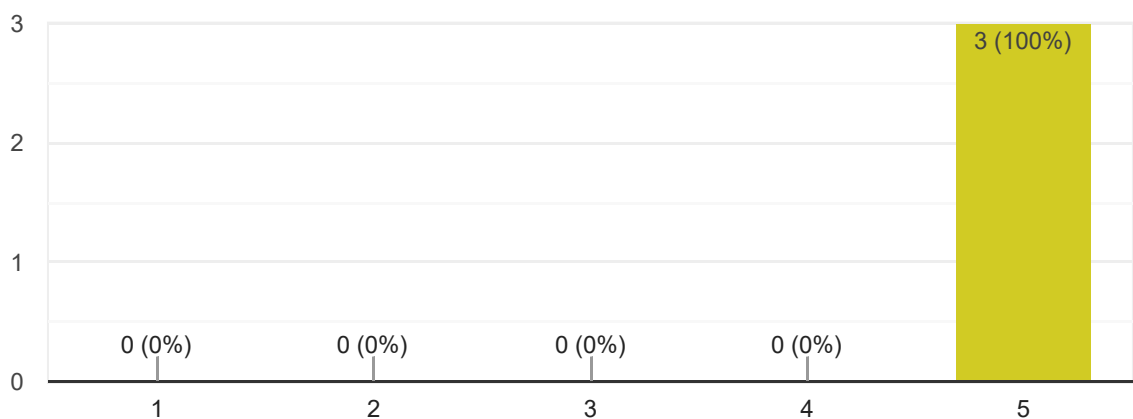
O Tabuleiro de Prog-poly possibilita algum aprendizado de algoritmos?

3 respostas



Qual foi o Grau de certeza: de sua resposta na questão acima?

3 respostas

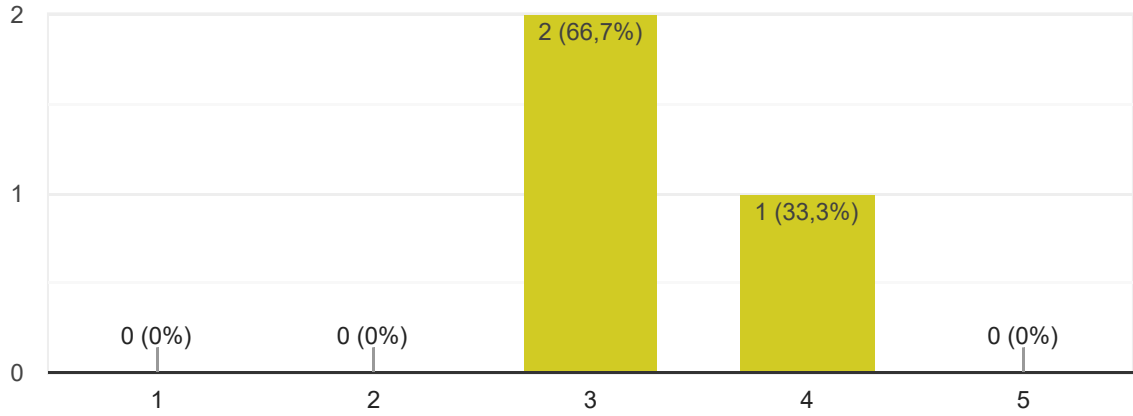


Os nomes das Empresas do Prog-poly possibilita algum aprendizado de



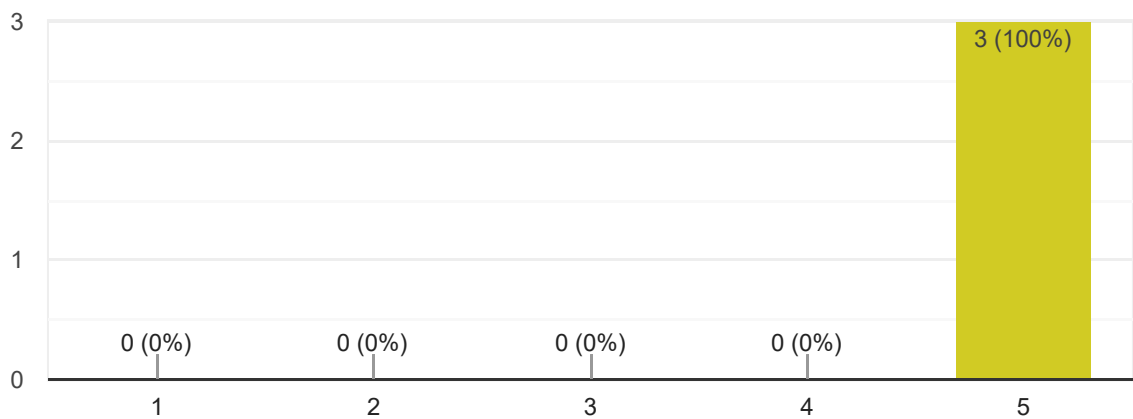
algoritmos?

3 respostas



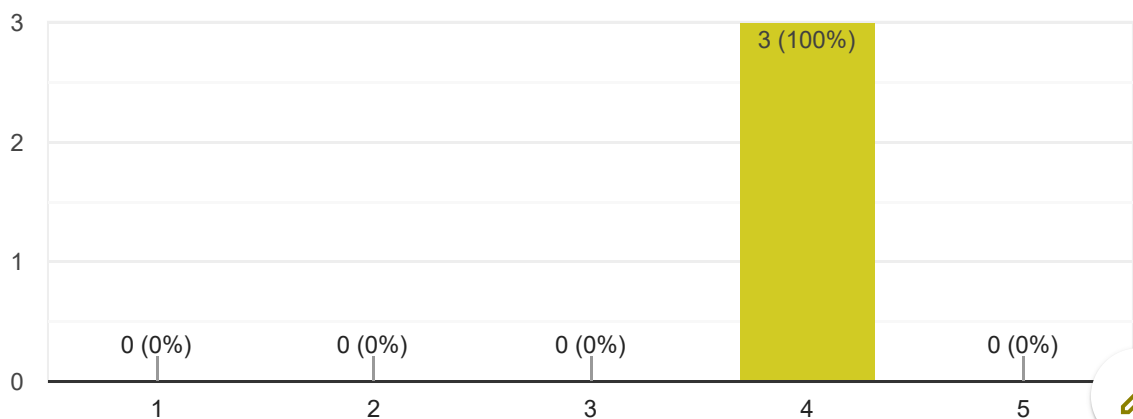
Qual foi o Grau de certeza: de sua resposta na questão acima?

3 respostas



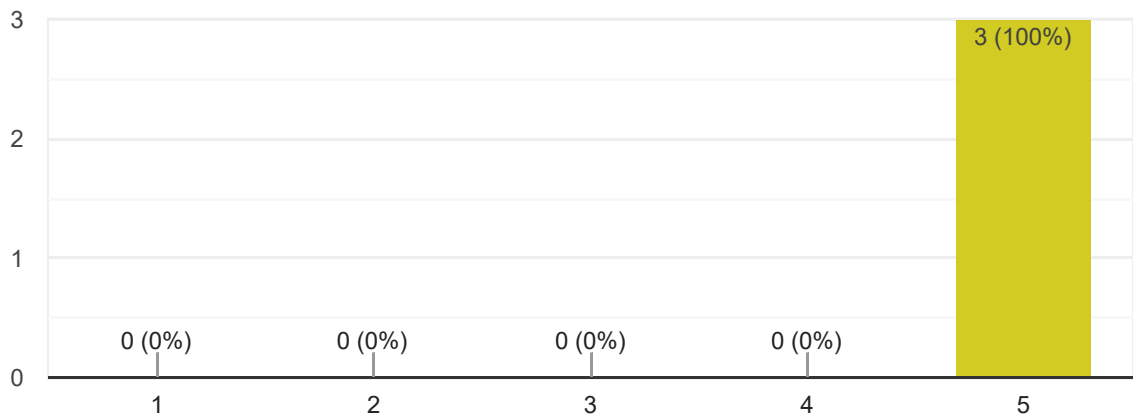
As cartas Sorte ou Revés possibilita algum aprendizado de algoritmos?

3 respostas



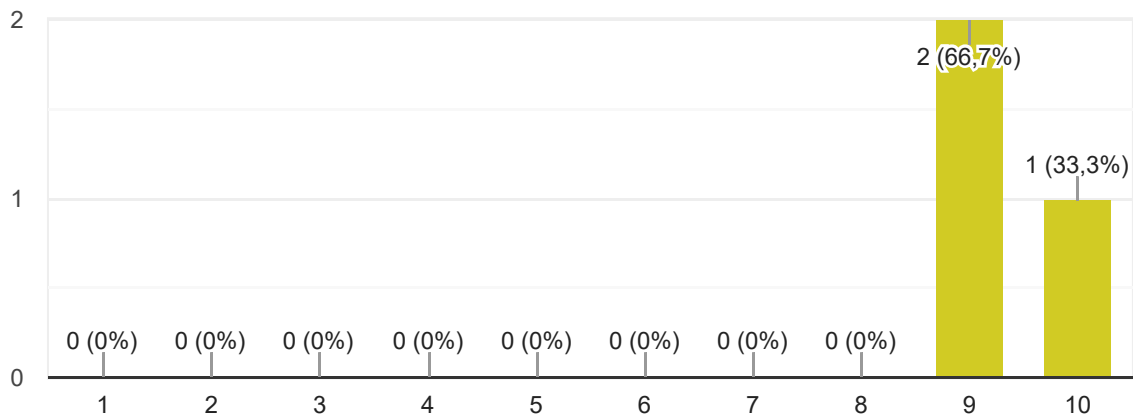
Qual foi o Grau de certeza: de sua resposta na questão acima?

3 respostas



Prog-poly motiva a aprendizagem de algoritmos?

3 respostas



Qual foi o Grau de certeza: de sua resposta na questão acima?

3 respostas



3

3 (100%)

Observações:

3 respostas

Corrigir a posição dos textos na fila que fica entre a área de descanso e o "vá para a cadeia", algumas cartas de empresas não apresentam o título completo do nome da empresa que está no tabuleiro podendo causar confusão (duas empresas ao lado da área de descanso)

Parabéns pelo trabalho, projeto que pode agregar muito aos alunos da graduação em cursos de computação. Acredito que para o ensino médio as perguntas estão com grau de profundidade muito alto.

Jogo muito bem montado! No entanto, é preciso ter noções de computação para jogar.

Sugestões:

2 respostas

Revisar perguntas e respostas, pois encontramos divergência ou ambiguidade, em mais de uma questão; melhorar o manual de instruções pois quem nunca jogou esse tipo de jogo poderá ter dificuldades; pintar os números de branco para facilitar a visualização; o pagamento de impostos pode atrapalhar o jogo pois precisa contar todo o dinheiro do jogador. Cobraram a taxa menor porém com valor fixo; enviar uma cópia final do jogo para todos que ajudaram na homologação do protótipo (RS).

Mudar o foco do público alvo do jogo de aprendizes iniciantes de computação para estudantes ou graduados em computação.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários



