UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE RUSSAS

CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

GABRIEL RODRIGUES RAMALHO GURGEL

UTILIZAÇÃO DA FERRAMENTA GAMIFICADA CODECOMBAT PARA O ENSINO DE PROGRAMAÇÃO

RUSSAS

GABRIEL RODRIGUES RAMALHO GURGEL

UTILIZAÇÃO DA FERRAMENTA GAMIFICADA CODECOMBAT PARA O ENSINO DE PROGRAMAÇÃO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Orientador: Prof. Dr. Markos Oliveira Freitas

RUSSAS

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

R613u Rodrigues Ramalho Gurgel, Gabriel.
Utilização da ferramenta gamificada CodeCombat para o ensino de programação / Gabriel Rodrigues Ramalho Gurgel. – 2023.
106 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Engenharia de Software, Russas, 2023.
Orientação: Prof. Dr. Markos Oliveira Freitas.

1. gamificação. 2. ensino. 3. programação. 4. computação. 5. CodeCombat. I. Título.
CDD 005.1

RESUMO

A programação de computadores se tornou algo fundamental para o funcionamento da sociedade atual, gerando uma alta demanda de profissionais da área. Para suprir essa demanda, foram criados vários cursos de nível técnico e superior de Tecnologia da Informação. Um problema muito recorrente neles é a alta evasão de estudantes sendo motivada por: impossibilidade de permanecer no curso por razões econômicas, vocacionais ou institucionais. O objetivo desta pesquisa é analisar se a utilização de uma ferramenta gamificada escolhida beneficia a motivação para o aprendizado de programação, por meio de experimentos e aplicação de questionários. Neste trabalho, foi feita uma análise de ferramentas que poderiam ser utilizadas para realizar experimentos em disciplinas de programação. A escolha da ferramenta chamada CodeCombat foi baseada nos seguintes motivos: facilidade do uso, presença de elementos gamificados, utilização da linguagem de programação python e disponibilidade em português. Dentre os elementos de gamificação, o CodeCombat contém: criação de avatares, mapa de missões, níveis e conquistas. Após isso, foi feito o planejamento de como a ferramenta seria integrada à disciplina de Fundamentos da Programação ministrada para cursos de graduação do Campus de Russas da Universidade Federal do Ceará. Então foi construído o design do experimento, que foi aplicado no semestre 2023.1, utilizando a turma do curso de Ciência da Computação como grupo experimental e a turma de Engenharia de Software como grupo de controle. Para colher os resultados, foi utilizado o questionário Intrinsic Motivation Inventory (IMI), com o objetivo de avaliar a motivação dos alunos para o aprendizado na disciplina após a utilização da ferramenta. Com os resultados obtidos, concluiu-se que, levando em consideração as limitações da versão grátis da ferramenta, o seu uso não trouxe um grande impacto na motivação dos alunos para a disciplina, mas ainda existe o potencial para o uso de outras formas.

Palavras-chave: gamificação; ensino; programação; computação; CodeCombat.

ABSTRACT

Computer programming became fundamental for current society's functioning, creating a high demand for professionals in the field. To fulfill this demand, many courses of technical and superior level were created. A recurring problem in these courses is the high student dropout rate, being motivated by: impossibility to remain in the course for economic, vocational or institutional reasons. The objective of this research is to analyze if the use of a chosen gamified tool benefits the motivation to learn programming, by doing experiments and applying questionnaires. In this work, an analysis of tools that could be used to do experiments in programming classes was done. The choice of the tool called CodeCombat was based in the following reasons: ease of use, presence of gamified elements, use of the Python programming language and availability in Portuguese. Among the gamified elements, CodeCombat has: avatar creation, mission map, levels and achievements. After this, the planning about how the tool would be integrated to the Programming Fundamentals course taught to graduation courses in Federal University of Ceará was done. Then the experiment was designed, being applied in the first semester of 2023, using a Computer Science class as experimental group and a Software Engineering class as control group. To collect the results, the Intrinsic Motivation Inventory (IMI) was used, with the goal to evaluate the students' motivation for learning in the class after the use of the tool. With the obtained results, it was concluded that, taking in consideration the free version of the tool's limitations, its use did not bring a big impact to the motivation of the students for the class, but the potential for using it in other ways still exists.

Keywords: gamification; teaching; programming; computer; CodeCombat.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A programação de computadores se tornou uma ferramenta muito importante para o desenvolvimento da humanidade, e a demanda por programadores cresce cada vez mais. Para suprir essa demanda, foram criados vários cursos de níveis técnico e superior de computação, mas existe um problema neles: a alta evasão de estudantes. Muitos estudos foram feitos, como o de Hoed et al. (2017), que afirma que os possíveis motivos são divididos em:

1. Questões econômicas: Impossibilidade de permanecer no curso por questões econômicas;
2. Questões vocacionais: Falta de identificação com o curso;
3. Questões institucionais: Abandono por fracasso nas disciplinas iniciais, deficiências prévias de conteúdos anteriores, inadequação aos métodos de estudo, dificuldades de relacionamento com colegas ou membros da instituição.

Foi descoberto que alunos que desistem de disciplinas iniciais do curso mencionaram falta de tempo e motivação como os principais motivos de desistência (HOED et al., 2017 apud KINNUNEN; MALMI, 2006). Uma possível saída proposta nesse trabalho seria o uso da gamificação.

A gamificação é uma forma de tornar o processo de ensino e aprendizagem de programação mais dinâmico e divertido. Aplicando conceitos de Game Design, este trabalho traz um estudo sobre a possibilidade de aumentar a motivação dos estudantes para o aprendizado podendo auxiliar o aluno nas questões vocacionais e institucionais e reduzir a evasão na disciplina e no curso. A ferramenta escolhida para ser testada após uma análise técnica foi a CodeCombat.

1.1 Questões de pesquisa

Este trabalho se propõe a responder as seguintes questões de pesquisa:

QP1: O uso do CodeCombat irá melhorar a motivação dos alunos de programação?

QP2: A ferramenta CodeCombat irá se adequar para o objetivo proposto?

1.2 Objetivos

1.2.1 Objetivo geral

Verificar se o CodeCombat aumenta a motivação dos alunos.

1.2.2 Objetivos específicos

1. Definir um grupo experimental e um grupo de controle;
2. Aplicar um experimento no grupo experimental;
3. Coletar informações sobre as pessoas que participaram nos dois grupos;
4. Avaliar e comparar as informações coletadas para responder às questões de pesquisa.

2 REVISÃO BIBLIOGRÁFICA

Para a revisão bibliográfica, foi realizada uma busca de artigos e trabalhos relevantes para o tema proposto e, após selecionados os mais relevantes para o trabalho, foi realizado um relatório técnico de cada. Os trabalhos selecionados foram os seguintes: O que Leva os Alunos dos Cursos Superiores de Computação a Evadirem? Um Estudo de Caso Feito na Universidade de Brasília (HOED et al., 2017), Why Students Drop Computing Science: Using Models of Motivation to Understand (BARR; KALLIA, 2022), Ensino de Programação para Futuros Não-Programadores: Contextualizando os Exercícios com as Demais Disciplinas de mesmo Período Letivo (CARVALHO et al., 2016), Effect of Gamification on the Motivation of Computer Programming Students (CUERVO-CELY et al., 2022), Uma experiência de gamificação no contexto do ensino remoto: análise da motivação e experiência dos jogadores (MOREIRA, 2021), Descrição e Comparação de Jogos Digitais para Auxiliar no Ensino de Programação (GOMES et al., 2018). Os trabalhos serão descritos em detalhes a seguir.

2.1 O que Leva os Alunos dos Cursos Superiores de Computação a Evadirem? Um Estudo de Caso Feito na Universidade de Brasília

O trabalho inicia frisando a importância da análise da evasão, pois é obrigação institucional zelar pelo bem estar e aprendizado dos discentes, além de a evasão gerar prejuízos financeiros.

Existem números no Brasil e no mundo indicando altas taxas de evasão. Na Irlanda, analisando-se os anos letivos de 2012/2013 e 2013/2014, verificou-se que as áreas de Construção e afins, Serviços, Ciência da Computação e Engenharia apresentaram a maior taxa de evasão (HOED et al., 2017 apud LISTON M.; PATTERSON V., 2017). Também observou-se que, nos cursos da área de Ciência da Computação da Universidade de Tecnologia de Helsinki, na Finlândia, as taxas de evasão tem variado entre 30% e 50% (HOED et al., 2017 apud KINNUNEN; MALMI, 2006).

Entre 2005 e 2009, houve uma redução de aproximadamente 20% do número de estudantes de Ciência da Computação na França (HOED et al., 2017 apud MURATET et al., 2009).

O objetivo geral do trabalho foi analisar as causas de evasão nos cursos superiores de Computação a partir de um estudo de caso feito na Universidade de Brasília (UnB). O

estudo considerou evadido o aluno que é desligado do curso antes de sua conclusão, seja espontaneamente (exceto por falecimento) ou devido a normas institucionais.

O primeiro conceito utilizado para fundamentar teoricamente o trabalho foi o de evasão. Os conceitos de evasão são: evasão de curso (desligamento apenas do curso, sem desvínculo com a instituição) evasão da instituição (desligamento da instituição na qual está matriculado) e evasão do sistema (abandono do ensino superior) (HOED et al., 2017 apud (MEC)., 1997).

Em relação ao currículo dos cursos superiores de Computação, há diversos autores que relatam dificuldades dos alunos em disciplinas relacionadas a Algoritmos e Matemática.

As condições que motivam a evasão escolar podem ser classificadas em três classes distintas:

1. Questão econômica: Impossibilidade de permanecer no curso por questões econômicas.
2. Questão vocacional: Falta de identificação com o curso.
3. Questão institucional: Abandono por fracasso nas disciplinas iniciais, deficiências prévias de conteúdos anteriores, inadequação aos métodos de estudo, dificuldades de relacionamento com colegas ou membros da instituição.

Além dessas causas de desistência, motivos adicionais foram apresentados por outros autores e usados na pesquisa.

Para a parte de metodologia, foi feito um estudo de caso na UnB. Os autores buscaram os Registros Acadêmicos no período de 2005-2015 nos cursos de Bacharelado em Ciência da Computação, Licenciatura de Computação e Engenharia da Computação ofertados pela universidade. Na pesquisa, foram incluídos diversos itens para que os ex-alunos respondessem quanto à influência de cada um deles sobre a decisão de abandono do curso. Foram disponibilizados 21 itens em um formulário de escala Likert, de 1 a 5, sobre como cada item influenciou sua saída no curso, mostrado na Figura 1.

No formulário de escala Likert da pesquisa, foram utilizados itens agrupados em questões socioeconômicas, vocacionais e institucionais. Outras classificações também foram utilizadas para itens que não se ajustaram perfeitamente nessas classificações, como problemas de saúde ou familiares.

Após a aplicação do questionário, notou-se que a maioria dos respondentes de cada curso foi do sexo masculino.

Os itens vocacionais, institucionais e socioeconômicos foram os mais citados pelos

Figura 1 – Perguntas utilizadas no questionário

Classificação	Itens
Questões socioeconômicas	Impossibilidade de conciliar o serviço com o estudo
	Dificuldades financeiras
	Não recebimento de auxílio estudantil
Questões vocacionais	Mudança para um curso de área diferente de Computação
	Falta de afinidade com o curso
	Dedicação a outros estudos
	Falta de identificação com profissionais dessa área
Questões institucionais	Critérios de avaliação dos discentes usados são inadequados ou muito rígidos.
	Dificuldades em disciplinas do curso ligadas à matemática
	Dificuldades em disciplinas do curso ligadas ao ensino de programação e algoritmos
	Dificuldades em outras disciplinas do curso
	Falta de informações sobre o que é abordado no curso.
	Infraestrutura física da instituição inadequada
	Dificuldades de relacionamento com os professores/funcionários
	Dificuldade dos professores em repassar de maneira compreensível os conteúdos
Questões familiares/pessoais/ de saúde	Distância da família
	Problemas afetivos
	Dificuldade de relacionamento com os colegas
	Problemas de saúde (seja seu ou de algum familiar)
Outras questões	Transferência para outra instituição (continuou cursando o mesmo curso em outra instituição)
	Mudança de município

Fonte: (HOED et al., 2017)

respondentes. Questões familiares/pessoais/de saúde e outras questões não apareceram entre as mais citadas. Algumas questões apareceram em todos os cursos como grandes influenciadores da evasão, sendo: dificuldade dos professores em repassar de maneira compreensível o conteúdo (item institucional); falta de informações sobre o que é abordado (item institucional); critérios de avaliação utilizados são inadequados ou muito rígidos (item institucional; e falta de afinidade com o curso (item vocacional).

Os alunos também puderam acrescentar motivos relevantes para sua saída que não tenham sido mencionados na pesquisa. Verificou-se que os itens mais acrescentados são institucionais. Alguns itens institucionais se destacaram por terem sido citados nos 3 cursos como currículo desatualizado/ementa inadequada, falta de adequação da academia com o mercado de trabalho e reprovações sucessivas.

Ao responderem o que levou os alunos à escolher computação, mais de 50% dos alunos em cada curso afirmaram ter sido motivados principalmente pela afinidade, curiosidade e interesse por computadores e pela área da Computação. É interessante também ressaltar que a falta de afinidade foi uma das questões mais pontuadas no formulário da pesquisa como causa da evasão, indicando que os alunos não possuem uma visão adequada do que realmente é estudado na computação. Ao serem consultados sobre a disciplina que julgam como a mais complexa, as mais citadas foram disciplinas de Lógica de Programação e Cálculo, sendo a dificuldade na disciplina de Lógica um dos fatores que mais causa evasão, segundo a pesquisa feita.

O trabalho conclui-se mencionando que, com as causas de evasão dos cursos de Computação, verifica-se que se trata de um problema com múltiplos fatores. Algumas causas se destacam, principalmente fatores institucionais e acadêmicos. O autor também observa que as causas de evasão citadas por outros autores também podem ser classificadas em questões vocacionais, institucionais, socioeconômicas, pessoais e outras questões. Com isso, pode-se verificar que há questões recorrentes e similares levantadas por outros fatores. Os gestores educacionais podem se beneficiar com o conhecimento das causas de evasão. Algumas medidas podem contribuir para diminuir a evasão, como a formação de grupo de estudos, estímulo aos trabalhos de monitoria e ministração de aulas de reforço para os alunos com dificuldade no início do curso. O autor pontua também que trabalhar para a permanência do aluno é um compromisso que toda instituição deve assumir com a formação do aluno.

2.2 Why Students Drop Computing Science: Using Models of Motivation to Understand Student Attrition and Retention

O trabalho começa discorrendo sobre as preocupações crescentes relacionadas a participação nos cursos de Ciência da Computação apesar das iniciativas para melhorar a diversidade. No Reino Unido, estudos demonstram que as mulheres representam somente 19,9% dos alunos que estudam computação no nível universitário. (BARR; KALLIA, 2022 apud KINNUNEN; MALMI, 2006), por exemplo, descobriram que alunos que desistiram de uma disciplina inicial do curso mencionaram falta de tempo e de motivação como os principais motivos da desistência. Entretanto, esses motivos são sustentados por outros fatores, como a dificuldade percebida do assunto, desafios cercando gerência de tempo e a preferência por outro assunto. Uma década depois, foi encontrado um conjunto igualmente complexo para os alunos desistirem da disciplina inicial (CS1) de computação, incluindo falta de tempo, habilidades de estudo mal desenvolvidas e prioridade de estudo em outros assuntos(BARR; KALLIA, 2022 apud PETERSEN et al., 2016).

Estudos também demonstraram porque as mulheres, em particular, desistem da disciplina. Fatores culturais, como a cultura dos "geeks machões", referindo a cultura de homens "sabichões", que infelizmente ainda domina as salas de aula, foram identificados. É perceptível que a exclusão das mulheres na computação é um problema que se auto perpetua, reforçado pelos estereótipos de gênero na computação, e pelas experiências das mulheres em ambientes de estudo e trabalho dominados por homens. Fatores positivos envolvem suporte da comunidade e

exposição para a computação.

Outro modelo, sugere que dois dos mais importantes fatores preditivos para a motivação são crenças específicas para as tarefas e o valor que os alunos atribuem para o domínio correspondente (BARR; KALLIA, 2022 apud ECCLES, 1983).

Para responder essas questões, foram entrevistados 32 alunos da Universidade de Glasglow.

Para a parte de fundamentação teórica, foram explicados os seguintes conceitos:

Modelo de valor de expectativa de escolhas relacionadas a conquistas: Em 1983, foi sugerido este modelo (BARR; KALLIA, 2022 apud ECCLES, 1983). A premissa básica deste modelo é que a escolha de um indivíduo, a persistência e a performance em um domínio, e sua motivação para realizar uma tarefa podem ser explicadas por dois fatores: sua expectativa de sucesso e o quanto ele valoriza determinada tarefa.

Quadro de referência generalizado interno/externo: Auto-conceito é descrito como uma visão composta de si mesmo, e, no contexto da psicologia educacional, é visto como crenças individuais relacionadas aos pontos fortes e fracos da pessoa. O quadro externo é a comparação que os alunos fazem entre as auto-percepções de suas habilidades em uma disciplina e a habilidade percebida de outros alunos no mesmo contexto. Já no quadro interno, os alunos comparam suas habilidades percebidas em um domínio específico com suas habilidades percebidas em outro domínio (BARR; KALLIA, 2022 apud MARSH, 1986).

O trabalho propõe duas questões de pesquisa:

QP1: Quais fatores influenciam as escolhas dos alunos da graduação em relação à Ciência da computação?

QP2: Até que ponto o modelo de expectativa e conquista de e o quadro de referência podem ser usados para explicar esses fatores?

Para esse estudo, alunos que estavam matriculados no terceiro e quarto ano da graduação foram consultados, para saber quais tinham estudado Ciência da Computação no primeiro ou segundo ano, mas não estavam mais matriculados em nenhuma disciplina desse curso. Essa consulta coletou 104 resultados. Uma vez que a coleta de dados de alunos é sensível, foi feita uma consulta de interesse para estabelecer a base para processamento adicional dos registros do aluno. A base para o uso desses dados estava clara, já que a diversidade de gênero é um problema persistente no setor de computação da universidade, e a defesa da diversidade e

inclusão está contida na estratégia da universidade.

Alunos que desistiram de Computação receberam um convite por e-mail, para responder a uma pesquisa de opinião em troca de um voucher da Amazon de €10. Não houve influência da pesquisa no desempenho dos alunos, pois ela foi baseada em um curso que os alunos já desistiram, e a pesquisa foi aprovada pelo Conselho de Ética da universidade. A pesquisa recebeu 33 respostas. Os alunos tiveram a opção de informar seu gênero, e todos optaram por informar. Das respostas recebidas, 33% se identificaram como mulheres, 64% como homens e 3% como não-binários. 70% informaram que tinham entre 16 e 18 anos quando entraram na graduação, 18% tinham entre 19 e 21, e 12% tinham 22 anos ou mais de idade. Além dessas perguntas, os inquiridos foram solicitados a responder porque eles não continuaram no curso de Ciência da Computação, e como seu gênero afetou sua experiência no curso, caso tenha afetado. As respostas para cada uma dessas três últimas perguntas foram analisadas por ambos os autores usando uma abordagem de dedução temática, baseado no modelo de valor de expectativa de escolhas relacionadas a conquistas.

Para os resultados, oito temas principais emergiram dos dados, que serão explicados a seguir:

Utilidade: O tema de utilidade se refere ao grau de utilidade do curso para os objetivos do aluno. 42% dos alunos falaram sobre sua decisão de não continuar na computação por ela não estar alinhada com seus objetivos futuros.

Custo: O tema de custo reflete como a decisão dos inquiridos em continuar no curso limitava o acesso a outras atividades ou demandava muito tempo e esforço emocional. No total, 28% dos alunos relataram problemas relacionados ao custo de se engajarem na Ciência da Computação. A análise dos autores revelaram dois sub-temas importantes: tempo, se referindo ao tempo e esforço ao se engajarem na computação; e afeto, se referindo ao impacto emocional em se engajar com a computação.

Valor intrínseco: O tema de valor intrínseco agrupa problemas relacionados ao interesse dos alunos em estudar computação por motivos relacionados exclusivamente a computação. No total, 23% das respostas dos alunos se encaixaram nessa categoria.

Expectativa de sucesso: Esse tema se refere à problemas com crenças dos alunos sobre o quão bem eles se sairiam na Ciência da Computação. 13% dos alunos relataram que um motivo para não continuar na computação foi a crença de que eles não iriam se sair bem no curso.

Realização Esse tema inclui as referências dos alunos para sua auto-imagem, valores pessoais e

crenças, ou seja, sua identidade. Basicamente, é relacionado com a identificação do aluno com o curso. 13% dos alunos mencionaram problemas alinhados a esse tema.

Comparações: Refere-se à decisão dos alunos em desistir como resultado de comparações feitas entre a Computação com outro assunto. No total, 25% dos alunos compararam Computação com outras áreas para justificar sua desistência. Dois sub-temas principais entram nesse tema: o quadro externo e interno. Em relação a diferença de gênero, houve um maior número de relatos de problemas de alunos do gênero feminino (36%) do que do masculino (19%) para comparações internas. Já nas comparações externas, houve somente relatos de problemas de alunas do gênero feminino.

Social: Refere-se a fatores relacionados a ter um ambiente acolhedor e uma rede social que os alunos podem utilizar para pedir ajuda ou fazer com que se sintam incluídos no curso. No total, 23% das respostas se encaixaram na categoria. Em comparação a outros temas, fatores sociais são mais importantes para alunas do gênero feminino (55% das alunas relataram problemas relacionados a esse fator) do que do masculino (5% dos alunos relataram problemas). O único aluno não binário também citou fatores sociais.

Dificuldade: Reflete a dificuldade percebida da Computação como um assunto. Difere-se do assunto de expectativa de sucesso, pois a percepção de um assunto como difícil não necessariamente vai impactar na expectativa de sucesso, pois ela pode ser influenciada por outros fatores. No total, expectativa de sucesso pode ser influenciada por fatores fora a dificuldade do assunto. No total, 42% dos alunos relataram problemas relacionadas ao tema.

O trabalho conclui informando que o objetivo do estudo foi investigar porque os alunos desistem da Ciência da computação aplicando teorias de motivação que explicam as escolhas dos alunos. Os autores descobriram que duas teorias motivacionais explicam a decisão dos alunos e se complementam. Futuramente, a pesquisa dos autores tem o objetivo de delinear os relacionamentos entre todos esses fatores examinando-os com respeito ao gênero e aos grupos de minorias. Admitindo as limitações de pesquisas de opinião, os autores propuseram que o trabalho futuro faça uso de uma entrevista semi-estruturada profunda, projetada com referência ao modelo de expectativa e valor e os resultados preliminares apresentados no trabalho.

2.3 Ensino de Programação para Futuros Não-Programadores: Contextualizando os Exercícios com as Demais Disciplinas de mesmo Período Letivo

O trabalho começa falando sobre as dificuldades dos alunos dos cursos de Engenharia e Ciências Exatas (exceto Computação) em se motivarem para estudar programação de computadores, pois eles não conseguem entender a importância nas suas atividades profissionais, resultando em altos índices de reprovação.

Para contornar essa dificuldade, uma equipe de professores da Universidade Federal do Amazonas concebeu uma metodologia de ensino-aprendizagem dinâmica, que não se limita a exposição de conteúdo, mas que trabalha com diferentes formas de exercícios. Esses exercícios são elaborados de uma forma contextualizada, levando em consideração o domínio de problemas relacionados ao curso e as disciplinas que os alunos estão cursando.

Após a introdução, o trabalho é contextualizado, falando sobre a disciplina alvo do trabalho, que é chamada de Introdução a Programação de Computadores(IPC), ministrada para cerca de 500 alunos do primeiro período de onze cursos de Engenharia e Ciências Exatas na Universidade Federal do Amazonas. Um dos problemas mencionados é que poucos professores se sentem encorajados a ministrar essa disciplina, que é normalmente atribuída a professores substitutos da universidade.

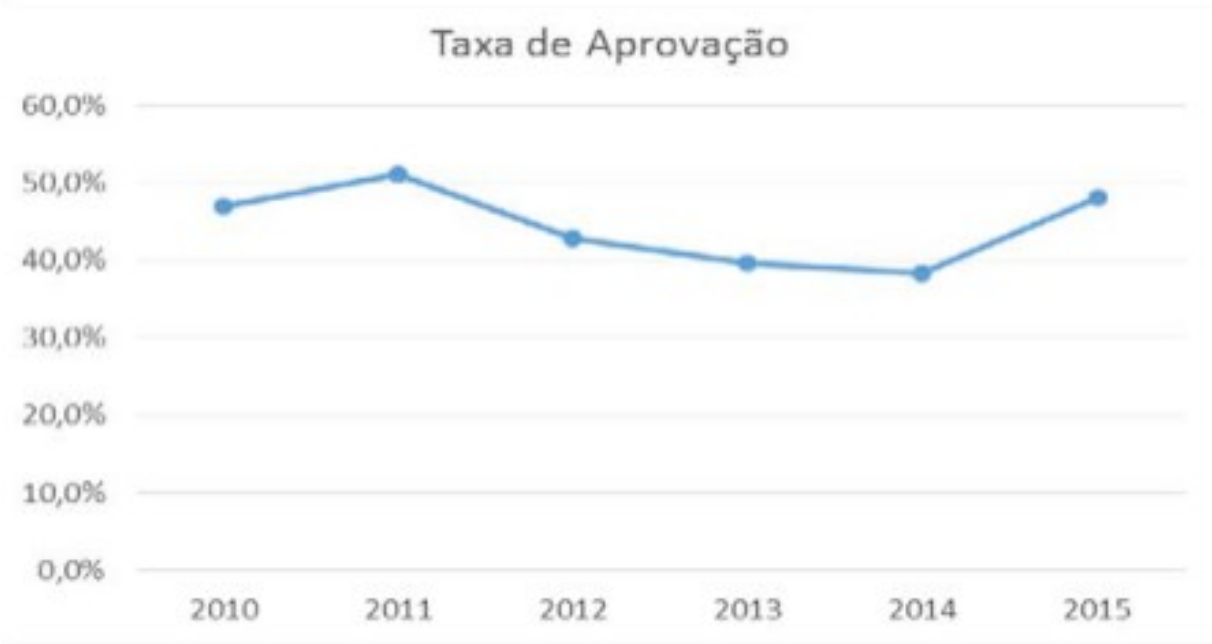
O gráfico da Figura 2 mostra a evolução da taxa de aprovação da disciplina de IPC. Os números decrescentes de 2011 à 2014, bem como o aumento do número de matrículas, motivaram os professores a reformular a metodologia de ensino, com o propósito de reduzir a taxa de reprovação por motivos de nota. Existem dois pontos destoantes no gráfico, em 2010 e 2015. Em 2010, um dos possíveis motivos é a adoção do SISU, onde muitos alunos selecionaram cursos que não possuíam vocação e só descobriram posteriormente, levando a abandonos. Em 2015, uma possível explicação é a adoção da metodologia apresentada no trabalho.

Uma explicação para parte da reprovação é o descontentamento do aluno com o curso, que não é influenciada somente pela disciplina de IPC.

Para tratar o problema da desmotivação relacionada a disciplina de IPC, os autores foram buscar na literatura relatos e técnicas que fornecessem embasamento para reformular a metodologia de ensino de uma forma eficiente e eficaz.

Baseado na revisão da literatura, foi decidido reformular a metodologia de ensino da disciplina. A metodologia aplicada no trabalho se caracteriza na remodelação da disciplina, que é composta por 60 horas de aulas presenciais, sendo 30 horas de carga horária teórica e 30

Figura 2 – Taxa de aprovação dos anos 2010 2015



Fonte: (CARVALHO et al., 2016)

de prática, para sete módulos de oito horas (quatro encontros de duas horas), mais um módulo inicial de quatro horas (dois encontros) de motivação e apresentação da disciplina. A distribuição do conteúdo se foi da seguinte forma:

1. Variáveis.
2. Estruturas condicionais.
3. Estruturas condicionais aninhadas.
4. Estrutura de repetição por condição
5. Vetores.
6. Estrutura de repetição por contagem.
7. Matrizes.

O oitavo módulo sobre “strings” teve o conteúdo ministrado mas não foi avaliado. O tópico de “funções” foi ministrado de forma diluída entre os módulos de 5 a 7. Cada módulo tinha quatro aulas de duas horas de duração, na seguinte sequência: aula teórica, laboratório de codificação, laboratório de exercícios e avaliação parcial. Os exemplos e exercícios eram cumulativos, ou seja, módulos mais avançados cobravam módulos mais iniciais, por conta disso, avaliações dos módulos mais avançados tinham maior peso. Nos laboratórios, os exercícios eram corrigidos pela ferramenta CodeBench.

Os resultados apresentados consistiram na taxa de aprovação das turmas em que a metodologia foi aplicada e comparada com a das turmas em que não foi, o índice de facilidade

e o índice de discriminação das questões elaboradas para as avaliações. Nas turmas onde a metodologia foi aplicada, a taxa de aprovação foi de 60%, enquanto nas que não foi, a taxa foi de 40%. Foi verificado também se houve enviesamento entre turmas que vieram mais preparadas do Ensino Médio e aplicação da metodologia. Os resultados sugeriram que a taxa de aprovação em IPC independe da metodologia aplicada. Também foram analisados os níveis de facilidade dos alunos para a realização das questões e, no final, também foi aplicada uma pesquisa para coletar os feedbacks dos alunos por meio de questionários, que no geral foram positivas.

Os autores concluíram o trabalho mencionando que os resultados preliminares indicaram um índice de aprovação maior que em períodos anteriores. Além disso, questões interdisciplinares se mostraram efetivas tanto pelo seu poder discriminatório quanto pelas avaliações dos alunos. Como sugestões para trabalhos futuros, o trabalho trouxe o desejo de fazer uso de Inventários de Conceitos para aferir o ganho de aprendizagem antes e depois da disciplina, além do detalhamento maior de dados.

2.4 Effect of Gamification on the Motivation of Computer Programming Students

O artigo tem o objetivo de examinar o efeito da gamificação assistida por computador na motivação do aprendizado de estudantes de programação de computadores. Atualmente, a área de programação de computadores é fundamental para o desenvolvimento tecnológico da Ciência da Computação e da indústria de Tecnologia da Informação (TI). Essa situação faz a demanda por profissionais da área aumentar, criando a necessidade de cursos de programação no meio acadêmico, e uma das principais dificuldades destes cursos é a dificuldade que os alunos enfrentam no aprendizado dos conceitos, resultando em desmotivação e culminando na evasão dos mesmos.

Para o desenvolvimento desta pesquisa, vários ambientes gamificados foram identificados com o objetivo de realizar uma análise comparativa e selecionar a ferramenta ideal para ser incorporada em um curso introdutório de programação. Uma revisão da literatura foi realizada e permitiu a identificação de diferentes ferramentas gamificadas para o ensino-aprendizagem de programação. Posteriormente, uma descrição de cada uma delas foi feita com suas características e objetivos de acordo com o propósito de cada ambiente de aprendizagem gamificado. As ferramentas analisadas pelos autores foram as seguintes: SoloLearn, CheckiO, Codingame, ViLLE, Stack Overflow, EasyLogic, Funprog, CodeCombat, CodeGym, CodeHS, Codecademy, CodeAvengers.

Finalmente, foi realizada uma análise comparativa para selecionar a ferramenta que se adequava às necessidades e propósitos para o desenvolvimento da pesquisa. A ferramenta selecionada foi a CodeGym, seguindo o critério de presença de maior número de elementos gamificados, além do filtro de que somente ferramentas que usavam as linguagens de programação Java ou Python seriam selecionadas. Também foi levado em consideração o fato que o CodeGym realiza uma introdução teórica do nível básico ao avançado dos conceitos de programação.

Sobre a ferramenta CodeGym, ela é uma ferramenta gamificada da empresa HiTech Rush Inc., de 2015, que possibilita o ensino de programação da linguagem Java. Os participantes podem se cadastrar na plataforma e aprender os tópicos básicos até os avançados. O conteúdo da plataforma é organizado em missões, onde cada missão é composta por diferentes níveis que o participante completa a medida que o usuário realiza os exercícios. A ferramenta apresenta o conteúdo por meio de uma narrativa adaptada para um contexto espacial, envolvendo naves espaciais, astronautas e outros planetas.

O projeto da pesquisa corresponde em um design sequencial explanatório misto, considerando a variável independente a intervenção educacional gerada pelo ambiente de aprendizado gamificado assistido por computador, e como a variável dependente, a motivação no aprendizado do ensino de programação. O experimento teve um grupo experimental e dois de controle. O grupo experimental utilizou a ferramenta e os de controle, não. Antes e depois da intervenção, foi aplicado um questionário chamado MSLQ-Colombia para medir a motivação dos alunos antes e após a aplicação do experimento. Foram selecionados 48 alunos para participar do experimento. O grupo experimental consistiu de 17 alunos do curso de Engenharia de Sistemas. Os grupos de controle consistiram de um grupo de 16 alunos de Engenharia Industrial, e outro com 15 alunos de Engenharia Mecânica.

Os níveis do CodeGym escolhidos e integrados ao currículo das aulas do grupo experimental foram:

Nível 0: Introdução

Nível 1: Introdução ao Java - saída, int e strings.

Nível 2: Introdução ao Java, variáveis, métodos.

Nível 3: Primeiro programa: Entrada no teclado, utilização de IDE.

Nível 4: Introdução a escolhas e repetições.

Esses níveis foram selecionados por serem compatíveis com o currículo do curso que o grupo experimental estava realizando.

O experimento seguiu 5 fases e 9 estágios. As fases foram as seguintes:

Fase 1: Diagnóstico Inicial. (Estágio 1)

Fase 2: Fase Formativa (Estágios 2, 3 e 5) Fase

3: Fase de Feedback (Estágios 4 e 6) Fase 4:

Diagnóstico Final (Estágios 7 e 8) Fase 5: Fase

5: Análise de Dados. (Estágio 9)

E os estágios foram os seguintes:

Estágio 1: Aplicação do questionário MSLQ-Colombia para o grupo experimental e os grupos de controle.

Estágio 2: Introdução teórica ao conteúdo. Repetido semanalmente.

Estágio 3: Treinamento do conteúdo utilizando a CodeGym. Repetido semanalmente.

Estágio 4: Tira dúvidas sobre os módulos do CodeGym.

Estágio 5: Atividades práticas aplicadas para todos os grupos, que utilizaram ou não a ferramenta.

Estágio 6: Aplicação de enquetes e esclarecimentos dos tópicos.

Estágio 7: Aplicação do MSLQ-Colombia uma segunda vez.

Estágio 8: Pesquisa de opinião, realizada pelo grupo experimental, sobre as mudanças na motivação provocadas pelo CodeGym.

Estágio 9: Análise quantitativa e qualitativa sobre os efeitos do uso da ferramenta na motivação.

Para analisar quantitativamente os dados coletados com o MSLQ-Colombia, um teste estatístico foi selecionado para mostrar se houve uma mudança significativa na motivação. As hipóteses consistem em:

H0: Gamificação assistida por computador tem nenhum efeito na motivação dos alunos para a aprendizagem de programação.

H1: A gamificação assistida por computação gera um efeito na motivação dos alunos para a aprendizagem de programação.

Para analisar qualitativamente, foram utilizadas pesquisas de opinião com respostas abertas e utilizando a escala Likert, que consiste em uma especificação do perguntado do quanto ele concorda com determinada afirmação.

No geral, os resultados obtidos no pré-teste mostraram que nos 6 primeiros aspectos motivacionais os alunos geralmente se identificaram com os valores mais altos da escala Likert. Na categoria relacionada à ansiedade nos processos de avaliação, as respostas se mostraram

divididas entre identificação com altos níveis de ansiedade ou preocupação relacionados à provas e avaliações, e ao não sentimento de descrição ou leve identificação com sentimentos de ansiedade nos processos avaliativos. No pós-teste, houveram melhoras nos aspectos de utilidade, importância e interesse.

Nas pesquisas de opinião, os alunos indicaram que sentiram uma melhora nos aspectos motivacionais relacionados a importância, utilidade, identificação ou interesse no conteúdo. Além disso, indicaram que, graças aos exercícios práticos do CodeGym, eles sentiram desejos autênticos de aprender, maior autoconfiança para abordar as tarefas e maiores expectativas para atingir seus objetivos de aprendizagem. Os alunos também mencionaram que os exercícios eram desafiadores, acrescentando ao conteúdo trabalhado em sala de aula. Em relação à ansiedade para as avaliações, as respostas foram mistas, com alguns alunos respondendo que o CodeGym permitiu que eles se sentisse mais seguros para as avaliações, e outros que responderam que o CodeGym não ajudou nesse quesito.

Para a conclusão, a valorização de tarefas foi o aspecto motivacional que mostrou uma maior evolução após o uso da ferramenta gamificada. Também foi mostrado nas análises qualitativas e quantitativas que, devido aos exercícios, os participantes sentiram vontade real de aprender, maior confiança para a realização dos exercícios e melhores expectativas para atingir as metas de aprendizado. Um possível problema é o espaço amostral pequeno mas, no geral, a pesquisa mostrou resultados promissores para o uso de ferramentas gamificadas no ensino de programação.

2.5 Uma experiência de gamificação no contexto do ensino remoto: análise da motivação e experiência dos jogadores.

Este trabalho relacionado foi realizado durante o isolamento social para combater a pandemia do Covid-19. Com isso, foi necessária a adaptação do formato de ensino presencial das universidades para o ensino remoto. Neste contexto, as ferramentas de e-learning desempenharam um papel muito importante e, para obter um bom desempenho, foi necessária a aplicação de diferentes estratégias pedagógicas devido a mudanças de hábitos dos alunos e professores, tornando o desempenho dos estudantes nesse novo contexto uma preocupação. Uma técnica que pode ser utilizada para apoiar o processo de ensino e aumentar a motivação dos estudantes é a gamificação, que se trata da utilização de técnicas de Game Design em contextos do mundo real. Para guiar esta pesquisa, três questões de pesquisa foram elaboradas:

QP1: Qual o desempenho dos estudantes na experiência gamificada com a plataforma e-learning?

QP2: Qual a motivação percebida pelos estudantes em relação aos elementos de gamificação implementados com a ferramenta e-learning conduzida de forma remota?

QP3: Qual a experiência de jogo percebida pelos estudantes com a utilização da ferramenta e-learning conduzida de forma remota?

Para responder a essas questões, foi conduzido um estudo de caso na disciplina de Qualidade de Software na UFC Campus de Russas, onde os dados a serem analisados foram coletados a partir de pontuações no sistema e-learning e por meio de questionários que traziam questões relacionadas à motivação e à experiência de jogo.

Para os procedimentos metodológicos, as etapas aplicadas foram a Revisão da literatura, Seleção da plataforma de gamificação, Implementação da gamificação na disciplina, Estudo de caso e Análise dos resultados.

Para a etapa de Revisão da literatura, o objetivo foi realizar um estudo de temas associados à pesquisa, além de um estudo de trabalhos que propuseram plataformas para gamificação.

Para a etapa de seleção de plataforma gamificada a principal fonte para pesquisa e exploração de sistemas gamificados foi conduzida por meio de pesquisas acadêmicas que utilizaram sistemas durante a abordagem de gamificação.

Para a etapa de implementação da gamificação na disciplina de Qualidade de Software, o principal objetivo foi a definição e construção da experiência gamificada, usando a ferramenta ClassCraft.

Para o estudo de caso, foram definidas as seguintes etapas:

Definição do contexto e participantes: foi realizado o detalhamento das principais variáveis do contexto.

Procedimento de coleta de dados utilizados: foi definida a utilização da taxonomia proposta por Bartle (MOREIRA, 2021 apud BARTLE, 1996) para a caracterização dos estudantes no ambiente gamificado, e a utilização dos questionários Intrinsic Motivation Inventory (IMI) e Game Experience Questionnaire (GEQ) para a investigação da motivação e experiência dos estudantes.

Preparação das atividades: Foram realizadas atividades como disponibilização de vídeos de orientações, disponibilização de material de apoio para os alunos e preparação do ambiente

gamificado.

A última etapa tratou da análise dos resultados colhidos com base nas pontuações obtidas nas atividades e dos resultados dos questionários aplicados no fim da realização do experimento.

Analisando a disponibilidade e funcionalidades das ferramentas identificadas, o módulo gratuito do ClassCraft oferece vantagens como flexibilidade, personalização, disponibilidade em web e mobile.

O estudo de caso foi realizado na disciplina de Qualidade de Software do campus de Russas, aplicando o questionário GEQ (Game Experience Questionnaire) para avaliar a experiência dos alunos na plataforma gamificada e o IMI (Intrinsic Motivation Inventory) para avaliar a motivação dos estudantes. Para entender os perfis dos jogadores, também foi utilizada a taxonomia proposta por Bartle(MOREIRA, 2021 apud BARTLE, 1996) para a classificação dos perfis dos jogadores. Essa categorização foi dividida em:

Conquistador/Realizador: Perfil de jogador mais relacionado a agir sobre o jogo, tendo interesse em prazer e desafios.

Explorador: É o perfil que mais respeita a interação durante o jogo, com o propósito de conhecê-lo melhor.

Socializador: Perfil que tem o objetivo de interagir mais com os outros jogadores.

Predador: Perfil que tem o objetivo de agir mais sobre os outros jogadores, com o objetivo de competição e vitória.

Para a preparação das atividades da disciplina, foram feitas as etapas de disponibilização de vídeo para orientar os estudantes sobre o uso da ferramenta, elaboração do material de apoio, preparação do ambiente, cadastro dos alunos na plataforma ClassCraft e Google Classroom, personalização envolvendo a criação de avatares por cada aluno e criação de equipes dentro do ClassCraft.

Finalmente, a execução do estudo envolveu a proposição das atividades no decorrer da disciplina. Os resultados obtidos pelo experimento foram os seguintes:

Primeiramente, foi feito um levantamento geral sobre o perfil de cada jogador, e depois, foi feito um levantamento relacionado a pontuação obtida nas missões pelos alunos em cada perfil.

Para a avaliação de experiência de jogo, foi utilizado o questionário GEQ, sendo dividido entre as categorias Tensão, Imersão Sensorial e Imaginativa, Fluxo, Efeito Positivo,

Efeito Negativo, Desafio e, por último, Competência.

Para a avaliação da motivação dos jogadores, foi utilizado o questionário IMI, sendo dividido nas dimensões: Competência Percebida, Escolha Percebida, Interesse/Prazer, Pressão/Tensão.

Finalmente, as respostas às questões de pesquisa, foram as seguintes. Para a QP1 (Qual o desempenho dos estudantes na experiência gamificada com a ferramenta e-learning), foi apresentada de acordo com o perfil de cada estudante, sendo observado que os estudantes com perfis exploradores e socializadores tiveram um maior desempenho durante a adoção da gamificação na disciplina, e os com perfil Conquistador obtiveram um menor desempenho, possivelmente devido à falta de experiência na aplicação e falta de incentivo entre a competitividade dos jogadores. Para a QP2 (Qual a motivação percebida em relação aos elementos de gamificação implementados com a ferramenta e-learning conduzida de forma remota?), foi percebido um aumento na motivação dos estudantes utilizando a plataforma gamificada. E para a QP3 (Qual a experiência de jogo percebida pelos estudantes com a utilização da ferramenta e-learning conduzida de forma remota?), foi observado que os estudantes gostaram da utilização dos elementos utilizados na gamificação e na utilização da plataforma em si.

2.6 Descrição e Comparação de Jogos Digitais para Auxiliar no Ensino de Programação

Este trabalho inicia discutindo a dificuldade de conduzir disciplinas que adequam teoria com prática, como disciplinas de programação, por misturarem fatores técnicos com humanos. E essas disciplinas são muito importantes devido à demanda por profissionais de tecnologia. O trabalho também menciona que as disciplinas de programação são muito difíceis, sendo essa dificuldade associada a altos índices de reprovação e evasão. Além disso, ele menciona o quão complexas são as linguagens de programação e cita a utilização de pseudolinguagens para lidar com isso em um contexto educacional. Finalmente, o trabalho menciona a utilização de jogos como uma possível ferramenta para o auxílio no ensino, e descreve e analisa algumas dessas ferramentas com o objetivo de que professores possam conhecer e identificar com mais facilidade alguma delas para aplicar em sala de aula.

A metodologia aplicada no trabalho foi uma pesquisa documental de natureza aplicada e objetivo descritivo. O material utilizado foi resultado da procura pelas palavras chave "Ensino", "Programação" e "Jogos" no Google Scholar e Google Search. Como resultado dessa busca, algumas listas de jogos que ensinam programação foram encontradas. Dessas listas, foram

selecionados os jogos que apareciam com maior frequência. Cada jogo selecionado foi testado pelo autor, onde os níveis iniciais foram verificados de forma a entender as suas mecânicas. Após o teste de cada jogo, foi considerada a facilidade de compreensão, de aprendizado e de intuitividade presente neles, destacando os jogos que fossem melhores para iniciantes. Essas informações foram sintetizadas na tabela da Figura 3. Depois de analisados e avaliados, foi feito um breve resumo de cada jogo.

Figura 3 – Tabela de análise de ferramentas

Jogo	Língua Portuguesa	Idade Indicada	Plataformas	Gratuidade
Algo Bot	Não	+9	Windows	Não
Blockly Games	Sim	+9	Web	Sim
CodeCombat	Sim	+13	Web	Apenas as primeiras fases são gratuitas
CodinGame	Não	+13	Web	Sim
CodeMonkey	Parcialmente	+9	Web	Não
Humans Resource Machine	Sim	+10	Windows, Linux, OS X, Android, iOS, Wii U, Nintendo Switch	Não
Lightbot	Sim	+9	Web, iOS, Android	Web: Sim, iOS e Android: Não
Scratch	Sim	+5	Web, Android, iOS	Sim
Tynker	Não	+3	Web, iOS	Não

Fonte: (GOMES et al., 2018)

O trabalho conclui afirmando que o uso de ferramentas que facilitem a comunicação e o aprendizado sempre são úteis, tanto para os professores quanto para os alunos, e também mencionou que o trabalho procurou apresentar alguns jogos digitais para o ensino de programação, fazendo um levantamento de diversas ferramentas e selecionando nove delas para apresentar e descrever. Destacou também que a lista das nove ferramentas apresentadas não é exaustiva, pois a quantidade existente de jogos para o ensino de programação é muito maior e sempre crescente.

O autor também relatou que uma das maiores dificuldades do trabalho foi, após jogar e ler sobre todos os jogos testados, decidir quais jogos deveriam entrar na lista que o trabalho dele apresenta, e quais aspectos deveriam ser selecionados na análise dele. O trabalho menciona que é importante fazer a correta identificação da ferramenta a ser utilizada em sala de aula, como ela deve ser aplicada, e como deve ser adequada ao nível escolar e cultural dos alunos. Mesmo com ferramentas lúdicas, o ensino de programação ainda é muito difícil, pois devem ser levados em consideração fatores como ritmo e experiência do aluno, além da metodologia de ensino.

Como trabalhos futuros, tem-se: a identificação e inclusão de mais jogos digitais.

Outra possibilidade também seria a avaliação da efetividade do aprendizado dos alunos em programação por meio de jogos comparando com turmas onde jogos não foram utilizados, e mais outra possibilidade seria o desenvolvimento de um jogo educacional após a análise das principais vantagens de cada ferramenta e reações dos alunos para as ferramentas apresentadas, com o objetivo de melhorar as qualidades que as mesmas apresentam.

2.7 Considerações finais

Este capítulo resumiu pesquisas que cobriram os temas mais relevantes para o trabalho, como evasão, uso de ferramentas para o auxílio de ensino de programação e gamificação. O próximo capítulo traz uma análise de possíveis ferramentas gamificadas que podem ser utilizadas no ensino de programação.

3 ANÁLISE DAS FERRAMENTAS

3.1 Introdução

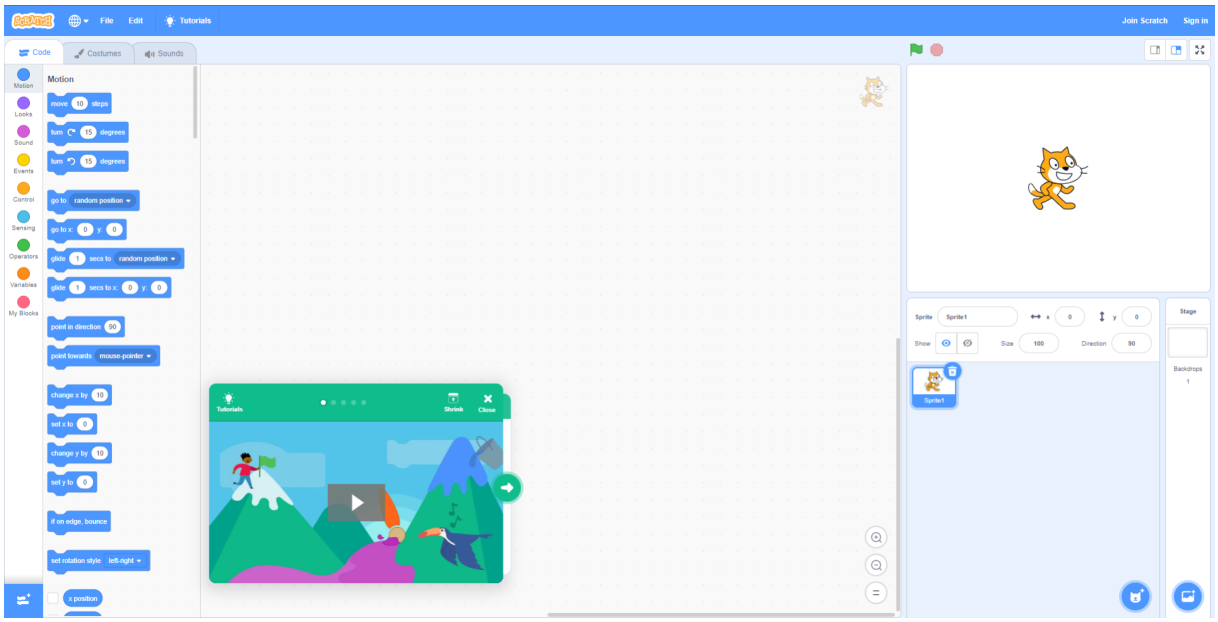
Antes da realização do experimento, foi feita uma análise técnica de diferentes ferramentas, buscando suas vantagens e desvantagens para a aplicação na disciplina, bem como a possibilidade de aplicação de elementos de gamificação. A análise foi realizada pelo autor, por meio de testes das ferramentas e de análises práticas, onde foram feitas anotações para auxiliar o entendimento da ferramenta e a identificação de suas vantagens e desvantagens.

3.2 Análise geral

3.2.1 Scratch

Uma foto da ferramenta segue na figura 4.

Figura 4 – Imagem do Scratch



Fonte: https://scratch.mit.edu.

3.2.1.1 Descrição

O Scratch é uma linguagem de programação baseada em blocos, onde você pode utilizar conceitos de programação construindo seus próprios jogos e animações.

3.2.1.2 Vantagens

Utiliza linguagem natural, muito bom para completos iniciantes.

Dá ao usuário a liberdade de criar a aplicação que quiser (dentro dos limites da plataforma, inclusive, no site oficial deles foram feitas várias animações, jogos, etc). Inclusive podem ser passados trabalhos para os alunos para que sejam desenvolvidos jogos/animações.

Tem muito material de apoio, inclusive em português.

Possui muitos artigos científicos que o utilizam.

3.2.1.3 Desvantagens

Não é bem "gamificado", ele é mais uma linguagem de programação simplificada. Não cobre sintaxe das linguagens de programação e conceitos mais avançados como orientação a objetos (até possui de uma forma limitada) e estrutura de dados (o que para uma disciplina de algoritmo não chega a ser tanto um problema, só a parte da sintaxe.)

3.2.2 CodeCombat

Uma foto da ferramenta segue na figura 5.

Figura 5 – Imagem do CodeCombat

Fonte: https://codecombat.com.

3.2.2.1 Descrição

CodeCombat é um jogo educacional para o ensino de programação, em que são oferecidos desafios que devem ser resolvidos utilizando conceitos de programação.

3.2.2.2 Vantagens

Utiliza linguagens de programação reais como Python, JavaScript, Lua, C++.

Exercita muitos dos principais fundamentos da programação, como sintaxe, métodos, loops, if/else, strings. Inclui também módulos de desenvolvimento de jogos e web.

É disponibilizado em vários idiomas, inclusive em português.

3.2.2.3 Desvantagens

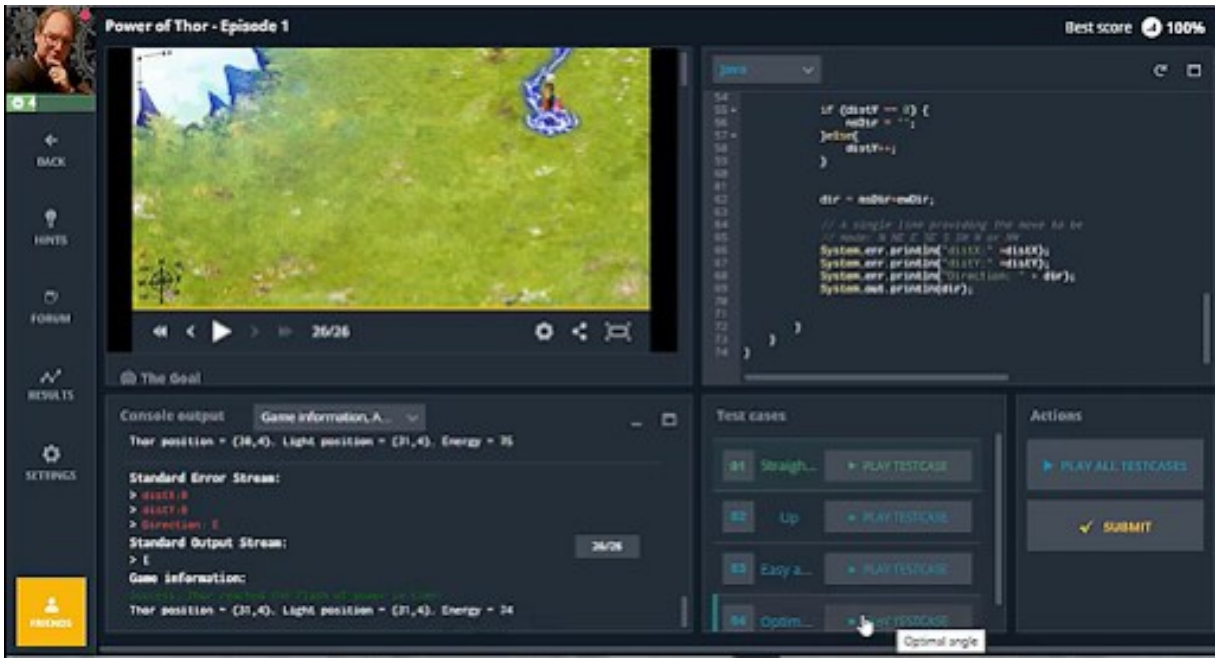
Não dá ao usuário a liberdade de construir as próprias aplicações como o Scratch.

Para usuários gratuitos só está disponível o primeiro módulo, e algumas linguagens como C++ e Java só podem ser utilizadas por usuários pagos.

3.2.3 CodinGame

Uma foto da ferramenta segue na figura 6.

Figura 6 – Imagem do CodinGame



Fonte: https://www.codingame.com/home.

3.2.3.1 Descrição

O CodinGame é uma ferramenta gamificada que oferece desafios, como um que envolve a utilização de if/else para uma espaçonave atacar vários inimigos, em mais de 25 linguagens de programação como C, C++, Python, Java além de competições.

Alguns exemplos de desafios:

The descent: Um jogo em que você usa um loop para evitar que a sua espaçonave bata em uma montanha.

Power of Thor: Um jogo em que você utiliza if else para ajudar Thor a chegar em um local onde ele possa recuperar seus poderes.

Mars Lander: Um jogo em que você utiliza if else para ajudar uma espaçonave a pousar em Marte. Dentre várias outras, muitas desenvolvidas pelos próprios usuários.

3.2.3.2 Vantagens

Como o CodeCombat, utiliza linguagens de programação reais, inclusive utiliza um número bem maior de linguagens que não estão inclusas no CodeCombat, como Java, Haskell, PHP, C#.

Exercita os fundamentos da programação, inclusive alguns mais avançados como grafos, programação dinâmica, sistemas multi agentes.

Tem desafios de competição e de cooperação em equipes.

3.2.3.3 Desvantagens

Não está disponível em português.

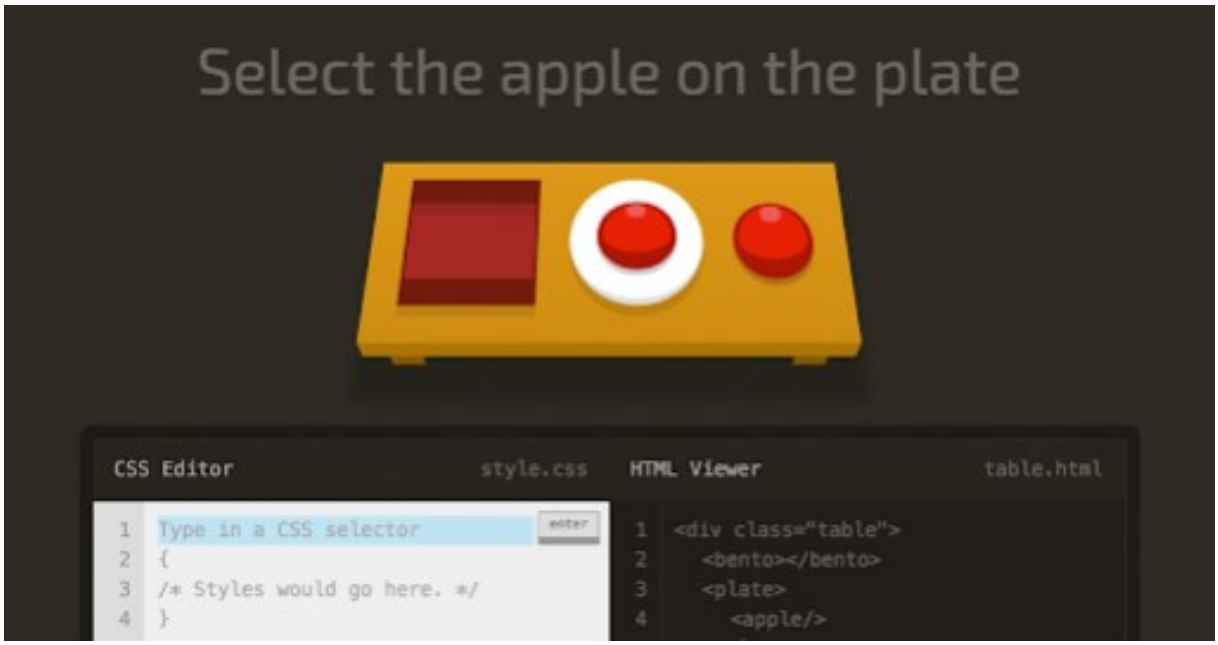
Tem atividades para todos os níveis, mas no geral programadores mais experientes irão tirar um maior proveito da ferramenta.

Não possui um “passo a passo” tão bem definido.

3.2.4 CSS Diner

Uma foto da ferramenta segue na figura 7.

Figura 7 – Imagem do CSS Diner



Fonte: https://flukeout.github.io.

3.2.4.1 Descrição

No CSS Diner, você pratica conceitos de CSS em uma mesa de jantar.

3.2.4.2 Vantagens

Exercita os fundamentos de CSS.

3.2.4.3 Desvantagens

Não tem em português.

Foca em uma habilidade muito específica.

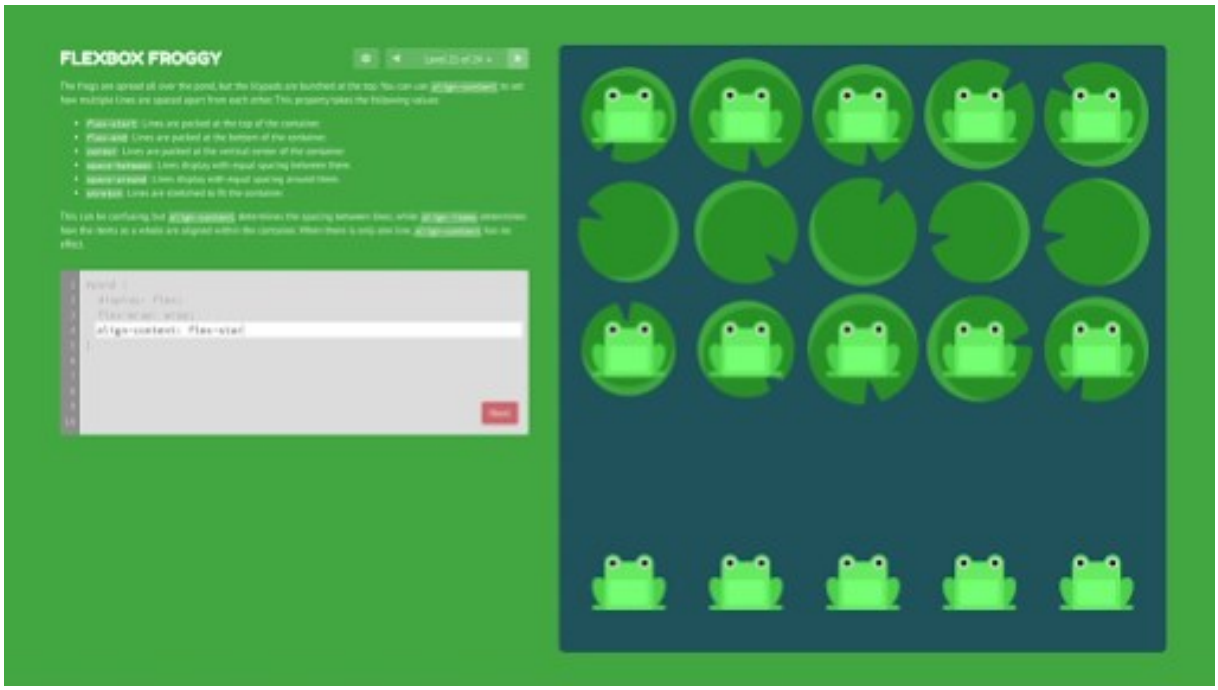
3.2.5 Flexbox Froggy

Uma foto da ferramenta segue na figura 8.

3.2.5.1 Descrição

Tem o objetivo de, utilizando o CSS Flexbox, colocar os sapos nas vitórias-régias corretas.

Figura 8 – Imagem do Flexbox Froggy



Fonte: https://flexboxfroggy.com.

3.2.5.2 Vantagens

Exercita o uso do CSS Flexbox.

3.2.5.3 Desvantagens

Não está disponível em português. Foca em uma habilidade muito específica.

3.2.6 Flexbox Defense

Uma foto da ferramenta segue na figura 9.

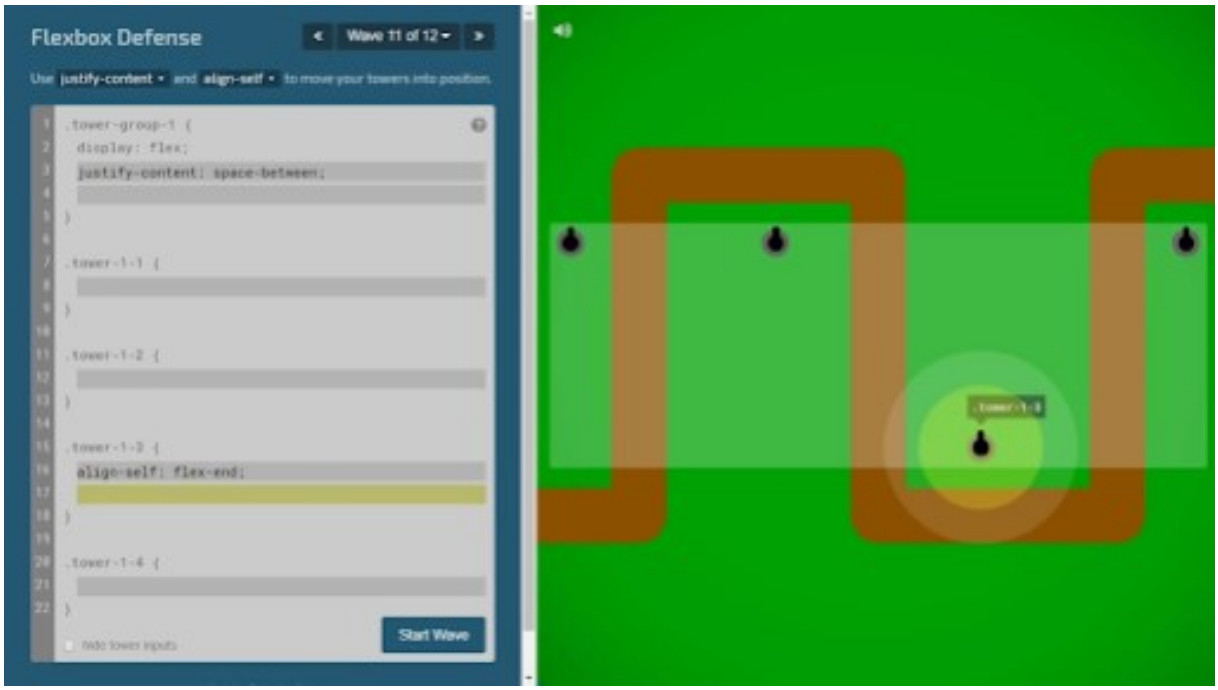
3.2.6.1 Descrição

Tem o objetivo de, utilizando o CSS Flexbox, posicionar torres para se proteger de inimigos.

3.2.6.2 Vantagens

Exercita o uso do CSS Flexbox.

Figura 9 – Imagem do Flexbox Defense



Fonte: http://www.flexboxdefense.com.

3.2.6.3 Desvantagens

Não tem em português. Foca em uma habilidade muito específica.

3.2.7 Grid Garden

Uma foto da ferramenta segue na figura 10.

3.2.7.1 Descrição

Tem o objetivo de, utilizando o CSS Grid, montar um jardim.

3.2.7.2 Vantagens

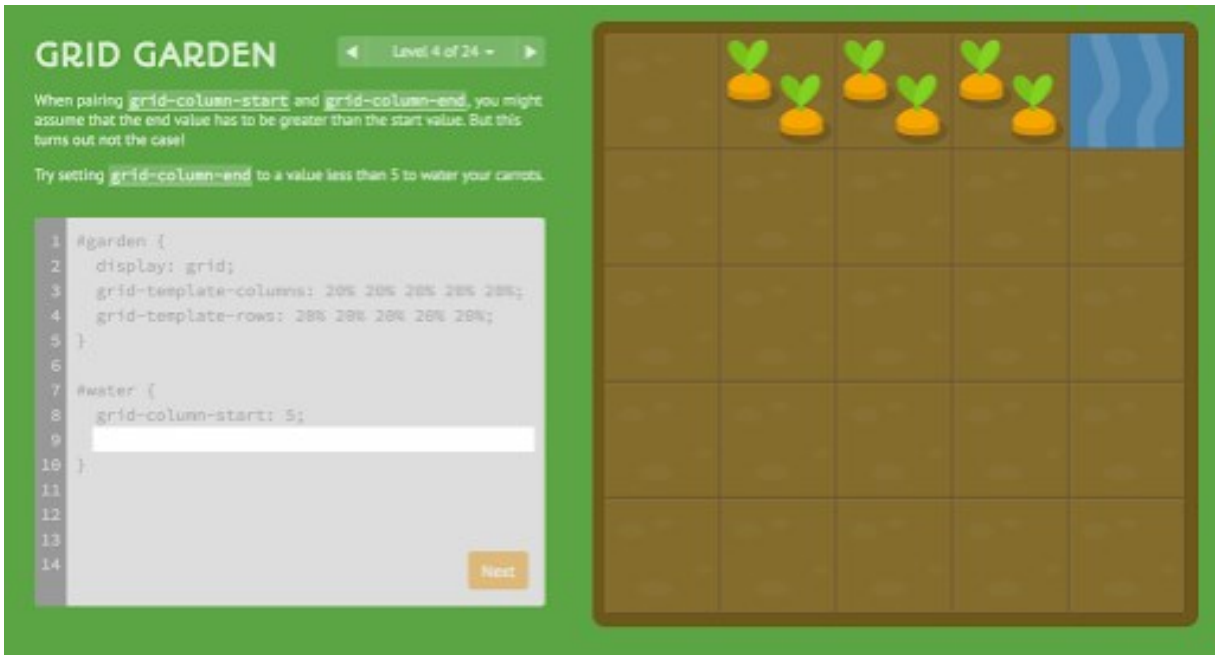
Exercita o uso do CSS Grid.

3.2.7.3 Desvantagens

Não tem em português.

Foca em uma habilidade muito específica.

Figura 10 – Imagem do Grid Garden

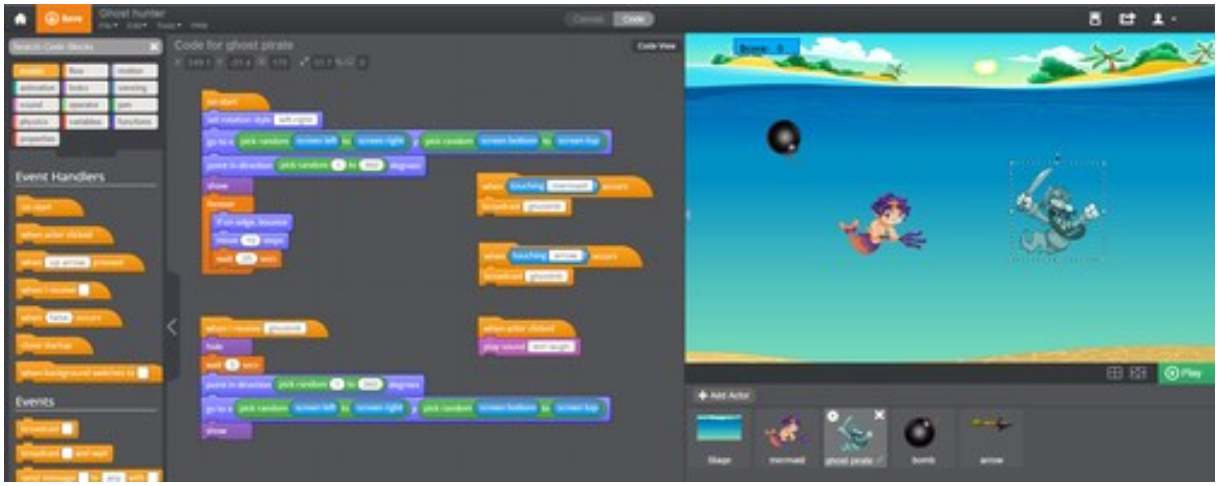


Fonte: https://cssgridgarden.com.

3.2.8 Tynker

Uma foto da ferramenta segue na figura 11.

Figura 11 – Imagem do Tynker



Fonte: https://www.tynker.com.

3.2.8.1 Descrição

É uma plataforma onde o jogador pode desenvolver projetos e cumprir desafios utilizando linguagens de programação/marcação/estilização.

3.2.8.2 Vantagens

De uma forma similar ao Scratch, dá ao usuário a liberdade de realizar projetos.

Utiliza linguagens de programação/script reais como Python, Java e JavaScript.

Possui muitos “templates” de projetos.

3.2.8.3 Desvantagens

Não está disponível em Português.

Não possui tanto material de apoio comparado ao Scratch, que é uma ferramenta bastante similar.

3.2.9 SQL Murder Mystery

Uma foto da ferramenta segue na figura 12.

Figura 12 – Imagen do SQL Murder Mistery

Fonte: https://mystery.knightlab.com.

3.2.9.1 Descrição

É um jogo em que o jogador utiliza SQL para resolver um mistério.

3.2.9.2 Vantagens

Exercita o uso do SQL.

3.2.9.3 Desvantagens

Não está disponível em português. Foca em uma habilidade muito específica. Requer conhecimento prévio de SQL

3.2.10 Elevator Saga

Uma foto da ferramenta segue na figura 13.

Figura 13 – Imagem do Elevator Saga

Fonte: https://play.elevatorsaga.com.

3.2.10.1 Descrição

Tem o objetivo de cumprir o desafio de transportar o máximo de pessoas possível, utilizando a linguagem JavaScript.

3.2.10.2 Vantagens

Exercita a linguagem Javascript.

3.2.10.3 Desvantagens

Não tem em português.

Foca em uma habilidade muito específica.

Requer conhecimento prévio de JavaScript.

3.2.11 CheckIO

Uma foto da ferramenta segue na figura 14.

Figura 14 – Imagem do CheckIO

Fonte: https://checkio.org.

3.2.11.1 Descrição

É um jogo de estratégia que permite o aprendizado de TypeScript ou Python por uma série de desafios.

3.2.11.2 Vantagens

Exercita os fundamentos da programação em Python ou Javascript.

3.2.11.3 Desvantagens

Não está disponível em português

3.2.12 RoboCode

Uma foto da ferramenta segue na figura 15.

Figura 15 – Imagem do RoboCode

Fonte: https://robocode.sourceforge.io.

3.2.12.1 Descrição

É um jogo de programação, com o objetivo de programar um robô para combater outros robôs utilizando Java.

3.2.12.2 Vantagens

Utiliza plataformas Java, uma linguagem de programação real

Estimula a competição entre os jogadores.

O processo para o desenvolvimento de um robô reflete o processo de desenvolvimento de um software.

3.2.12.3 Desvantagens

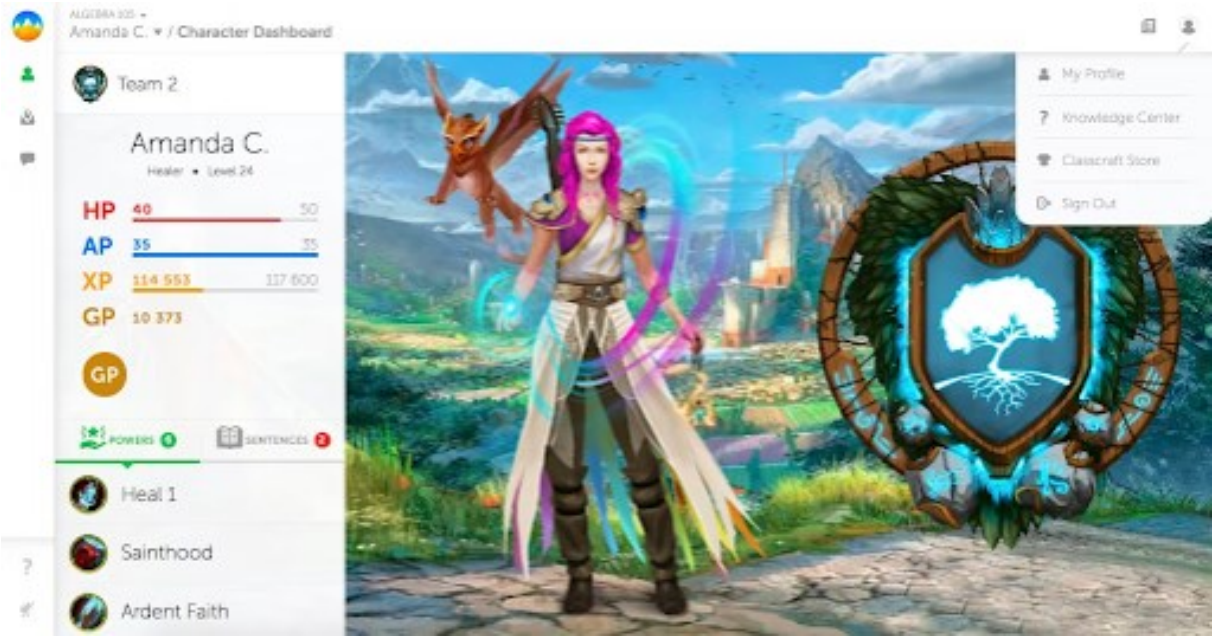
Não está traduzido para o português.

Completos iniciantes na programação podem ter dificuldades em utilizá-lo.

3.2.13 ClassCraft

Uma foto da ferramenta segue na figura 16.

Figura 16 – Imagem do ClassCraft



Fonte: https://www.classcraft.com/pt/.

3.2.13.1 Descrição

O ClassCraft é uma ferramenta que transforma a sala de aula em um ambiente gamificado, podendo o instrutor transformar um conjunto de atividades em missões, e os alunos podem criar personagens que mais se enquadram no seu perfil, escolhendo sua classe. Um

personagem tem HP (pontos de vida, a energia vital do personagem), pontos de ação (que são utilizados para o personagem realizar ações dentro das atividades) e pontos de experiência que são ganhos quando as missões são concluídas.

3.2.13.2 Vantagens

Torna o ambiente de estudo mais divertido, e o sistema de missões faz o aluno "sentir melhor" o seu progresso na disciplina. Estimula o trabalho em equipe pois, no ambiente, os alunos são divididos em equipes.

3.2.13.3 Desvantagens

Não possui funcionalidades voltadas ao ensino de programação, é mais voltada para gamificar a sala de aula.

3.2.14 RoboMind

Uma foto da ferramenta segue na figura 17.

3.2.14.1 Descrição

O RoboMind é uma ferramenta que permite que você programe um robô para que ele se movimente e interaja dentro do ambiente oferecido, podendo, por exemplo, manipular objetos e desviar de obstáculos.

3.2.14.2 Vantagens

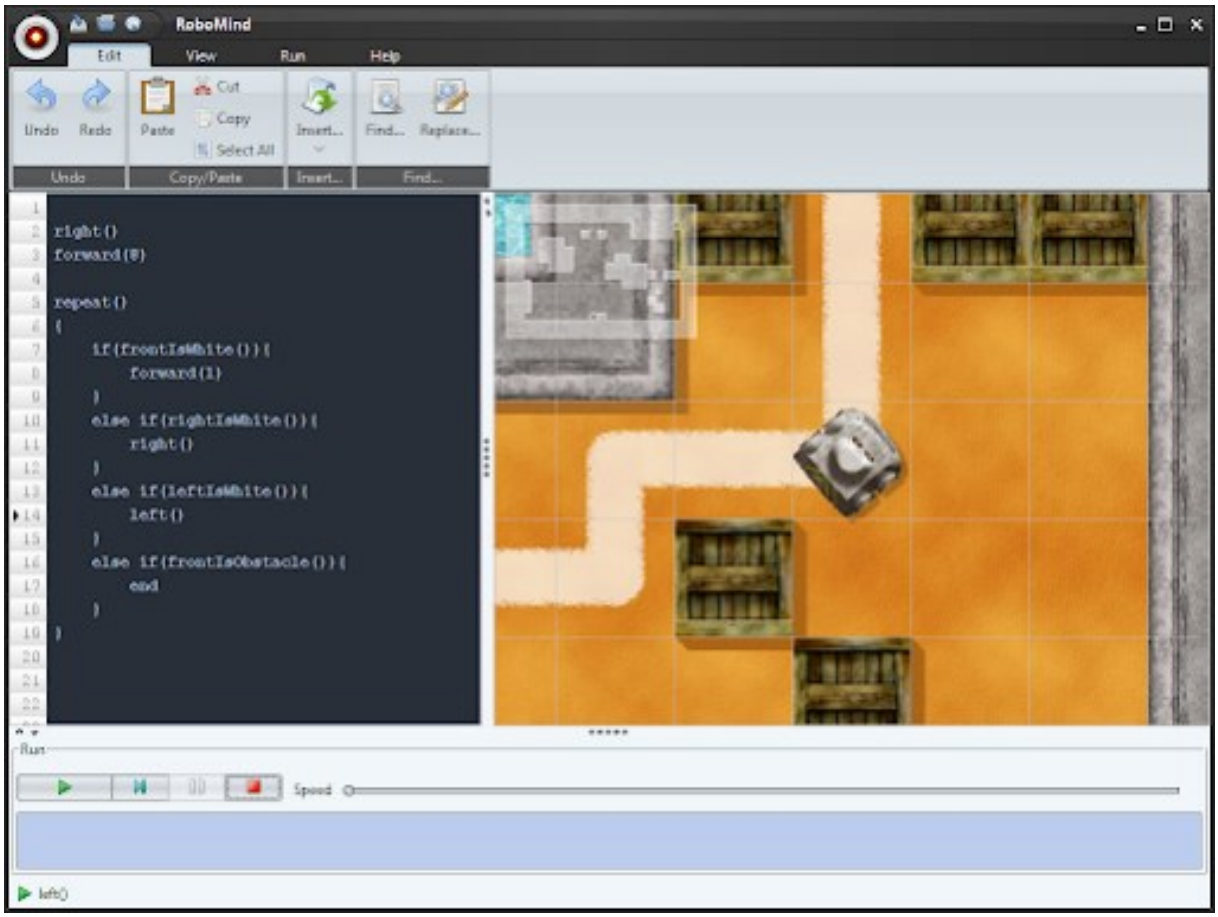
Exercita muitos dos principais conceitos de computação, como condicionais e loops, e até mesmo serve como introdução para alguns conceitos mais avançados como Máquina de Turing.

Está disponível em português

3.2.14.3 Desvantagens

Não envolve muita gamificação (de certa forma até envolve apesar de não ter sido a intenção, pois o seu funcionamento é parecido com o de um video-game).

Figura 17 – Imagem do RoboMind



Fonte: https://www.robomind.net/en/.

Apesar de estar disponível em português, muito do material de apoio dele não está.

3.2.15 Conclusão

Considerando as limitações de tempo e de recursos para aplicação dos jogos, devem ser escolhidas poucas ferramentas. A ferramenta CodeCombat foi a que melhor se encaixou nos objetivos, pois está disponível em português, aborda o aprendizado de uma forma gamificada e utiliza a linguagem de programação Python, excelente para iniciantes. Uma análise mais aprofundada desta ferramenta será feita a seguir.

3.3 Análise do CodeCombat

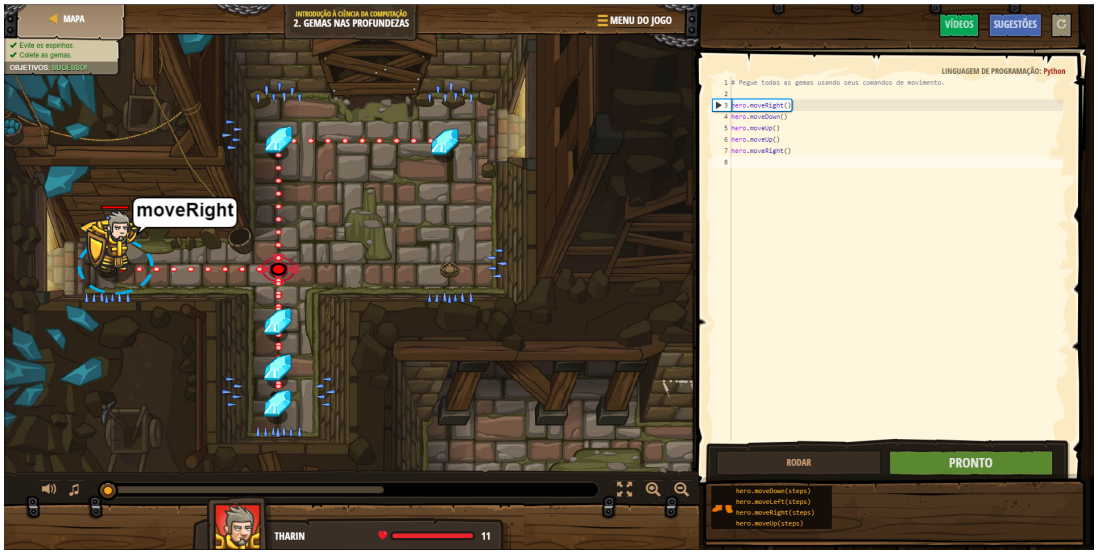
3.3.1 Introdução

A seguir será realizada uma análise mais aprofundada do que a realizada anteriormente da ferramenta CodeCombat, mostrando suas características, relação com os conceitos de

gamificação, conteúdos abordados, e integração com a disciplina envolvida no experimento.

O CodeCombat se trata de uma ferramenta para que os alunos aprendam a programar enquanto jogam um jogo. Nele, o usuário pode controlar um personagem e utilizá-lo para escapar de masmorras (Figura 18), coletar tesouros e derrotar inimigos, tudo isso utilizando linguagens de programação. Ele também oferece aos professores a possibilidade de criar equipes dentro da ferramenta e colocar os alunos dentro da mesma equipe (Figura 19, permitindo monitorar o progresso de cada aluno, e criar competições. As linguagens de programação oferecidas pela ferramenta são: Python, JavaScript, CoffeeScript, Lua, C++ e Java (Figura 20).

Figura 18 – Imagem de exemplo do CodeCombat

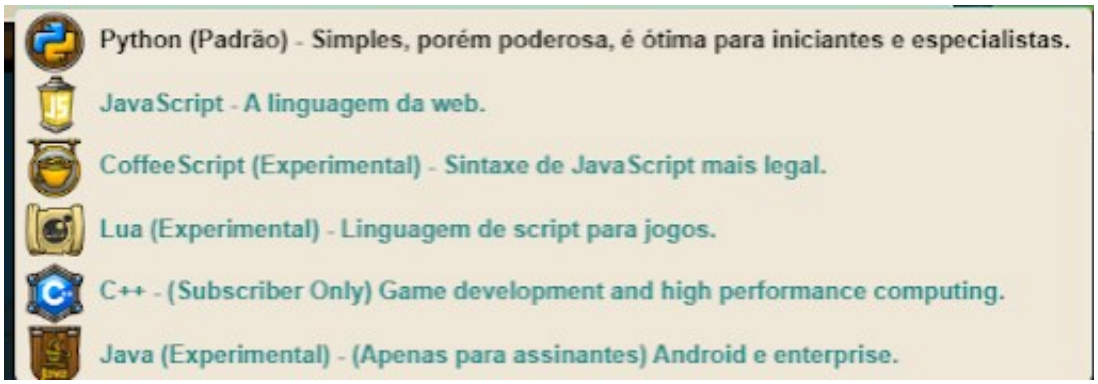


Fonte: https://codecombat.com.

Figura 19 – Imagem da perspectiva de líder de um time

Fonte: https://codecombat.com.

Figura 20 – Linguagens de programação utilizadas



Fonte: https://codecombat.com.

3.3.2 Características gamificadas do CodeCombat

3.3.2.1 Criação de avatares

A ferramenta permite que o usuário crie avatares com diferentes valores de status de ataque e vida, seguindo o estilo e dificuldade que o usuário prefira jogar, de uma forma mais ofensiva ou defensiva, ou mais fácil ou difícil. Um exemplo de criação de avatares está na Figura 21.

Figura 21 – Exemplo de avatar do tipo Guerreiro

Fonte: https://codecombat.com.

3.3.2.2 Mapa de missões

O jogo oferece para o jogador um mapa de missões, em que cada missão cobra algum conceito de programação. Para o primeiro módulo são cobrados os conceitos de sintaxe básica, argumentos, strings, loops while, variáveis, algoritmos. Exemplificado na figura 22.

Figura 22 – Mapa de missões

Fonte: https://codecombat.com.

3.3.2.3 Níveis

Para cada missão, o jogo oferece uma série de objetivos para serem cumpridos, sempre cobrando algum conceito de programação. Também é percetível a utilização de uma barra de progresso para mostrar o andamento da missão e pontos de vida que são perdidos caso o seu avatar seja atacado por algum inimigo ou colida com algum obstáculo, como espinhos. Exemplificado na figura 23.

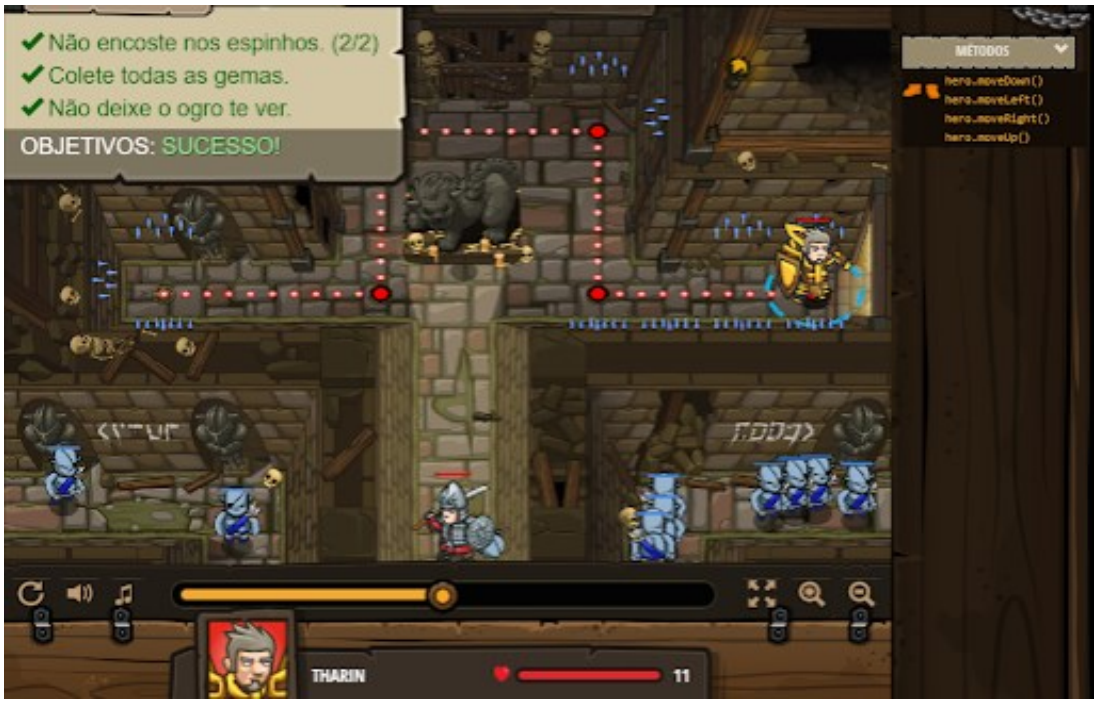
3.3.2.4 Dicas

Caso o jogador cometa erros, o jogo automaticamente oferece dicas ao mesmo por meio de “pop-ups”, como mostrado na Figuras 24 e 25.

3.3.2.5 Conquistas

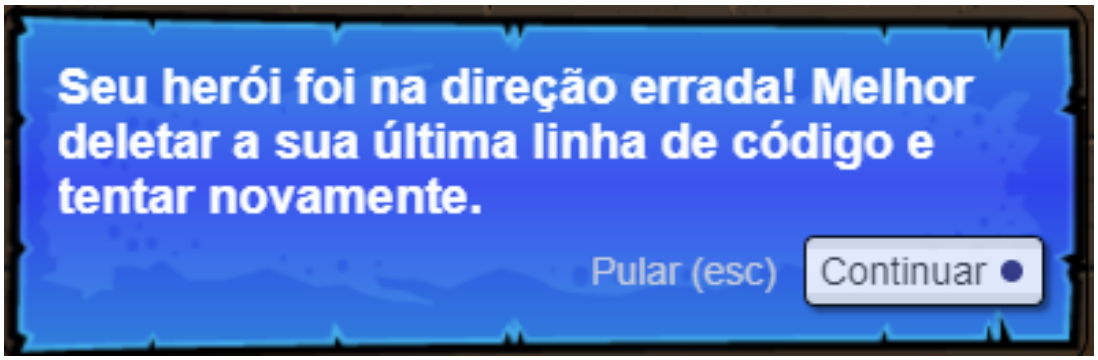
Conforme o jogador vai concluindo suas missões, ele vai recebendo conquistas para medir o seu progresso, exemplicado na Figura 26.

Figura 23 – Exemplo de nível



Fonte: https://codecombat.com.

Figura 24 – Exemplo de dicas



Fonte: https://codecombat.com.

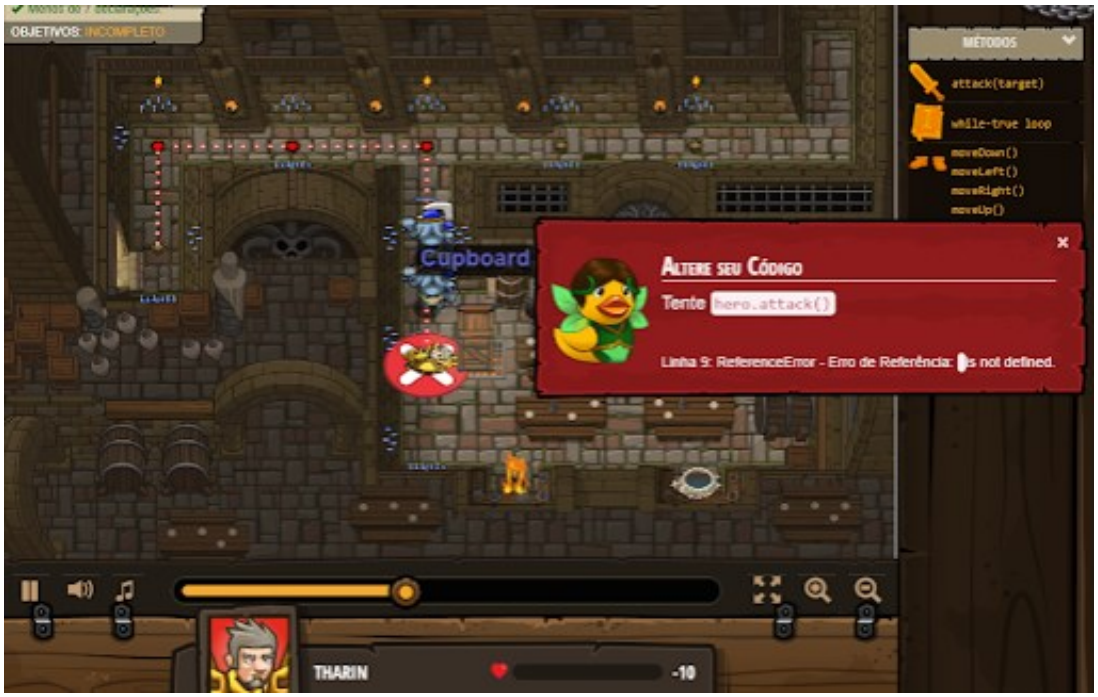
3.3.2.6 Side quests

À medida que o jogador conclui os níveis, vão aparecendo desafios opcionais (funcionando como side quests) para reforçar os conceitos. Exemplificado na figura 27.

3.3.2.7 Clãs e competições

A ferramenta permite que se criem “equipes”, onde o líder pode monitorar o progresso dos membros da equipe e os membros podem desafiar uns aos outros para desafios. Exemplificado na figura 28.

Figura 25 – Exemplo de dicas



Fonte: https://codecombat.com.

Figura 26 – Conquistas

ACHIEVEMENTS

Conquista	Últimos Ganhos	Montante
Cupboards of Kithgard Languished	setembro 5º 2022	
Cell Commentary Complete	setembro 4º 2022	
Completed 10 Strings Levels	setembro 5º 2022	
Completed 20 Arguments Levels	setembro 5º 2022	
Completed 10 Arguments Levels	setembro 4º 2022	
Completed 20 Basic Syntax Levels	setembro 5º 2022	
Completed 10 Basic Syntax Levels	setembro 4º 2022	
Kithgard Librarian Complete	setembro 4º 2022	
Fire Dancing Complete	setembro 4º 2022	

Fonte: https://codecombat.com.

Figura 27 – Desafios

Fonte: https://codecombat.com.

Figura 28 – Clãs e competições

Fonte: https://codecombat.com.

3.3.3 Módulos do CodeCombat

O CodeCombat tem vários módulos. O primeiro é disponibilizado gratuitamente, e os módulos seguintes podem ser acessados mediante assinatura. Os principais conceitos abordados em cada módulo estão descritos a seguir.

Módulo 1: Introdução a Ciência da Computação: Sintaxe básica, argumentos, strings, loops while, variáveis, algoritmos.

Módulo 2: Desenvolvimento de jogos 1: Sintaxe básica, argumentos, usar objetos do jogo,

construir labirintos, criar um jogo jogável e compartilhável.

Módulo 3: Desenvolvimento de web 1: HTML básico, CSS básico, alterar páginas da web existentes, criar uma página web compartilhável.

Módulo 4: Ciência da computação 2: Sintaxe básica, argumentos, strings, loops while, variáveis, comandos if, funções, parâmetros, strings avançadas.

Módulo 5: Desenvolvimento de jogos 2: Sintaxe básica, funções, strings, comandos if, argumentos, manipulação básica de entrada, IA básica do jogo, criar um jogo jogável e compartilhável.

Módulo 6: Desenvolvimento de web 2: Sintaxe básica, strings, loops while, variáveis, argumentos, comandos if, funções, HTML básico, CSS básico, Web script básico, HTML avançado, JavaScript básico, manipulação básica de eventos, criar uma página web interativa compartilhável.

Módulo 7: Ciência da computação 3: Sintaxe básica, argumentos, comandos if, variáveis, strings, loops while, aritmética, strings avançadas, manipulação de entrada, funções, parâmetros, lógica booleana, comandos break, comandos continue.

Módulo 8: Desenvolvimento de jogos 3: Sintaxe básica, dados de eventos.

Módulo 9: Ciência da Computação 4: Sintaxe básica, argumentos, variáveis, strings, comandos if, aritmética, loops while, arrays, funções, parâmetros, objetos literais, laço for, lógica booleana.

Módulo 10: Ciência da Computação 5: Sintaxe básica, argumentos, variáveis, strings, comandos if, loops while, arrays, laço for, funções, lógica booleana, algoritmos, comandos break, aritmética, objetos literais, parâmetros, vetores, operações matemáticas, recursividade.

Módulo 11: Ciência da Computação 6: Sintaxe básica, argumentos, variáveis, strings, comandos if, loops while, arrays, laço for, funções, lógica booleana, algoritmos, comandos break, aritmética, objetos literais, parâmetros, vetores, operações matemáticas, recursividade.

3.4 Níveis do Módulo 1 do CodeCombat

Os conceitos abordados em cada nível do Módulo 1 (Introdução a Ciência da Computação) estão listados a seguir.

Nível 1: Masmorra Kithgard: Sintaxe básica.

Nível 2: Gemas nas profundezas: Sintaxe básica.

Nível 3: Guarda Sombrio: Sintaxe básica.

Nível 4: Mina inimiga: Sintaxe básica, argumentos, inteiros.

Nível 5: Nomes próprios: Sintaxe básica, argumentos, strings.

Nível 6: Comentários na cela: Sintaxe básica, argumentos, strings.

Nível 7: Bibliotecário de Kithgard: Sintaxe básica, argumentos, strings.

Nível 8: O prisioneiro: Sintaxe básica, argumentos, strings, inteiros.

Nível 9: Dança no fogo: Sintaxe básica, loops.

Nível 10: Labirinto Kithmaze: Sintaxe básica, loops, argumentos, inteiros.

Nível 11: Continue a descer: Sintaxe básica, loops, argumentos, inteiros.

Nível 12: Porta misteriosa: Sintaxe básica, loops, argumentos, strings.

Nível 13: Bata e Corra: Sintaxe básica, loops, argumentos, strings.

Nível 14: Armários de Kithgard: Sintaxe básica, loops, argumentos, inteiros, strings.

Nível 15: Inimigo Conhecido: Sintaxe básica, loops, argumentos, strings, variáveis.

Nível 16: Mestre dos nomes: Sintaxe básica, argumentos, variáveis.

Nível 17: Labirinto Final: Sintaxe básica, loops, variáveis.

Nível 18: Portões de Kithgard: Sintaxe básica, loops, argumentos, inteiros.

3.4.1 Problemas da ferramenta

3.4.1.1 Limitações para usuários gratuitos

O jogo oferece somente o primeiro módulo e a utilização da linguagem Python para usuários gratuitos.

3.4.1.2 Material de apoio

O material de apoio da ferramenta não possui legendas em português.

3.5 Considerações

Levando em consideração as características de todas as ferramentas, a ferramenta CodeCombat foi selecionada devido elementos mencionados anteriormente, além de cobrir temas relevantes para a disciplina em que o experimento será aplicado, melhor explicado no capítulo a seguir.

4 METODOLOGIA

4.1 Introdução

Para este trabalho, foi utilizada a ferramenta selecionada (CodeCombat) em um experimento com a disciplina de Fundamentos de Programação. Para isso, foram utilizados dois grupos, onde o grupo experimental foi a turma ofertada para o curso de Ciência da Computação (CC), ministrada pelo professor Dr. Markos Oliveira Freitas, e o grupo de controle foi a turma ofertada para o curso de Engenharia de Software (ES), ministrada pelo professor Dr. Pablo Luiz Braga Soares. O design do experimento, adaptado do experimento realizado por CUERVO-CELY et al. (2022), será apresentado a seguir.

4.2 Design do experimento

O experimento foi realizado com a separação do grupo de experimento, onde foi feita a aplicação do CodeCombat como forma de exercitação dos conteúdos, e o de controle, onde a disciplina foi conduzida de forma tradicional. O intuito do experimento é medir a motivação no início e fim do experimento, aplicando o questionário IMI (Intrinsic Motivation Inventory), como feito anteriormente em um experimento similar por MOREIRA (2021), nos dois grupos. Devido à necessidade de analisar os resultados antes do final do semestre, o experimento foi aplicado somente na primeira metade do semestre. Para o questionário, foram selecionadas e adaptadas as perguntas mais relevantes do IMI para o contexto. O experimento seguiu os passos indicados na Figura 29:

Figura 29 – Fluxograma de design do experimento

Fonte: Elaborada pelo autor.

As etapas 1 e 5 foram aplicadas, respectivamente, no início e fim do experimento. Já as etapas 2, 3 e 4 foram aplicadas ciclicamente, seguindo a aplicação de novos conteúdos.

4.3 Disciplina de Fundamentos de Programação

A disciplina de Fundamentos de Programação possui o objetivo de introduzir aos alunos o pensamento computacional, a elaboração de algoritmos e a introdução a linguagens de programação. A disciplina possui duas aulas semanais de duas horas, uma delas é teórica, realizada em sala de aula, e a outra é prática, realizada em laboratório. A linguagem de programação utilizada é Python. O plano de ensino da turma do grupo experimental pode ser visualizado no Anexo A, enquanto o da turma do grupo de controle pode ser visualizado no Anexo B. As turmas do semestre 2023.1 foram ministradas pelos professores Dr. Markos Oliveira Freitas (grupo experimental) e Dr. Pablo Luiz Braga Soares (grupo de controle).

4.4 Integração do CodeCombat com a disciplina de Fundamentos de Programação

Foi feita uma análise de quais conceitos cada nível do CodeCombat aborda, de acordo com os conceitos descritos na Seção 3.4, e foi realizada uma comparação com cada aula do plano da disciplina. Após isso, foi criado um plano junto ao professor da disciplina do grupo experimental, mostrado na 1. Alguns conteúdos, que são tradicionalmente abordados do meio para o final da disciplina, como tópicos de função e repetição, foram trazidos para o início para se adequar aos conteúdos do CodeCombat. Da mesma forma, conteúdos que não são abordados nos níveis aplicados do CodeCombat foram levados para a segunda metade do semestre letivo. Também foi sentida uma necessidade de bloquear os níveis do CodeCombat para que os níveis seguissem o conteúdo da disciplina. Entretanto, para evitar uma demora muito grande entre as liberações, foi decidido que os níveis não seriam necessariamente liberados somente depois da aula sobre o tópico, e sim aos poucos durante a disciplina.

4.5 Questionário utilizado

O questionário selecionado para ser utilizado foi o Intristic Motivation Inventory (IMI), devido à sua relevância para os objetivos da pesquisa (Anexo C). As categorias selecionadas, julgadas como mais relevantes para o contexto foram: Interesse, Competência Percebida, Esforço/Importância, Pressão/Tensão e Valor/Utilidade. As perguntas selecionadas para o questionário inicial e final serão listadas nas Tabelas 2 e 3. A ordem das perguntas foi aleatorizada, para evitar que todas as perguntas de uma mesma categoria aparecessem juntas. As respostas foram em 5 níveis, 1 a 5, onde 1 indica "Discordo fortemente"e 5 indica "Concordo fortemente".

Tabela 1 – Calendário FUP

Data	Conceito abordado na aula	Níveis liberados do CodeCombat
14/03	Introdução à disciplina e ao curso	-
16/03	Introdução à ementa/programação e abertura do questionário inicial.	-
23/03	Primeiro programa	1, 2, 3, 3a, 3b
28/03	Variáveis e expressões aritméticas	-
30/03	Variáveis e expressões aritméticas (aula prática) e encerramento do questionário inicial	-
04/04	Aula de tira-dúvidas	4, 4a, 4b, Desafio Passos Longos, 5, 5a, 5b, Desafio Perigosos, Desafio Combo hora de dormir
06/04	Feriado (Semana Santa)	-
11/04	Funções	-
13/04	Funções (aula prática)	9, 10, 11, 11a, 11b, Desafio Loop no Armazém, 12
18/04	Sem aula - SESCOMP	-
20/04	Sem aula - SESCOMP	13, 14, 14a, 14b
25/04	Comandos de repetição - enquanto e para e abertura do questionário final	15, 16, 16a, 16b, Desafio Mestre dos Nomes de Depurar, 17, Desafio Combo Lugar Seguro, 18
27/04	Comandos de repetição - enquanto e para (aula prática)	-
13/05	Encerramento do questionário final	-

Foi utilizada a marcação (R) para perguntas cujos valores dos resultados são invertidos para que o nível 1 indice um resultado mais negativo e o nível 5 indique um resultado mais positivo. As perguntas do questionário inicial foram baseadas nas perguntas do questionário final, mas se referindo ao futuro ao invés do presente ou passado. Ao final, foi calculada a média dos resultados para ter um resultado geral para a categoria. Para a turma da CC, foram feitas algumas perguntas relacionadas exclusivamente para o CodeCombat, na categoria Valor/Utilidade, além de algumas perguntas extras, junto com algumas perguntas subjetivas para coletar opiniões (Tabela 4). Como o CodeCombat não foi aplicado para a turma da ES, os questionários para o grupo de controle não contêm essas perguntas.

4.6 Ameaças à validade

Existem algumas questões que fogem ao controle desse experimento, e que podem acabar ameaçando a sua validade. Dentre elas, destacam-se:

Tabela 2 – Perguntas do questionário inicial

Pergunta	Categoria	Turmas aplicadas
Eu acho que as atividades da disciplina serão muito prazerosas.	Interesse	CC e ES
Eu acredito que as atividades da disciplina serão muito interessantes.	Interesse	CC e ES
Eu acho que as atividades da disciplina serão chatas (R).	Interesse	CC e ES
Eu acho que as atividades da disciplina não vão prender a minha atenção (R).	Interesse	CC e ES
Eu espero gostar de realizar as atividades da disciplina.	Interesse	CC e ES
Espero que as atividades da disciplina sejam divertidas de se fazer.	Interesse	CC e ES
Não estou confiante nas minhas habilidades de programação (R).	Competência Percebida	CC e ES
Eu me acho bom em programação comparado aos meus colegas.	Competência Percebida	CC e ES
Eu me acho bom em programação.	Competência Percebida	CC e ES
Eu me considero bastante habilidoso com programação.	Competência Percebida	CC e ES
Eu pretendo me esforçar bastante na disciplina.	Esforço/Importância	CC e ES
Pretendo fazer todo o possível para ir bem na disciplina.	Esforço/Importância	CC e ES
Para mim, é importante ir bem e tirar notas boas nessa disciplina.	Esforço/Importância	CC e ES
Eu me sinto nervoso para as atividades/avaliações da disciplina (R).	Pressão/Tensão	CC e ES
Estou tranquilo para as atividades/avaliações da disciplina.	Pressão/Tensão	CC e ES
Eu acredito que o uso do CodeCombat terá algum valor para mim.	Valor/Utilidade	CC
Eu acredito que o CodeCombat vai ser útil para aprender programação.	Valor/Utilidade	CC
Eu acredito que o CodeCombat vai me ajudar a aprender programação.	Valor/Utilidade	CC
Acredito que aprender programação será benéfico para mim.	Valor/Utilidade	CC e ES
Acredito que aprender programação é importante.	Valor/Utilidade	CC e ES

1. Metodologias diferentes: As turmas utilizadas como grupo experimental e de controle foram ministradas por professores diferentes, que tinham metodologias e planejamentos diferentes para a disciplina. Assim, o professor da turma de controle pode ter uma metodologia que também crie uma sensação de motivação dos alunos, levando os dois

grupos a apresentarem resultados semelhantes.

2. Quantidade de respostas: Um dos dois grupos pode ter uma quantidade pequena de participantes da pesquisa, que não seja estatisticamente significativa, e que pode enviesar os resultados
3. Diferentes participantes: Se a quantidade de respostas for pequena, pode ser que os alunos que responderem ao questionário inicial sejam diferentes dos alunos que responderem ao questionário final, podendo levar a uma comparação injusta.

4.7 Considerações finais

Esta seção trouxe a metodologia utilizada na adaptação da disciplina de Fundamentos de Programação para a utilização do CodeCombat, e como isso foi avaliado pelos alunos da disciplina. O próximo capítulo apresenta e discute os resultados dessa avaliação.

Tabela 3 – Perguntas do questionário final

Pergunta	Categoria	Turmas aplicadas
As atividades da disciplina estão sendo muito prazerosas.	Interesse	CC e ES
As atividades da disciplina estão sendo muito interessantes.	Interesse	CC e ES
As atividades da disciplina estão sendo chatas (R).	Interesse	CC e ES
Eu acho que as atividades da disciplina não estão prendendo a minha atenção (R).	Interesse	CC e ES
Eu estou gostando de realizar as atividades da disciplina.	Interesse	CC e ES
As atividades da disciplina até agora estão sendo divertidas.	Interesse	CC e ES
Não estou confiante nas minhas habilidades de programação (R).	Competência Percebida	CC e ES
Eu me acho bom em programação comparado aos meus colegas.	Competência Percebida	CC e ES
Eu me acho bom em programação.	Competência Percebida	CC e ES
Eu me considero bastante habilidoso com programação.	Competência Percebida	CC e ES
Eu estou me esforçando bastante na disciplina.	Esforço/Importância	CC e ES
Estou fazendo todo o possível para ir bem na disciplina.	Esforço/Importância	CC e ES
Para mim, é importante ir bem e tirar notas boas nessa disciplina.	Esforço/Importância	CC e ES
Eu me sinto nervoso para as atividades/avaliações da disciplina (R).	Pressão/Tensão	CC e ES
Estou tranquilo para as atividades/avaliações da disciplina.	Pressão/Tensão	CC e ES
Eu acredito que o uso do CodeCombat teve algum valor para mim.	Valor/Utilidade	CC
Eu acredito que o CodeCombat foi útil para aprender programação.	Valor/Utilidade	CC
Eu acredito que o CodeCombat me ajudou a aprender programação.	Valor/Utilidade	CC
Acredito que aprender programação está sendo benéfico para mim.	Valor/Utilidade	CC e ES
Acredito que aprender programação é importante.	Valor/Utilidade	CC e ES
Comparado às atividades da plataforma AME, achei as fases do CodeCombat difíceis.	Extra	CC
Me senti frustrado usando o CodeCombat.	Extra	CC

Tabela 4 – Perguntas subjetivas do questionário

Pergunta aberta
Caso você tenha se sentido frustrado utilizando o CodeCombat, por quais motivos você acredita que tenha se sentido dessa forma? Nesse espaço, envie sugestões para possíveis aplicações futuras do CodeCombat.

5 RESULTADOS

5.1 Introdução

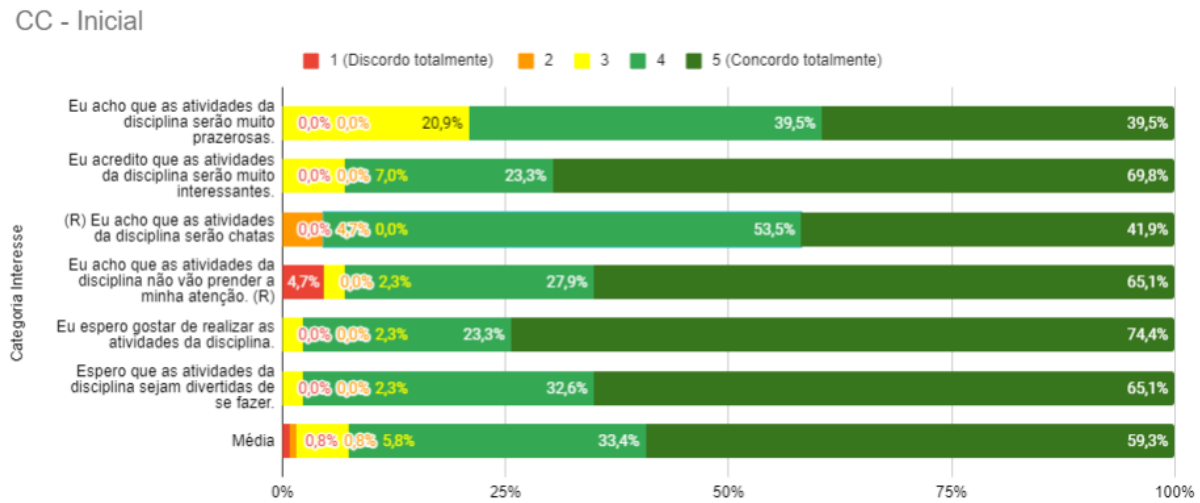
Este capítulo apresenta os resultados dos questionários aplicados. Primeiramente serão apresentados os resultados dos questionários inicial e final da turma (experimental) de Ciência da Computação (CC), onde 43 alunos responderam ao questionário inicial e 40 alunos responderam ao questionário final. De uma forma similar, serão apresentados os resultados para a turma de controle de Engenharia de Software (ES), onde 26 alunos responderam ao questionário inicial e 20 responderam ao questionário final. Após isso, será feita a comparação das duas turmas. Nos gráficos, são considerados os graus 1 e 2 como discordância da afirmação, 3 como neutro e os graus 4 e 5 como concordância.

5.2 Resultado IMI Ciência da Computação

5.2.1 Aplicação inicial do questionário

No início do semestre, conforme indicado no gráfico da Figura 30, os alunos indicaram bons índices de interesse, com uma porcentagem de concordância acima de 75% em todas as perguntas, e de 92,7% na média.

Figura 30 – Resultado IMI CC Inicial Categoria Interesse

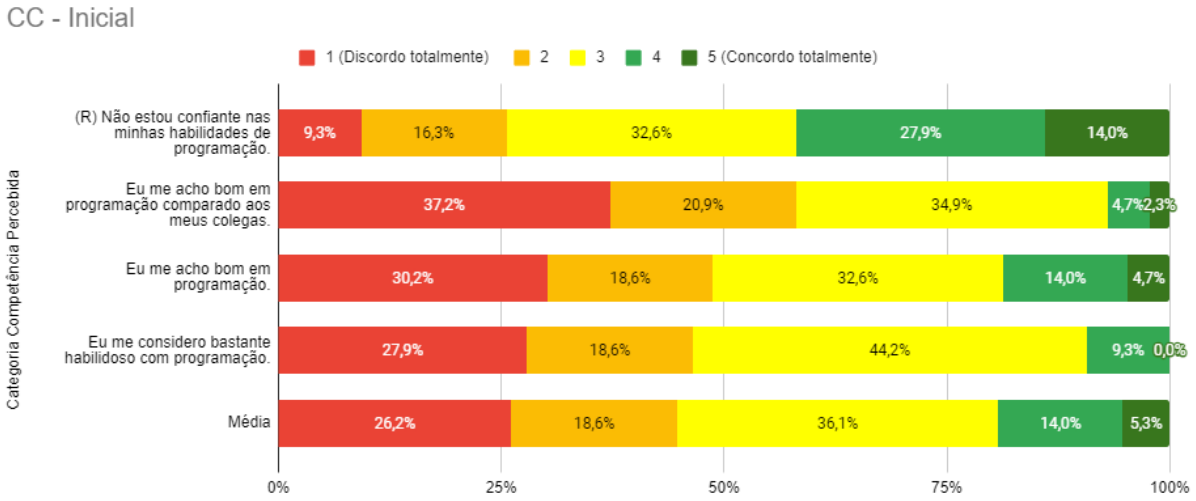


Fonte: Elaborada pelo autor.

Na categoria Competência Percebida, conforme indicado no gráfico da Figura 31, percebe-se que os alunos não se mostraram seguros com suas habilidades com programação.

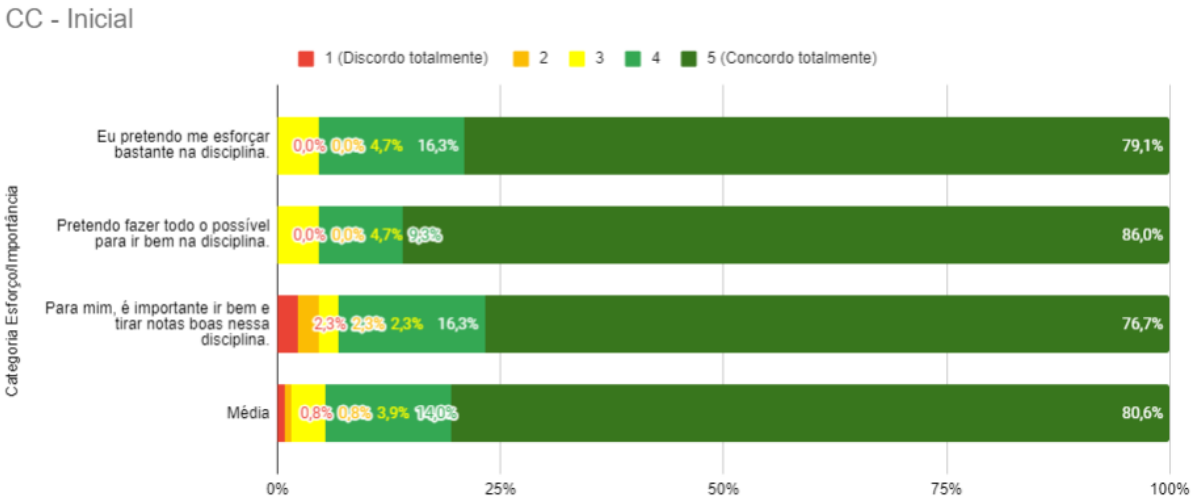
Alunos que não tinham contato prévio nem experiência com programação foram instruídos a marcar a opção 1. Em todas as perguntas, mais de 50% dos alunos responderam discordando ou de forma neutra, e 80% na média.

Figura 31 – Resultado IMI CC Inicial Categoria Competência



Na categoria de Esforço/Importância, conforme indicado no gráfico da Figura 32, mais de 90% dos alunos comunicaram a concordância da intenção de se esforçar o máximo na disciplina em todas as perguntas, com 94% de concordância na média.

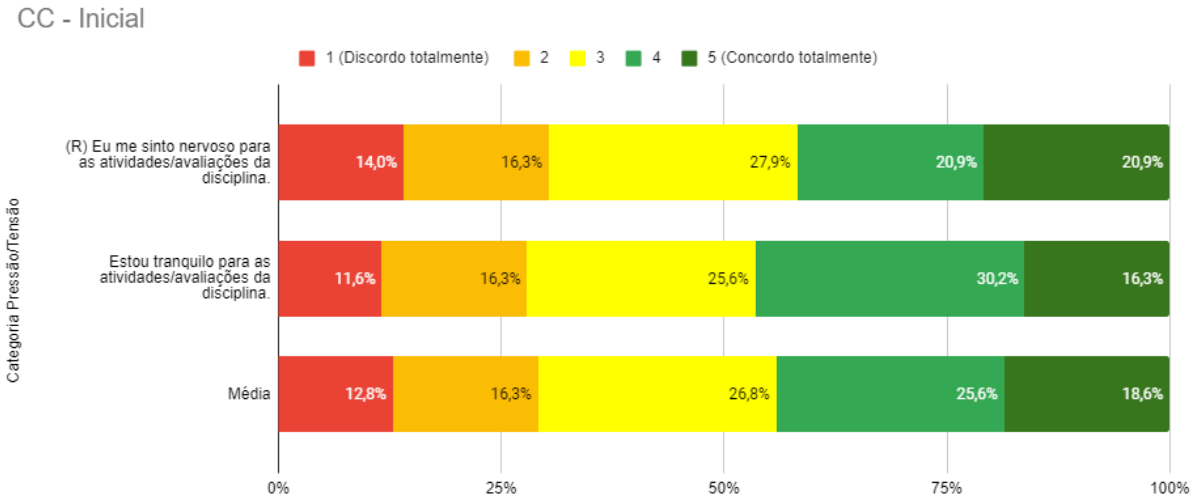
Figura 32 – Resultado IMI CC Inicial Categoria Esforço



Fonte: Elaborada pelo autor.

Na categoria de Pressão/Tensão, conforme indicado no gráfico da Figura 33, mais de 65% dos alunos comunicaram que não estavam nervosos ou estavam se sentindo de uma forma neutra para as avaliações da disciplina em todas as perguntas, e 70% na média.

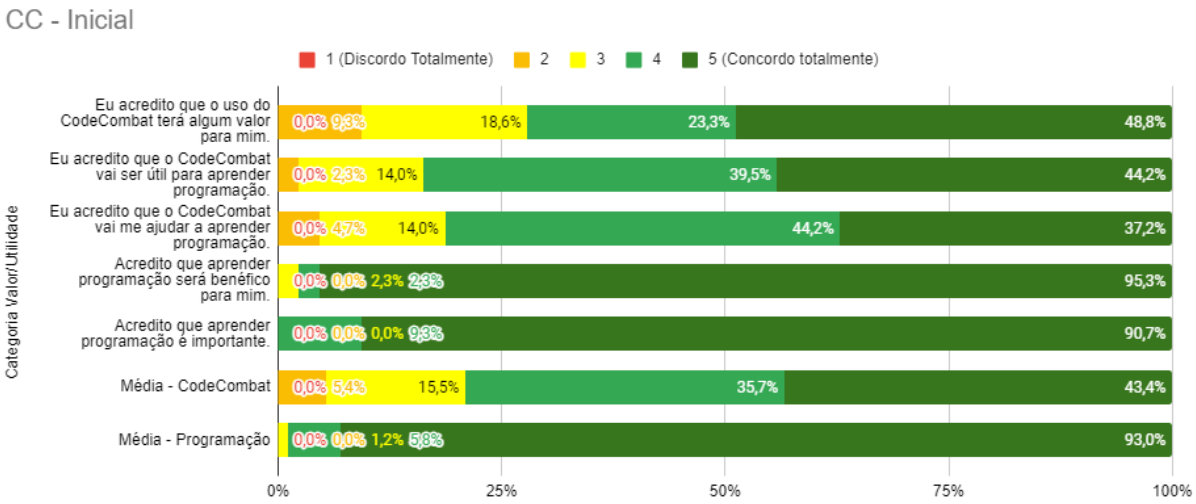
Figura 33 – Resultado IMI CC Inicial Categoria Pressão



Fonte: Elaborada pelo autor.

Na categoria de Valor/Utilidade, as três primeiras perguntas estão relacionadas ao valor da utilização do CodeCombat, e as duas últimas, ao valor do aprendizado de programação. Conforme indicado no gráfico da Figura 34, mais de 97% dos alunos concordaram com a utilidade do aprendizado de programação em todas as perguntas, e mais de 70% concordaram com a utilidade do CodeCombat em todas as perguntas. Na média, mais de 85% dos alunos demonstraram concordância do valor e da utilidade do CodeCombat, e 98% demonstraram concordância da importância da programação.

Figura 34 – Resultado IMI CC Inicial Categoria Valor

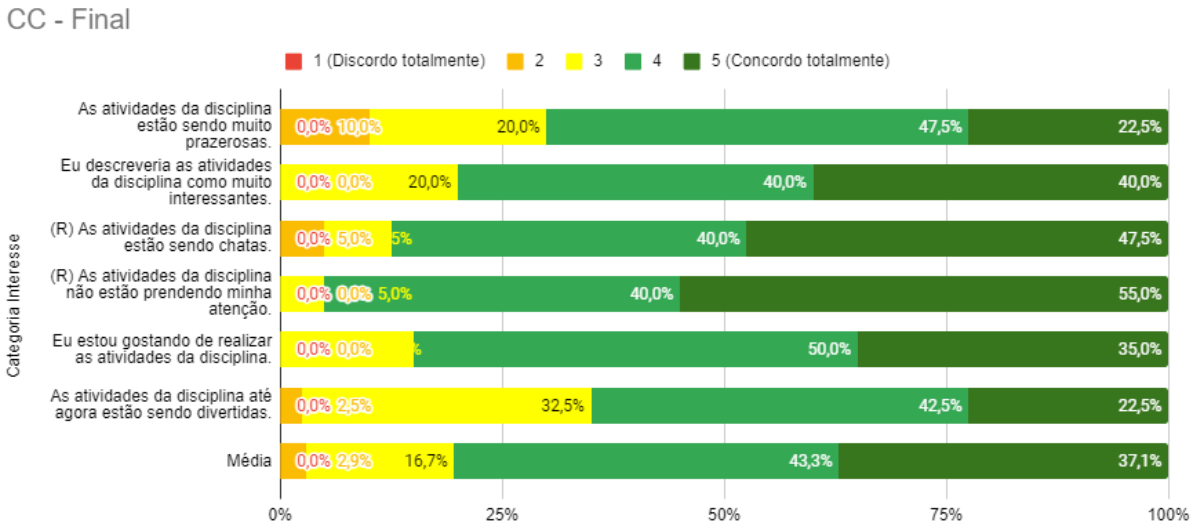


Fonte: Elaborada pelo autor.

5.2.2 Aplicação final do questionário

No decorrer do semestre, conforme indicado no gráfico da Figura 35, os alunos apresentaram uma queda nos índices de interesse, com uma porcentagem de concordância acima de 65% em todas as perguntas, e de 80,4% na média.

Figura 35 – Resultado IMI CC Final Categoria Interesse



Fonte: Elaborada pelo autor.

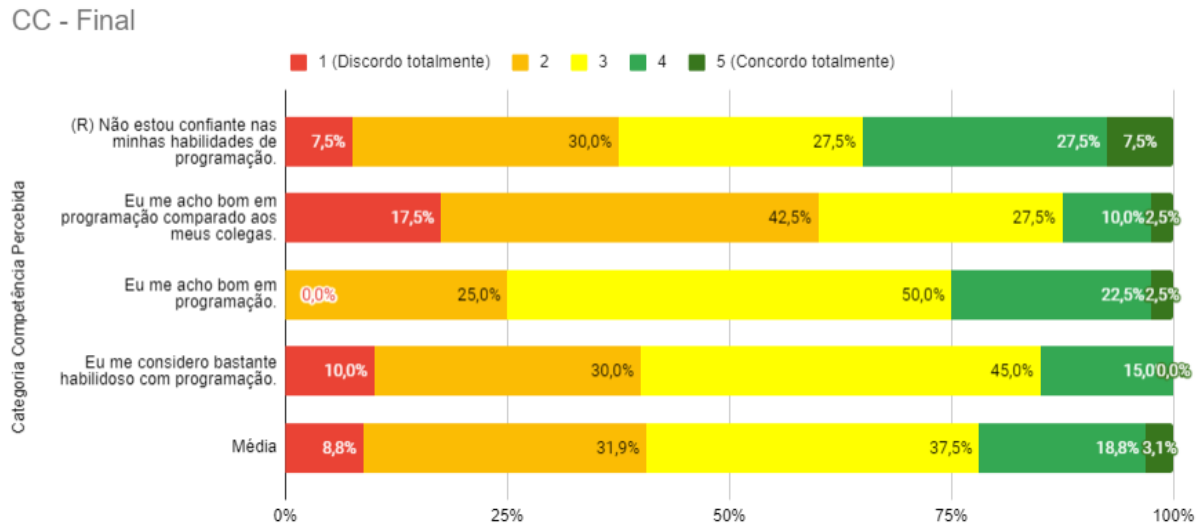
Na categoria Competência Percebida, conforme indicado no gráfico da Figura 36, percebe-se que os alunos continuam não se mostrando seguros com suas habilidades com programação, mas com uma sutil melhora na média. Em todas as perguntas, mais de 65% dos alunos responderam discordando ou de forma neutra, e 78% na média.

Na categoria de Esforço/Importância, conforme indicado no gráfico da Figura 37, mais de 87% dos alunos comunicaram a concordância da intenção de se esforçar ao máximo na disciplina em todas as perguntas, demonstrando uma pequena queda, com 93,4% de concordância na média, também demonstrando uma pequena queda.

Na categoria de Pressão/Tensão, conforme indicado no gráfico da Figura 38, 60% dos alunos comunicaram que não estavam nervosos ou estavam se sentindo de forma neutra para as avaliações da disciplina em todas as perguntas, e 60% na média.

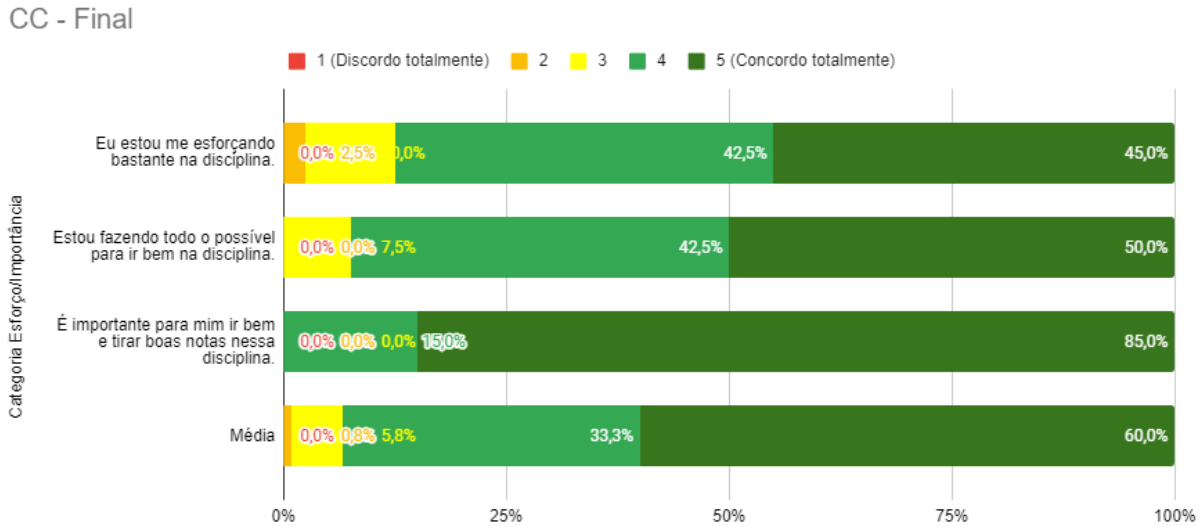
Na categoria de Valor/Utilidade, conforme indicado no gráfico da Figura 39, mais de 97% dos alunos concordaram com a utilidade do aprendizado de programação em todas as perguntas, e mais de 70% concordaram com a utilidade do CodeCombat em todas as perguntas. Na média, 78% dos alunos demonstraram concordância do valor e da utilidade do CodeCombat,

Figura 36 – Resultado IMI CC Final Categoria Competência



Fonte: Elaborada pelo autor.

Figura 37 – Resultado IMI CC Final Categoria Esforço



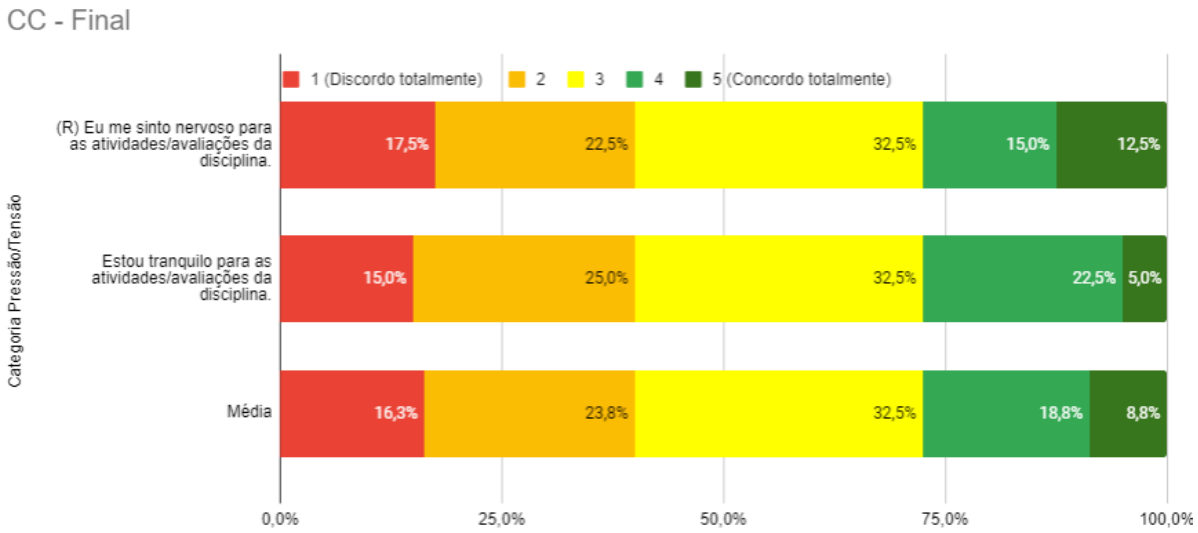
Fonte: Elaborada pelo autor.

e 98% demonstraram concordância da importância da programação. Algo para se notar, é que alguns alunos marcaram que, para eles, o CodeCombat não havia valor, algo que não foi colocado no questionário inicial.

5.2.3 Discussão

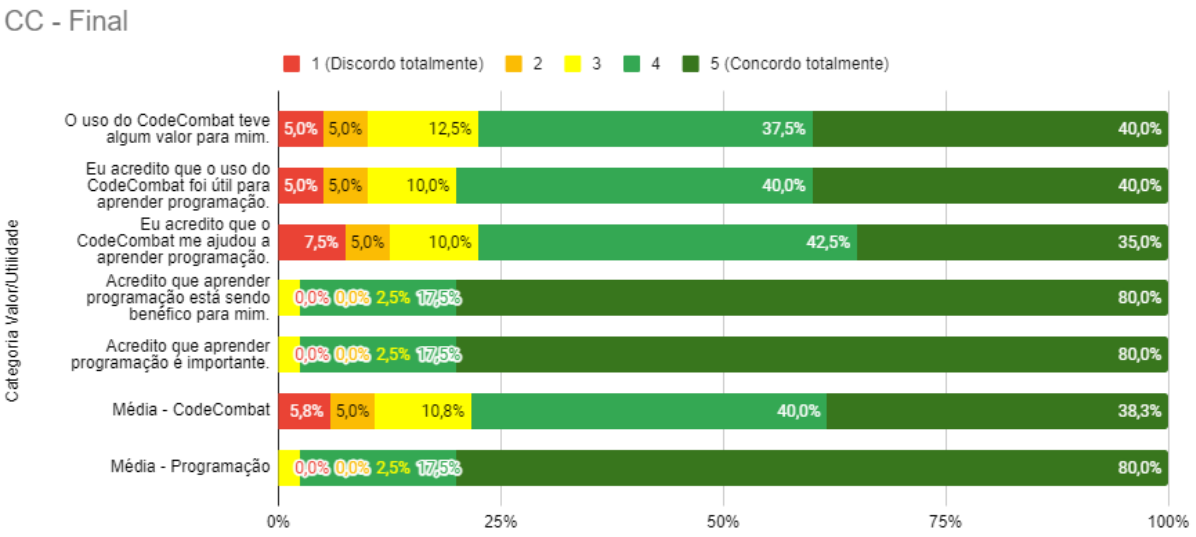
Em todas as categorias, com exceção da Competência Percebida, houve uma queda nos resultados dos questionários, na comparação do inicial com o final, indicando que a utilização do CodeCombat da forma que foi aplicado não gerou um grande impacto.

Figura 38 – Resultado IMI CC Final Categoria Pressão



Fonte: Elaborada pelo autor.

Figura 39 – Resultado IMI CC Final Categoria Valor



Fonte: Elaborada pelo autor.

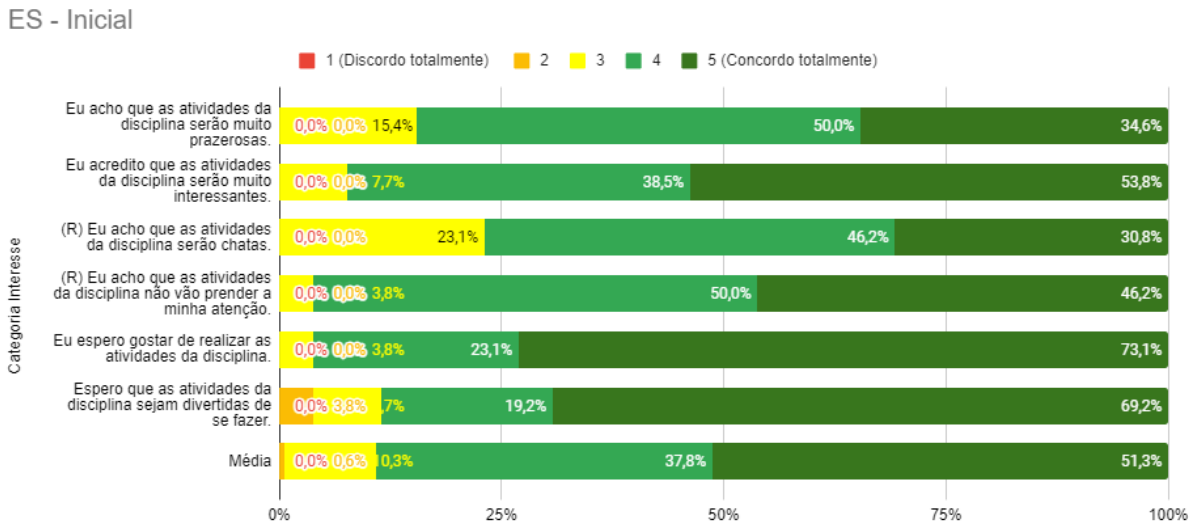
5.3 Resultado IMI Engenharia de Software

5.3.1 Aplicação inicial do questionário

No início do semestre, conforme indicado no gráfico da Figura 40, os alunos, de forma similar à turma de Ciência da Computação, indicaram bons índices de interesse, com uma porcentagem de concordância acima de 75% em todas as perguntas, e de 90% na média.

Na categoria Competência Percebida, conforme indicado no gráfico da Figura 41, percebe-se que os alunos não se mostraram seguros com suas habilidades com programação. Assim como na turma de Ciência da Computação, os alunos foram instruídos a marcar 1 caso

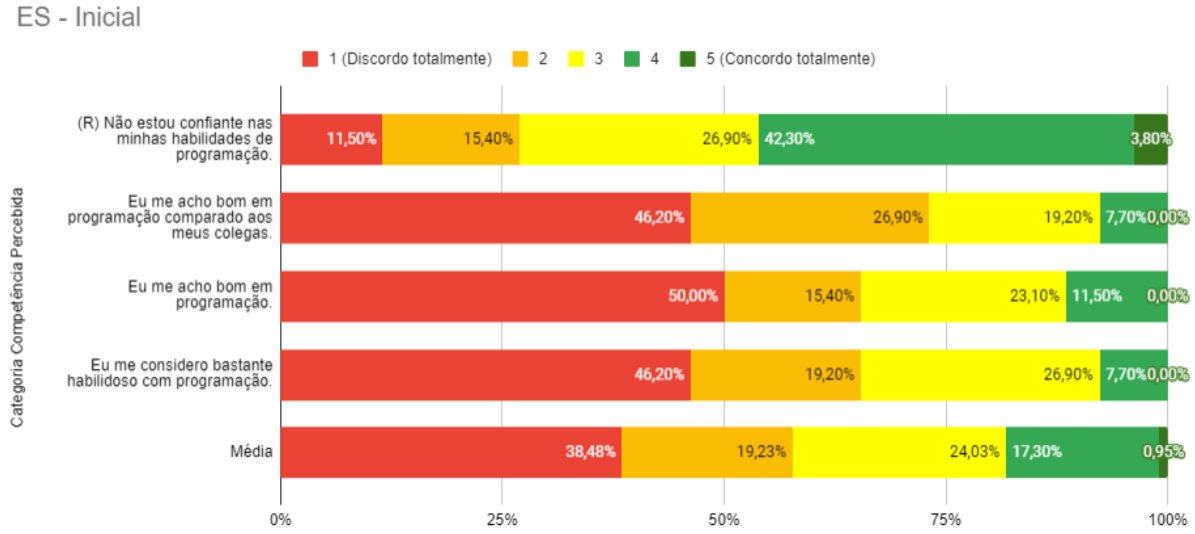
Figura 40 – Resultado IMI ES Inicial Categoria Interesse



Fonte: Elaborada pelo autor.

não tivessem experiência prévia com programação. Em todas as perguntas mais de 50% dos alunos responderam discordando ou de forma neutra, e 81% na média.

Figura 41 – Resultado IMI ES Inicial Categoria Competência

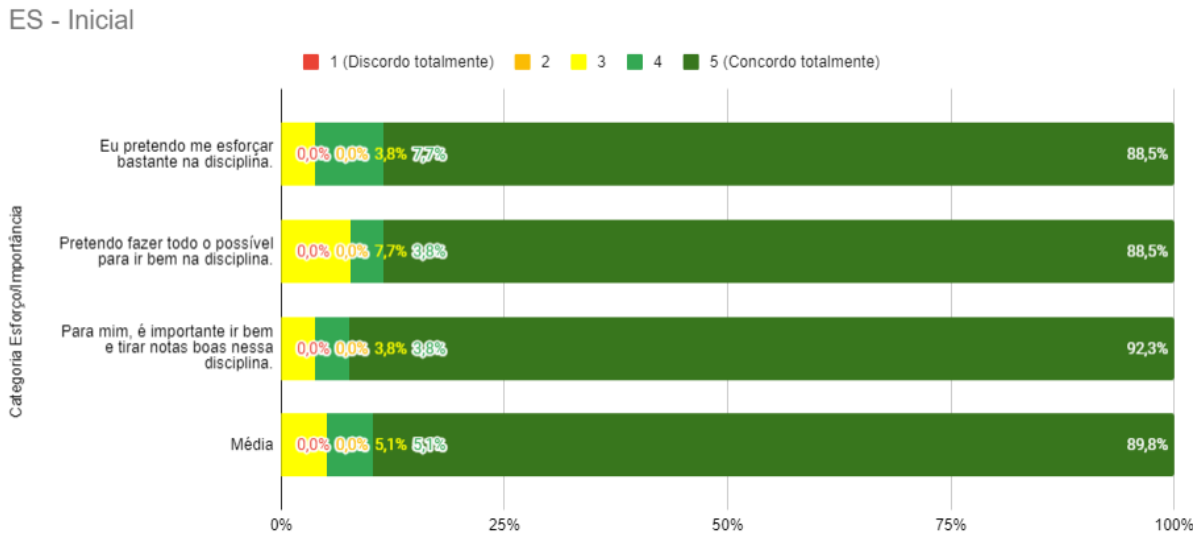


Fonte: Elaborada pelo autor.

Na categoria de Esforço/Importância, conforme indicado no gráfico da Figura 42, mais de 90% dos alunos comunicaram a concordância da intenção de se esforçar o máximo na disciplina em todas as perguntas, com 89% de concordância na média.

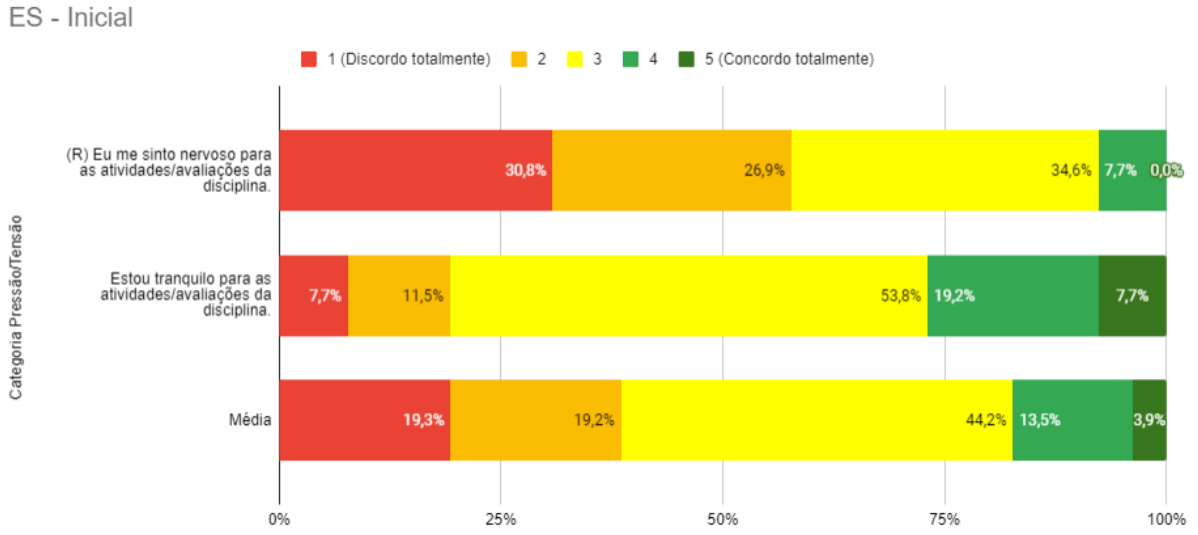
Na categoria de Pressão/Tensão, conforme indicado no gráfico da Figura 43, 42% dos alunos comunicaram que não estavam nervosos ou estavam se sentindo de forma neutra para as avaliações da disciplina em todas as perguntas, e 60% na média.

Figura 42 – Resultado IMI ES Inicial Categoria Esforço



Fonte: Elaborada pelo autor.

Figura 43 – Resultado IMI ES Inicial Categoria Pressão



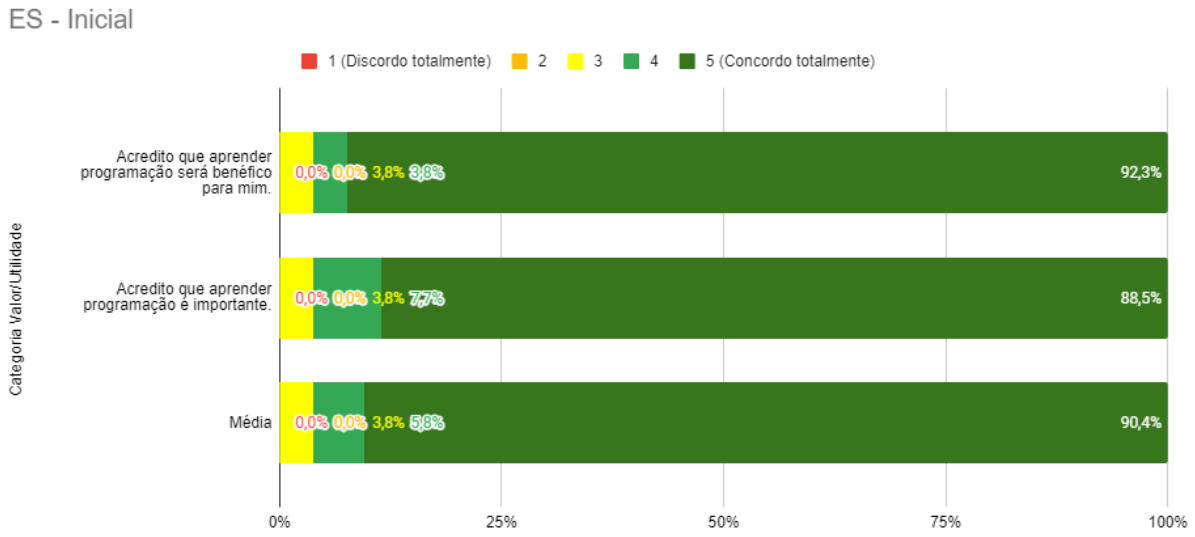
Fonte: Elaborada pelo autor.

Na categoria de Valor/Utilidade (Figura 44), por conta da turma não ter utilizado o CodeCombat, somente foram feitas perguntas relacionadas a programação. Nela, 96% dos alunos responderam concordando com o valor e utilidade da programação em todas as perguntas, e 96% dos alunos responderam com concordância na média.

5.3.2 Aplicação final do questionário

No decorrer do semestre, de forma similar à turma de Ciência da Computação, conforme indicado no gráfico da Figura 45, os alunos apresentaram uma queda nos índices de

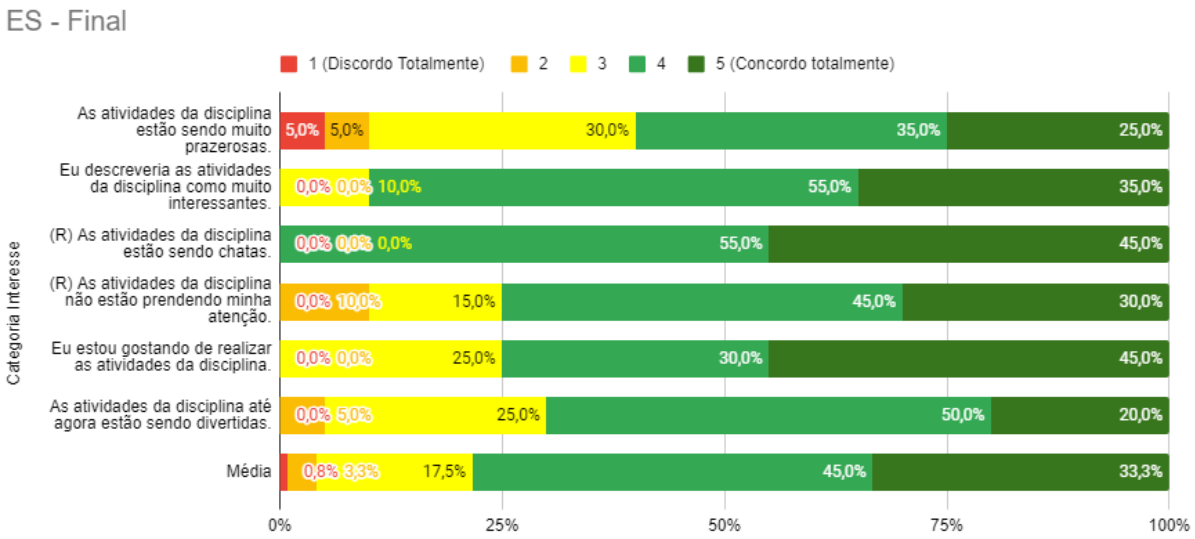
Figura 44 – Resultado IMI ES Inicial Categoria Valor



Fonte: Elaborada pelo autor.

interesse, com uma porcentagem de concordância acima de 60% em todas as perguntas, e de 78,3% na média.

Figura 45 – Resultado IMI ES Final Categoria Interesse

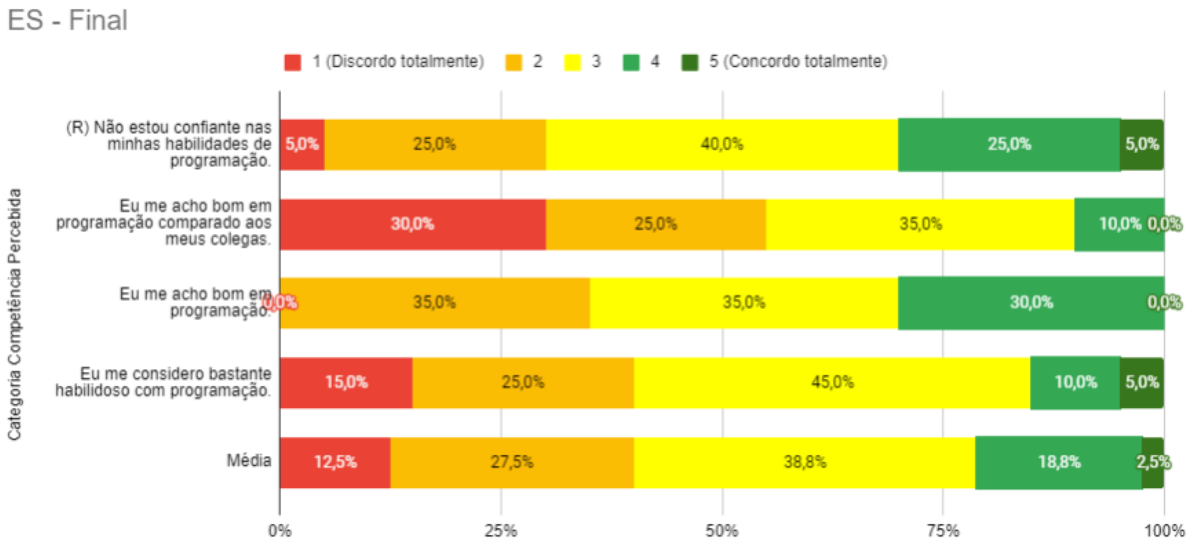


Fonte: Elaborada pelo autor.

Na categoria Competência Percebida, conforme indicado no gráfico da Figura 46, de forma similar à turma de CC, percebe-se que os alunos continuam não se mostrando seguros com suas habilidades com programação, mas com uma sutil melhora na média, onde houve uma diminuição dos alunos que selecionaram "discordam totalmente"e um aumento nas posteriores.

Na categoria de Esforço/Importância, conforme indicado no gráfico da Figura 47, mais de 85% dos alunos comunicaram a concordância da intenção de se esforçar ao máximo na

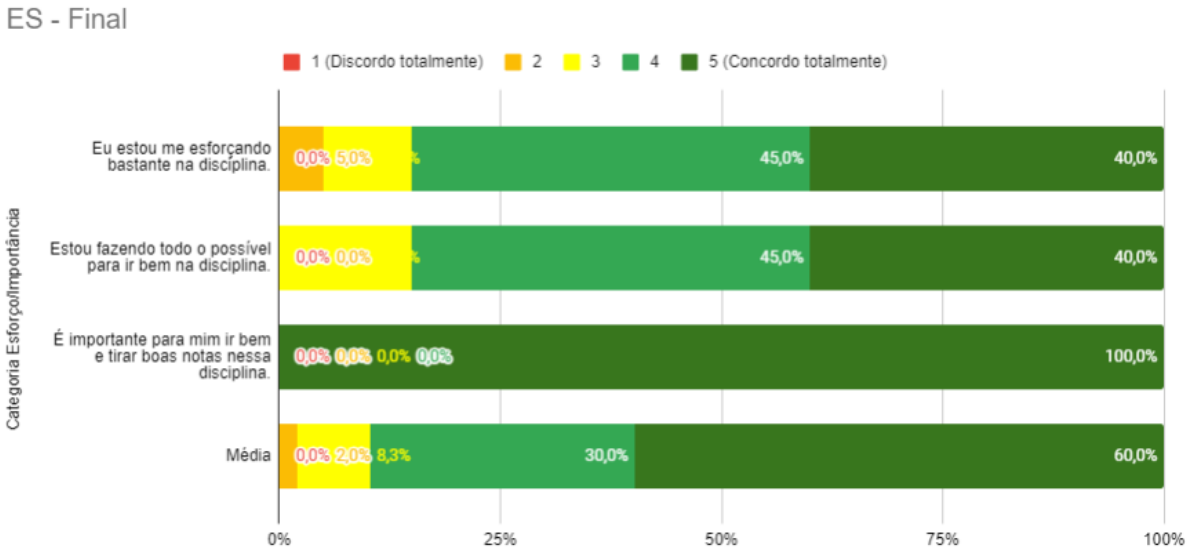
Figura 46 – Resultado IMI ES Final Categoria Competência



Fonte: Elaborada pelo autor.

disciplina em todas as perguntas, demonstrando uma pequena queda, com 90% de concordância na média, demonstrando uma sutil subida.

Figura 47 – Resultado IMI ES Final Categoria Esforço

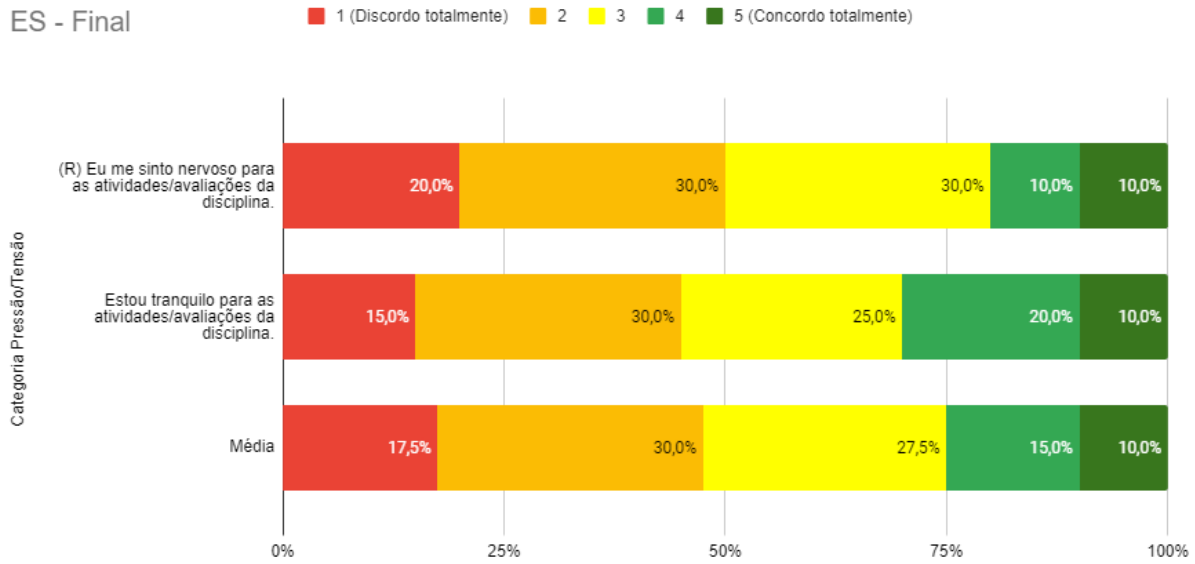


Fonte: Elaborada pelo autor.

Na categoria de Pressão/Tensão, conforme indicado no gráfico da Figura 48, 50% dos alunos comunicaram que não estavam nervosos ou estavam se sentindo de forma neutra para as avaliações da disciplina em todas as perguntas, e 52% na média.

Na categoria de Valor/Utilidade (Figura 49, por conta da turma não ter utilizado o CodeCombat, novamente, somente foram feitas perguntas relacionadas à programação. Nela,

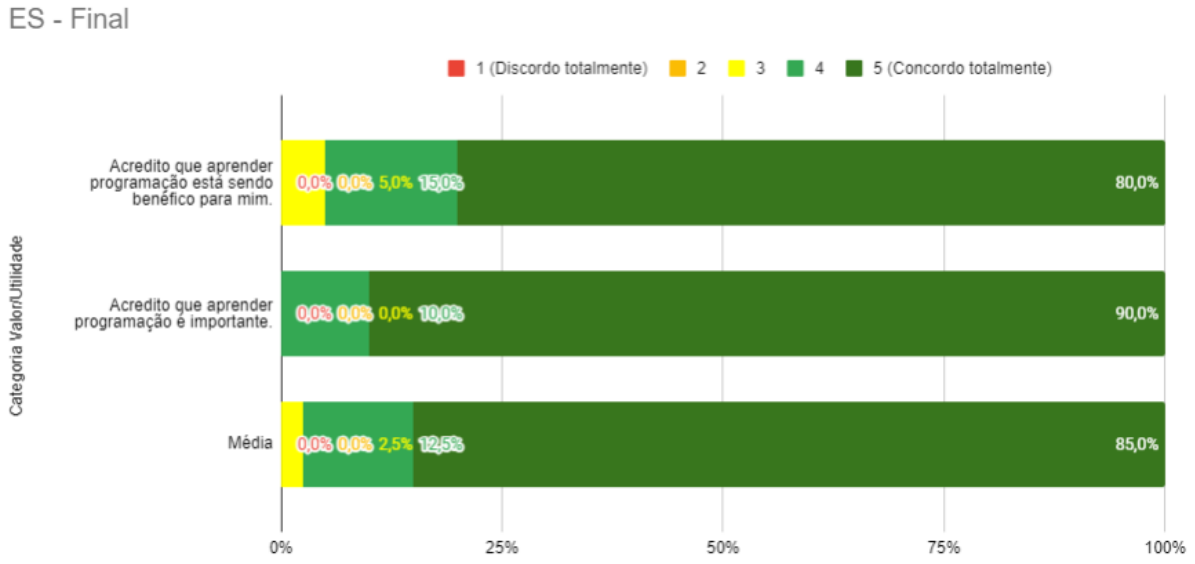
Figura 48 – Resultado IMI ES Final Categoria Pressão



Fonte: Elaborada pelo autor.

95% dos alunos responderam concordando com o valor e utilidade da programação em todas as perguntas, e 98% dos alunos responderam com concordância na média.

Figura 49 – Resultado IMI ES Final Categoria Valor



Fonte: Elaborada pelo autor.

5.3.3 Discussão

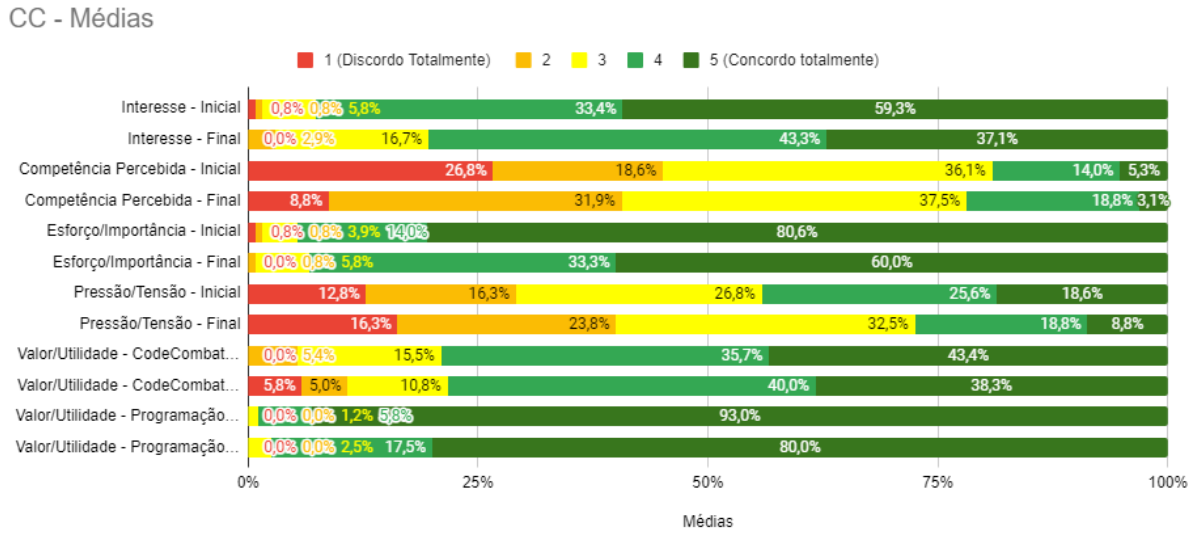
De forma similar ao que aconteceu na turma de Ciência da Computação, foi percebido que, em todas categorias exceto as de Competência Percebida e de Esforço/Importância, houve

uma queda no resultados dos questionários, na comparação do inicial com o final.

5.4 Comparação das turmas

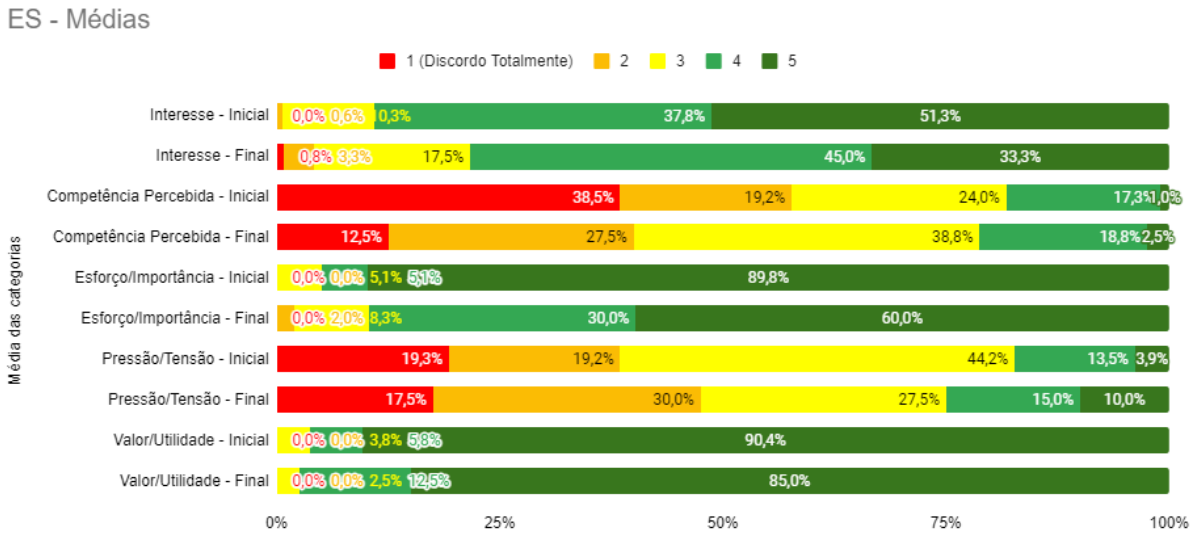
Para realizar as comparações entre as turmas, foram organizadas, nos gráficos das Figuras 50 e 51, as médias das categorias das respostas dos questionários de cada turma, nas aplicações inicial e final.

Figura 50 – Médias das categorias iniciais da turma de CC



Fonte: Elaborada pelo autor.

Figura 51 – Médias das categorias da turma de ES



Fonte: Elaborada pelo autor.

No geral, comparando os gráficos das duas turmas, ficou perceptível que os resultados

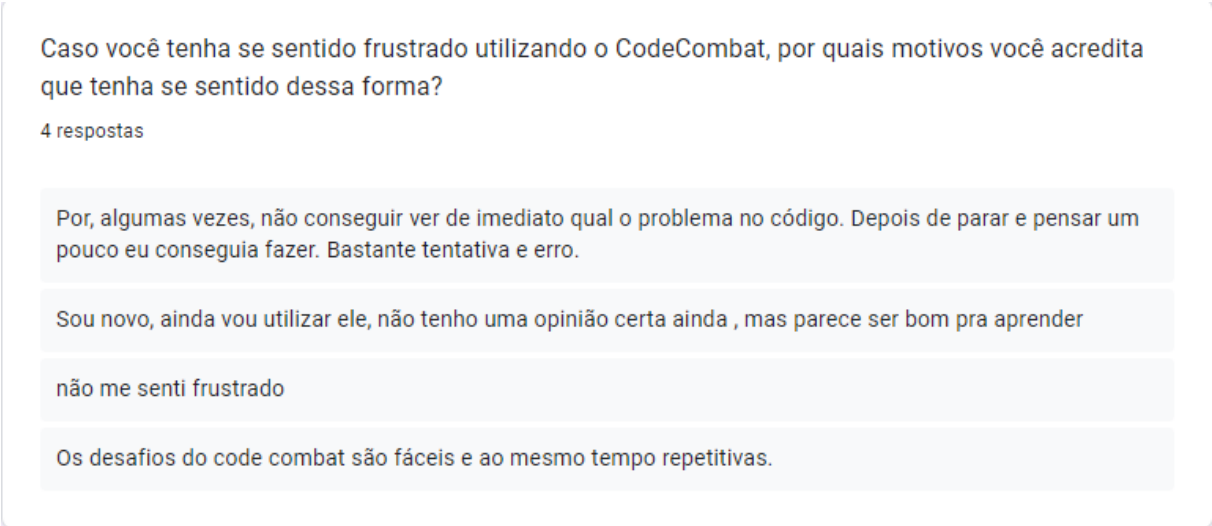
dos questionários foram bastante semelhantes, havendo quedas e melhoras nos índices de motivação nas mesmas categorias, sugerindo que a aplicação do CodeCombat não trouxe um grande impacto na motivação dos alunos. É importante lembrar que foi aplicado somente o primeiro módulo do CodeCombat, gerando uma limitação no experimento.

5.5 Análise qualitativa dos dados

5.5.1 Respostas das perguntas subjetivas

Nas Figuras 52 e 53, estão as respostas das perguntas subjetivas do questionário.

Figura 52 – Respostas da primeira pergunta subjetiva



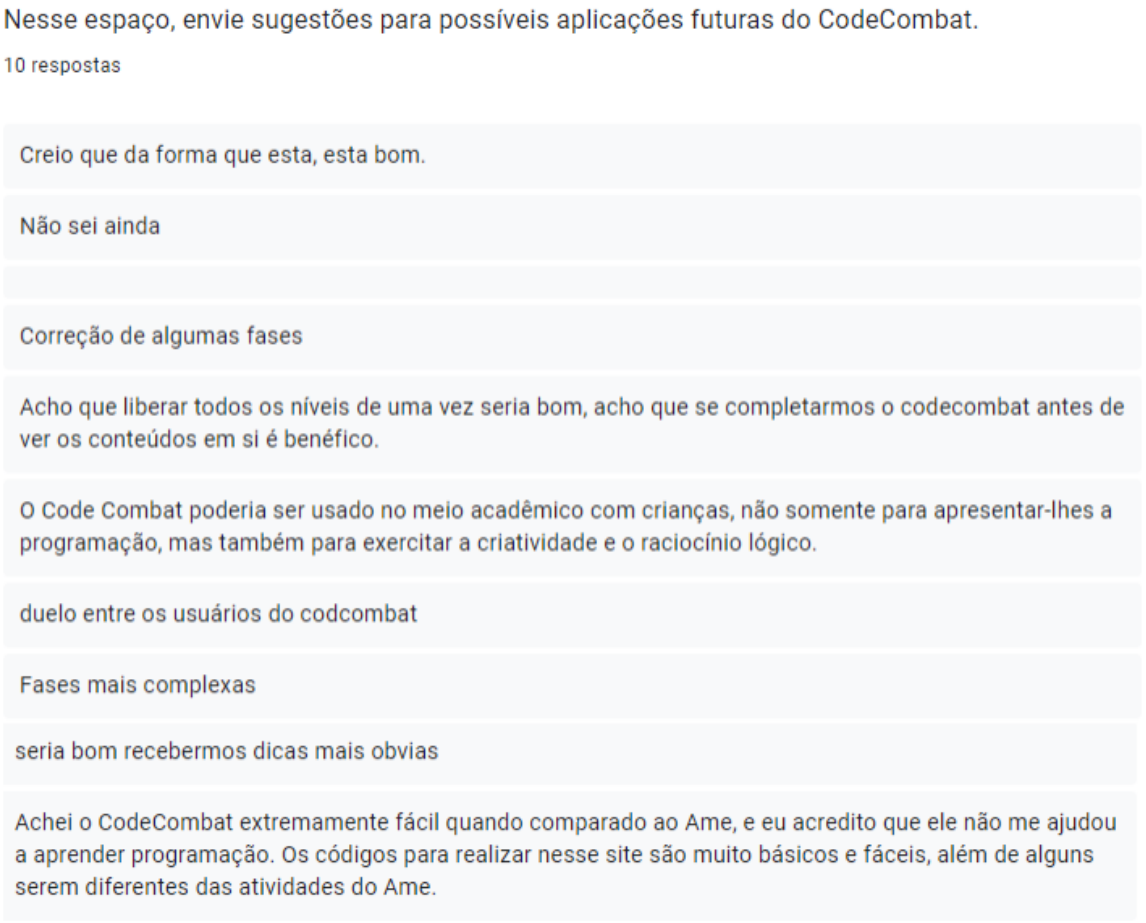
Fonte: Elaborada pelo autor.

5.5.2 Discussão

Nas perguntas subjetivas, no geral, os alunos apontaram sobre as diferenças de nivelamento entre os níveis do CodeCombat e as atividades de programação aplicadas na disciplina. Os alunos consideraram que a dificuldade dos níveis do CodeCombat estava muito abaixo das atividades praticadas na disciplina. Os alunos também comentaram sobre a forma que os níveis foram liberados, pois demonstraram o sentimento que eles deviam ser disponibilizados mais rapidamente.

É importante enfatizar que os níveis do CodeCombat foram utilizados para apresentar os comandos da linguagem, ou seja, para realizar uma introdução à linguagem. Já as atividades da disciplina possuem o objetivo de treinar a capacidade dos alunos de resolver problemas. A

Figura 53 – Respostas da segunda pergunta subjetiva



Fonte: Elaborada pelo autor.

liberação das fases foi feita, conforme o planejamento do experimento, para que acompanhasse o conteúdo da disciplina.

5.6 Considerações finais

Pode-se concluir, levando em consideração as limitações das fases gratuitas do CodeCombat e as respostas das perguntas objetivas e os feedbacks das perguntas subjetivas, que não houve uma diferença significativa da turma que usou o CodeCombat quando comparada com os alunos da turma que não utilizou a ferramenta.

6 CONCLUSÃO

6.1 Principais contribuições

A gamificação tem como objetivo tornar as atividades em algo mais lúdico, para aumentar a motivação dos alunos. Foi feita uma análise de possíveis ferramentas para serem aplicadas, e foi escolhido o CodeCombat. Após isso, foi elaborada a metodologia do experimento, alinhando a ferramenta com o conteúdo da disciplina de Fundamentos da Programação da Universidade Federal do Ceará Campus de Russas. Então, foram separados os grupos de experimento e controle, e foi aplicado o questionário IMI para medir os níveis de motivação dos dois grupos, no começo e no fim do experimento, com os dois grupos apresentando resultados similares.

Foi concluído, com os resultados do experimento, que o uso do CodeCombat não gerou um grande impacto na motivação dos alunos. A justificativa para isso envolve alguns fatores, como o bloqueio dos módulos avançados do CodeCombat para usuários gratuitos, o nivelamento da dificuldade da ferramenta com o conhecimento dos alunos e a metodologia de aplicação do experimento.

6.2 Trabalhos futuros

Para trabalhos futuros, podem ser feitas novas aplicações do CodeCombat com mudanças na metodologia. Devido à limitação dos módulos pagos do CodeCombat, a ferramenta pode ser utilizada em um tempo menor e com uma dedicação mais exclusiva, como uma introdução para os alunos.

Também podem ser analisadas e testadas outras ferramentas, como a Ozaria, dos mesmos desenvolvedores do CodeCombat, mas que traz algumas diferenças para o CodeCombat.

Em próximos experimentos, podem ser aprimoradas as metodologias utilizadas, aplicando mais questionários e também avaliando o desempenho dos alunos que utilizam a ferramenta, não somente a motivação. Também pode ser realizada uma separação mais rigorosa entre o grupo experimental e controle, selecionando os grupos de uma forma aleatória e que sejam da mesma turma.

Podem, ainda, ser reconsideradas algumas ferramentas não avaliadas neste estudo, como o Scratch e o ClassCraft. Por último, outra possibilidade seria considerar o desenvolvimento

de uma nova ferramenta, levando em consideração as falhas apontadas na pesquisa, para melhor suprí-las.

REFERÊNCIAS

BARR, M.; KALLIA, M. Why students drop computing science: Using models of motivation to understand. Koli Calling '22: Proceedings of the 22nd Koli Calling International Conference on Computing Education Research, Conferências Ibero-Americanas WWW/Internet e Computação Aplicada 2017, p. 1–6, 2022.

BARTLE, R. Hearts, clubs, diamonds, spades: Players who suit muds. Journal of MUD, v. 1, p. 19, 1996.

CARVALHO, L. S. G.; GADELHA, B. F.; NAKAMURA, F. G.; OLIVEIRA, D. B. F.; OLIVEIRA, E. H. T. Ensino de programação para futuros não-programadores: Contextualizando os exercícios com as demais disciplinas de mesmo período letivo. 2016: ANAIS DO XXIV WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, p. 2116–2125, 2016.

CUERVO-CELY, K. D.; RESTREPO-CALLE, F.; RAMíREZ-ECHEVERRY, J. J. Effect of gamification on the motivation of computer programming students. Journal of Information Technology Education:Research, Journal of Information Technology Education, v. 20, p. 01–23, 2022.

ECCLES, J. Expectancies, values and academic behaviors. J. T. Spence (Ed.), Achievement and achievement motives: Psychological and sociological approaches, p. 75–146, 1983.

GOMES, H.; MELO, J.; FARRAPO, H.; BONATES, M.; COUTINHO, E. Descrição e comparação de jogos digitais para auxiliar no ensino de programação. Revista Sistemas e Mídias Digitais (RSMD), v. 3, 04 2018.

HOED, R. M.; LADEIRA, M.; LEITE, L. L. O que leva os alunos dos cursos superiores de computação a evadirem? um estudo de caso feito na universidade de brasília. Conferências Ibero-Americanas WWW/Internet e Computação Aplicada 2017, p. 159–166, 2017.

KINNUNEN, P.; MALMI, L. Why students drop out cs1 course? In: International Computing Education Research Workshop. Proceedings of the second international workshop on Computing education research, 9-10.9.2006, Canterbury, United Kingdom. ICER 2006. United States: ACM, 2006. p. 97. ISBN 1-59593-494-4.

LISTON M., F. D.; PATTERSON V., . A study of progression in irish higher education. Higher Education Authority, Dublin., 2017.

MARSH, H. W. Verbal and math self-concepts: An internal/external frame of reference model. American Educational Research Journal, v. 23, n. 1, p. 129–149, 1986.

(MEC)., M. da E. Diplomação, retenção e evasão nos cursos de graduação em instituições de ensino superior públicas. Brasília - DF: Comissão Especial de Estudos Sobre a Evasão nas Universidades Públicas Brasileiras., p. 20, 1997.

MOREIRA, S. M. C. Uma experiência de gamificação no contexto do ensino remoto: análise da motivação e experiência dos jogadores. 2021. 82f. Dissertação (Bacharelado em Engenharia de Software) – Universidade Federal do Ceará, Campus de Russas, Russas-CE, 2021.

MURATET, M.; TORGUET, P.; JESSEL, J.-P.; VIALLET, F. Towards a serious game to help students learn computer programming. International Journal of Computer Games Technology, v. 2009, 01 2009.

PETERSEN, A.; CRAIG, M.; CAMPBELL, J.; TAFLIOVICH, A. Revisiting why students drop cs1. In: Proceedings of the 16th Koli Calling International Conference on Computing Education Research. New York, NY, USA: Association for Computing Machinery, 2016. (Koli Calling '16), p. 71–80. ISBN 9781450347709. Disponível em: https://doi.org/10.1145/2999541.2999552.

ANEXO A – PLANO DE ENSINO DA DISCIPLINA DE FUNDAMENTOS DA PROGRAMAÇÃO - CIÊNCIA DA COMPUTAÇÃO

UFC

CAMPUS DE RUSSAS

PLANO DE ENSINO DE DISCIPLINA

Ano/Semestre: 2023/1

1. Identificação				
1.1. Unidade: Campus Russas				
1.2. Curso: Ciência da Computação				
1.3. Estrutura Curricular (ano-período): 2018.2				
1.4. Nome da Disciplina: Fundamentos de Programação				
1.5. Código da Disciplina: RUS0297				
1.6. Caráter da Disciplina: (X)Obrigatória ( )Optativa CONFORME FORMULÁRIO DE CRIAÇÃO				
1.7. Regime de Oferta da Disciplina: ( )Semestral ( )Anual ( ) Modular				
1.8. Carga Horária (CH) Total: 64 h	C.H. Teórica: 32 h	C.H. Prática: 32 h	C.H. EaD: 0 h	C.H. Extensão: 0 h
1.9. Pré-requisitos (quando houver): Não há.				
1.10. Co-requisitos (quando houver): Não há.				
1.11. Equivalências (quando houver): RUS0001 – Fundamentos de Programação				
1.12. Professor(es): Markos Oliveira Freitas				

2. Justificativa

A disciplina de Fundamentos de Programação propicia a fundamentação básica na arte de programar, envolvendo conceitos de algoritmos, soluções de problemas através de computadores, bem como de linguagens de programação estruturada. O aprendizado nas disciplinas que envolvem programação é facilitado na medida em que os alunos adquirem uma base sólida nesta disciplina.

3. Ementa

Algoritmos, Conceitos Fundamentais de Programação, Expressões, Controles de Fluxo, Funções e Procedimentos, Vetores e Matrizes, Cadeias de Caracteres, Tipos Estruturados e Arquivos.

4. Objetivos – Geral e Específicos

Objetivo Geral

- Ensinar os alunos os conceitos fundamentais de algoritmos e programação.

Objetivos Específicos

O aluno, ao final do semestre, deverá ser capaz de:

- Desenvolver a habilidade dos alunos em programação estruturada, enfocando na construção de programas corretos, confiáveis, seguros, eficientes;
- Compreender e utilizar as estruturas de dados básicas, como vetores e matrizes;
- Desenvolver a habilidade de solucionar problemas através do computador.

5. Calendário de Atividades

Data	Descrição do Conteúdo	Carga Horária
14/03	Sem aula – recepção de ingressantes	
	Introdução ***Objetivo principal:*** ***- Descrever o funcionamento da disciplina***	**4 hrs**
16/03	Introdução [sala] *Objetivos específicos:* *- Contextualizar a disciplina e o curso*	
21/03	Introdução à ementa [sala] *Objetivos específicos:* *- Definir a ementa* *- Descrever as formas de avaliação* *- Descrever as atividades discentes* *- Descrever a importância da disciplina* *- Relacionar o conteúdo da disciplina com outras disciplinas* *- Contextualizar o conteúdo da disciplina com o curso e com a profissão* *- Definir algoritmo* *- Definir pensamento algorítmico* *- Definir linguagem de programação* *- Comparar linguagem de programação com linguagem binária e linguagem natural*	
	Conceitos básicos de programação ***Objetivo principal:*** ***- Implementar programas com expressões aritméticas***	**8 hrs**
23/03	Primeiro programa [lab] *Objetivos específicos:* *- Utilizar o ambiente de desenvolvimento* *- Ler código-fonte com comandos de saída de dados* *- Escrever código-fonte com comandos de saída de dados* *- Ler código-fonte com comandos de entrada de dados* *- Escrever código-fonte com comandos de entrada de dados* *- Ler código-fonte com comentários* *- Escrever código-fonte com comentários* *- Identificar erros em código-fonte*	
28/03	Variáveis e expressões aritméticas [sala] *Objetivos específicos:* *- Ler código-fonte com variáveis* *- Escrever código-fonte com variáveis* *- Definir tipo, nome e valor de variáveis* *- Descrever tipos de variáveis* *- Descrever regras para nomes de variáveis* *- Escrever código-fonte com comandos de entrada e saída de variáveis de tipos diferentes*	
30/03	Variáveis e expressões aritméticas [lab] CodeCombat: 1, 2, 3, 3a, 3b, e Desafio Passos Cuidadosos	
04/04	*Aula de tira-dúvidas* [sala] *Objetivos específicos:* *- Consolidar conhecimentos*	
06/04	Sem aula – feriado de Semana Santa	
	Funções e comandos de repetição ***Objetivo principal:*** ***- Implementar programas com funções e comandos de repetição***	**8 hrs**
11/04	Funções [sala] *Objetivos específicos:* *- Definir funções* *- Comparar funções com programas: entrada → execução → saída* *- Descrever parâmetros e retornos* *- Definir assinatura de funções* *- Definir escopo de variáveis* *- Ler código-fonte com funções* *- Escrever código-fonte com funções*	
13/04	Funções [lab] CodeCombat: 4, 4a, 4b, Desafio Passos Longos	

	CodeCombat: 5, 5a, 5b, Desafio Passos Perigosos, Desafio Combo Hora de Dormir CodeCombat: 6, 7, 8	
18/04	Sem aula – SESCOMP	2 hrs
20/04	Sem aula – SESCOMP	2 hrs
25/04	Comandos de repetição – enquanto e para [sala] *Objetivos específicos:* *- Definir comando de repetição* *- Descrever o comando enquanto* *- Definir comando de repetição aninhado* *- Definir contador* *- Definir acumulador* *- Ler código-fonte com comandos de repetição* *- Escrever código-fonte com comandos de repetição* *- Definir comando de repetição* *- Descrever o comando para* *- Ler código-fonte com comandos de repetição* *- Escrever código-fonte com comandos de repetição*	
27/04	Comandos de repetição – enquanto e para [sala] CodeCombat: 9, 10, 11, 11a, 11b, Desafio Loop no Armazém, 12 CodeCombat: 13, 14, 14a, 14b CodeCombat: 15, 16, 16a, 16b CodeCombat: Desafio Mestre dos Nomes de Depurar CodeCombat: 17, Desafio Combo Lugar Seguro CodeCombat: 18	
	Comandos de decisão e funções recursivas ***Objetivo principal:*** ***- Implementar programas com comandos de decisão***	**12 hrs**
02/05	Comandos de decisão e quebras em comandos de repetição [sala] *Objetivos específicos:* *- Definir comando de decisão* *- Descrever teste de decisão* *- Descrever o comando se … então* *- Descrever o comando se … então … senão* *- Descrever o comando de escolha* *- Definir comando de decisão aninhado* *- Ler código-fonte com comandos de decisão* *- Escrever código-fonte com comandos de decisão* *- Definir quebras em comando de decisão* *- Descrever o comando de interrupção de repetição* *- Descrever o comando de continuação de laço* *- Ler código-fonte com quebras em comandos de repetição* *- Escrever código-fonte com quebras em comandos de repetição*	
04/05	Comandos de decisão e quebras em comandos de repetição [lab]	
09/05	Expressões booleanas [sala] *- Definir expressões relacionais* *- Definir operadores relacionais: maior, maior ou igual, menor, menor ou igual, igual e diferente* *- Definir expressões booleanas* *- Definir operadores booleanos: conjunção, disjunção, negação* *- Definir tabela-verdade* *- Aplicar a tabela-verdade a uma expressão booleana* *- Definir as equivalências de DeMorgan* *- Ler código-fonte com expressões booleanas* *- Escrever código-fonte com expressões booleanas*	
11/05	Expressões booleanas [lab]	
16/05	Funções recursivas [sala] *Objetivos específicos:* *- Definir funções recursivas* *- Comparar funções recursivas com funções não-recursivas* *- Ler código-fonte com funções recursivas* *- Escrever código-fonte com funções recursivas*	
18/05	Funções recursivas [lab]	
	Estrutura de dados homogêneas e heterogêneas ***Objetivo principal:***	**18 hrs**

	- Implementar programas com comandos de repetição aplicados a estruturas homogêneas, como vetores e matrizes, e a estruturas heterogêneas como listas, conjuntos e dicionários	
23/05	Vetores e listas [sala] *Objetivos específicos:* *- Definir vetores em programação* *- Descrever acesso a elementos de vetores* *- Relacionar comandos de repetição a vetores* *- Descrever principais operações em vetores* *- Ler código-fonte com vetores* *- Escrever código-fonte com vetores* *- Definir listas em programação* *- Descrever acesso a elementos de listas* *- Relacionar comandos de repetição a listas* *- Descrever principais operações em listas* *- Ler código-fonte com listas* *- Escrever código-fonte com listas*	
25/05	Vetores e listas [lab]	
30/05	Matrizes e listas de listas [sala] *Objetivos específicos:* *- Definir matrizes em programação* *- Descrever acesso a elementos de matrizes* *- Relacionar comandos de repetição a matrizes* *- Descrever principais operações em matrizes* *- Ler código-fonte com matrizes* *- Escrever código-fonte com matrizes*	
01/06	Matrizes e listas de listas [lab]	
06/06	*Aula de tira-dúvidas* [sala] *Objetivos específicos:* - Consolidar conhecimentos	
08/06	Sem aula – feriado de Corpus Christi	
13/06	Conjuntos e dicionários [sala] *Objetivos específicos:* *- Definir conjuntos em programação* *- Descrever acesso a elementos de conjuntos* *- Relacionar comandos de repetição a conjuntos* *- Descrever principais operações em conjuntos* *- Ler código-fonte com conjuntos* *- Escrever código-fonte com conjuntos* *- Definir dicionários em programação* *- Descrever acesso a elementos de dicionários* *- Relacionar comandos de repetição a dicionários* *- Descrever principais operações em dicionários* *- Ler código-fonte com dicionários* *- Escrever código-fonte com dicionários*	
15/06	Conjuntos e dicionários [lab]	
20/06	Dicionários [sala]	
22/06	Dicionários [lab]	
	Arquivos ***Objetivo principal:*** ***- Implementar programas com arquivos***	**10 hrs**
27/06	Arquivos [sala] *Objetivos específicos:* *- Definir arquivos em programação* *- Descrever acesso de leitura e escrita de arquivos* *- Ler código-fonte com arquivos* *- Escrever código-fonte com arquivos*	
29/06	Arquivos [lab]	
04/07	*Aula de prática / tira-dúvidas* [sala] *Objetivos específicos:* - Consolidar conhecimentos	
06/07	*Aula de prática / tira-dúvidas* [lab] *Objetivos específicos:* - Consolidar conhecimentos	

11/07	*Aula de prática / tira-dúvidas* sala *Objetivos específicos:* - Consolidar conhecimentos	
13/07	*Aula extra para ajuste de datas*	
18/07	Sem aula – período de avaliações finais	
20/07	**Avaliação Final** sala	

Legenda

sala	Aula em sala de aula
lab	Aula em laboratório

6. Metodologia de Ensino

A disciplina é dividida em aulas teóricas e aulas práticas. As aulas teóricas acontecerão em sala de aula, enquanto as aulas práticas acontecerão em laboratório de informática.
Nas aulas teóricas, a metodologia de ensino será construtiva sempre que possível, focando o diálogo com o aluno. Quando não for possível, será expositiva.
Nas aulas práticas, o conteúdo visto nas aulas teóricas será aplicado por meio da resolução de listas de problemas, na linguagem de programação Python.
Na disciplina, será utilizado o Ambiente Multimeios de Ensino-Aprendizagem (AME), que pode ser acessado pelo site ame2.russas.ufc.br. Através do AME, serão disponibilizados os materiais da disciplina, bem como as listas de problemas e as atividades a serem entregues.
Além disso, a disciplina utilizará uma ferramenta chamada Code Combat (codecombat.com), para estimular o aprendizado de programação com gamificação.

7. Atividades Discentes

Os alunos deverão fazer as listas de problemas disponibilizadas pelo AME. No AME, alguns dos problemas poderão ser testados através de correção automática pelo sistema, antes de enviados ao professor.
Além das atividades do AME, os alunos deverão resolver os problemas apresentados pelo Code Combat. Em cada fase do Code Combat, o aluno deverá programar as ações do seu personagem, que deve fugir de uma masmorra.

8. Sistema de Avaliação

Conforme o Regimento Geral da UFC, a avaliação de rendimento do aluno far-se-á segundo os critérios de assiduidade e eficiência. Na verificação da assiduidade será aprovado o aluno que frequentar 75% (setenta e cinco por cento) ou mais da carga horária da disciplina, vedado o abono de faltas. A verificação da eficiência compreenderá, no mínimo, duas avaliações progressivas e uma avaliação final. Será aprovado por média o aluno que apresentar média aritmética das notas resultantes das avaliações progressivas igual ou superior a 07 (sete). O aluno que apresentar a média igual ou superior a 04 (quatro) e inferior a 07 (sete), será submetido à avaliação final. Nesse caso, o aluno será aprovado quando obtiver nota igual ou superior a 04 (quatro) na avaliação final e média final igual ou superior a 05 (cinco).

A frequência do aluno será baseada na sua presença em aula.
A nota do aluno depende das atividades entregues pelo AME e da resolução dos problemas do Code Combat.
Pelo AME, serão 10 atividades, na forma de lista de problemas. Cada atividade terá uma nota de 0 a 10 pontos, e a média aritmética dessas 10 atividades é a primeira nota (N1) da disciplina.
Pelo Code Combat, serão 7 atividades. Cada atividade vale 0 ou 10 pontos, e a média dessas atividades é a segunda nota da disciplina (N2).
A média parcial será calculada como uma média ponderada entre a nota N1 e a nota N2, onde a nota N1 terá peso 8,0 e a nota N2 terá peso 3,5.
Além disso, serão enviados alguns questionários para os alunos responderem que valem 0,5 ponto adicionado ao total anterior.
Assim, o aluno pode fazer um total de 12 pontos na disciplina. Esse sistema de avaliação permite que o aluno tenha liberdade e tranquilidade de cometer erros ao longo da disciplina, sem que isso tenha impacto significativo na nota.
Outras atividades valendo nota poderão ser realizadas, a depender do andamento da turma.

9. Bibliografia Básica e Complementar

Bibliografia Básica:
1. MEDINA, M.; FERTIG, C. Algoritmos e programação: teoria e prática 2ed. Nova-tec, 2004.ISBN: 9788575220733/857522073X
2. ASCENCIO, A. F. G.;CAMPOS, E. A. V. Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ e Java. 2 ed. Prentice Hall, 2007. ISBN: 978576051480

3. CELES, W.; CERQUEIRA, R.; RANGEL, J. L. Introdução à estrutura de dados: com técnica de programação em C. Elsevier, 2004. ISBN: 8535212280

Bibliografia Complementar:
1. FORBELLONE, A. L. V. ; EBERSPACHER, H.F. Lógica de programação: a construção de algoritmos. 3 ed. Prentice Hall, 2005.
2. Fundamentos de Programação - 3ª Ed. Joyanes, Luis Aguilar; Joyanes, Luis Aguilar. Amgh Editora.
3. Fundamentos De Programação Usando C - 4ª Ed. De Sá, Marques, Lidel – Zamboni.
4. Lógica de Programação - 3ª Edição. Forbellone, Andre L. V. Makron Books.
5. Algoritmos - Lógica para Desenvolvimento de Programação de Computadores. Oliveira, Jayr Figueiredo de; Manzano, Jose Augusto N. G. Editora Erica.

10. Parecer

Assinatura do Professor

Professor Responsável:

Aprovação da Coordenação do Curso

Coordenador do Curso:

Aprovação da Coordenação Acadêmica

Coordenadora Acadêmica:

ATENÇÃO! As informações a serem preenchidas neste formulário devem ser exatamente iguais àquelas constantes no formulário de criação/regulamentação da disciplina aprovado pela Câmara de Graduação.

Documento assinado eletronicamente por **MARKOS OLIVEIRA FREITAS**, **Professor do Magistério Superior**, em 07/03/2023, às 15:12, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufc.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4132565** e o código CRC **54015905**.

Referência: Processo nº 23067.009814/2023-35 SEI nº 4132565

ANEXO B – PLANO DE ENSINO DA DISCIPLINA DE FUNDAMENTOS DA PROGRAMAÇÃO - ENGENHARIA DE SOFTWARE

UFC

CAMPUS DE RUSSAS

PLANO DE ENSINO DE DISCIPLINA

Ano/Semestre: 2023/1

1. Identificação				
1.1. Unidade: Campus Russas				
1.2. Curso: Engenharia de Software				
1.3. Estrutura Curricular (ano-período): 2018.1				
1.4. Nome da Disciplina: Fundamentos de Programação				
1.5. Código da Disciplina: RUS0297				
1.6. Caráter da Disciplina: (X)Obrigatória ( )Optativa CONFORME FORMULÁRIO DE CRIAÇÃO				
1.7. Regime de Oferta da Disciplina: (X)Semestral ( )Anual ( ) Modular				
1.8. Carga Horária (CH) Total: 64h	C.H. Teórica: 32h	C.H. Prática: 32h	C.H. EaD: ---	C.H. Extensão: ---
1.9. Pré-requisitos (quando houver): ---				
1.10. Co-requisitos (quando houver): ---				
1.11. Equivalências (quando houver): Fundamentos de Programação				
1.12. Professor(es): Pablo Luiz Braga Soares				
2. Justificativa				
A disciplina de Fundamentos de Programação propicia a fundamentação básica na arte de programar, envolvendo conceitos de algoritmos, soluções de problemas através de computadores, bem como de linguagens de programação				

estruturada. O aprendizado nas disciplinas que envolvem programação é facilitado na medida em que os alunos adquirem uma base sólida nesta disciplina.

3. Ementa

Algoritmos, Conceitos Fundamentais de Programação, Expressões, Controles de Fluxo, Funções e Procedimentos, Vetores e Matrizes, Cadeias de Caracteres, Tipos Estruturados e Arquivos.

4. Objetivos – Geral e Específicos

Objetivo Geral: Ensinar os alunos os conceitos fundamentais de algoritmos e programação.
Objetivos Específicos: Desenvolver a habilidade dos alunos em programação estruturada, enfocando na construção de programas corretos, confiáveis, seguros, eficientes;
Compreender e utilizar as estruturas de dados básicas, como vetores e matrizes;
Desenvolver a habilidade de solucionar problemas através do computador.

5. Calendário de Atividades

Data	Descrição do Conteúdo	Carga Horária
14/03/2023	Recepção dos Ingressantes	0h
16/03/2023	Apresentação da Disciplina/Ementa	2h
21/03/2023	Introdução à Lógica de Programação	2h
23/03/2023	Tipos Primitivos / Constantes / Variáveis	2h
28/03/2023	Expressões Lógicas/ Expressões Aritméticas	2h
30/03/2023	Comando de Atribuição/ Entrada e Saída	2h
04/04/2023	Estrutura de Seleção Simples e Composta	2h
06/04/2023	Recesso escolar e Ponto facultativo – Semana Santa	0h
11/04/2023	Estrutura de Seleção Simples e Composta	2h
13/04/2023	Estrutura de Seleção Encadeada e Múltipla Escolha	2h
18/04/2023	Sescomp 2023	2h
20/04/2023	Sescomp 2023	2h
25/04/2023	Prova 01	2h
27/04/2023	Correção Prova 01/ Repetição com teste no início(Enquanto)	2h
02/05/2023	Repetição com teste no início(Enquanto)	2h
04/05/2023	Repetição com teste no início(Enquanto)	2h
09/05/2023	Repetição com Variável de Controle(Para)	2h
11/05/2023	Repetição com Variável de Controle(Para)	2h
16/05/2023	Vetores	2h
18/05/2023	Vetores	2h
23/05/2023	Cadeia de Caracteres	2h
25/05/2023	Cadeia de Caracteres	2h
30/05/2023	Matrizes	2h
01/06/2023	Matrizes	2h

06/06/2023	Prova 02	2h
08/06/2023	Ponto facultativo – Corpus Christi	0h
13/06/2023	Correção Prova 02	2h
15/06/2023	Registro	2h
20/06/2023	Registro	2h
22/06/2023	Funções/Procedimentos	2h
27/06/2023	Funções/Procedimentos	2h
29/06/2023	Arquivos	2h
04/07/2023	Arquivos	2h
06/07/2023	Prova 03	2h
11/07/2023	2 Chamada das Provas 01, 02 e 03/Divulgação dos Resultados	2h
18/07/2023	Prova Final	0h

6. Metodologia de Ensino

Aulas teóricas expositivas com o uso do quadro branco, pincel e data show. Além de aulas práticas em laboratório com o uso de computadores, softwares e sistemas operacionais diversos. Estudos individuais e em grupo. Resolução de exercícios no computador. Nas aulas teóricas, a metodologia de ensino será construtiva sempre que possível, focando na exposição das principais estruturas de lógica de programação e diálogo com o aluno. Nas aulas práticas, o conteúdo visto nas aulas teóricas será aplicado por meio da resolução de listas de problemas, na linguagem de programação Python. Além do incentivo na participação de maratonas de programação, onde os alunos serão divididos em equipes de três para resolver problemas diversos.

7. Atividades Discentes

Assiduidade às aulas. Participação do aluno no desenvolvimento das aulas. Lista de exercícios. Provas escritas. Participação do aluno em trabalhos realizados individualmente e/ou em grupo. Participação das atividades organizadas pelos monitores e competições entre equipes.

8. Sistema de Avaliação

Texto padrão:

Conforme o Regimento Geral da UFC, a avaliação de rendimento do aluno far-se-á segundo os critérios de assiduidade e eficiência. Na verificação da assiduidade será aprovado o aluno que frequentar 75% (setenta e cinco por cento) ou mais da carga horária da disciplina, vedado o abono de faltas. A verificação da eficiência compreenderá, no mínimo, duas avaliações progressivas e uma avaliação final. Será aprovado por média o aluno que apresentar média aritmética das notas resultantes das avaliações progressivas igual ou superior a 07 (sete). O aluno que apresentar a média igual ou superior a 04 (quatro) e inferior a 07 (sete), será submetido à avaliação final. Nesse caso, o aluno será aprovado quando obtiver nota igual ou superior a 04 (quatro) na avaliação final e média final igual ou superior a 05 (cinco).

A avaliação de aprendizagem acontecerá na forma de três avaliações progressivas, sendo a média do
aluno obtida de acordo com a seguinte fórmula:

- Presença

1. A presença será computada para os alunos que estiverem presentes ao final de cada aula;
● Avaliações
1. Serão três avaliações denominadas de A1, A2 e A3
2. As avaliações terão início no horário da disciplina e estão marcadas de acordo com o cronograma
deste documento;
3. Cada avaliação terá um prazo de até 2h para sua resolução;
4. Da mesma forma, para os alunos que por qualquer motivo não tenham conseguido comparecer no
dia da avaliação, será concedido o direito de realizar segunda chamada, desde que o aluno faça a
solicitação formal em até no máximo 3 dias úteis após a avaliação.
5. Média final do aluno será dada de acordo com a fórmula a seguir:
6. é = (1+ 2+ 3)/3

9. Bibliografia Básica e Complementar

Bibliografia Básica (sugere-se a inclusão de, pelo menos, 03 títulos):
1.
MEDINA, M.; FERTIG, C. Algoritmos e programação: teoria e prática 2ed. Novatec, 2004.ISBN: 9788575220733/857522073X
2.
ASCENCIO, A. F. G.;CAMPOS, E. A. V. Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ e Java. 2 ed. Prentice Hall, 2007. ISBN: 978576051480
3.
CELES, W.; CERQUEIRA, R.; RANGEL, J. L. Introdução à estrutura de dados: com técnica de programação em C. Elsevier, 2004. ISBN: 8535212280 Bibliografia Complementar (sugere-se a inclusão de, pelo menos, 05 títulos – de acordo com instrumento
de avaliação de Curso de Graduação, INEP/maio-2012 ou legislação posterior):
1.
FORBELLONE, A. L. V. ; EBERSPACHER, H.F. Lógica de programação: a construção de algoritmos. 3 ed. Prentice Hall, 2005.
2.
Fundamentos de Programação - 3ª Ed. Joyanes, Luis Aguilar; Joyanes, Luis Aguilar. Amgh Editora.
3.
Fundamentos de Programação Usando C - 4ª Ed. De Sá, Marques, Lidel – Zamboni.
4.
Lógica de Programação - 3ª Edição. Forbellone, Andre L. V. Makron Books.
5.
Algoritmos - Lógica para Desenvolvimento de Programação de Computadores. Oliveira, Jayr Figueiredo de; Manzano, Jose Augusto N. G. Editora Erica.

10. Parecer

Assinatura do Professor

Professor Responsável:

Aprovação da Coordenação do Curso

Coordenador do Curso:

Aprovação da Coordenação Acadêmica

Coordenadora Acadêmica:

ATENÇÃO! As informações a serem preenchidas neste formulário devem ser exatamente iguais àquelas constantes no formulário de criação/regulamentação da disciplina aprovado pela Câmara de Graduação.

Documento assinado eletronicamente por **PABLO LUIZ BRAGA SOARES**, **Professor do Magistério Superior**, em 29/05/2023, às 15:07, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufc.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4146632** e o código CRC **6B017386**.

Referência: Processo nº 23067.001876/2023-07 SEI nº 4146632

ANEXO C – QUESTIONÁRIO IMI

Intrinsic Motivation Inventory (IMI)

Scale Description

The Intrinsic Motivation Inventory (IMI) is a multidimensional measurement device intended to assess participantsÕ subjective experience related to a target activity in laboratory experiments. It has been used in several experiments related to intrinsic motivation and self-regulation (e.g., Ryan, 1982; Ryan, Mims & Koestner, 1983; Plant & Ryan, 1985; Ryan, Connell, & Plant, 1990; Ryan, Koestner & Deci, 1991; Deci, Eghrari, Patrick, & Leone, 1994). The instrument assesses participantsÕ interest/enjoyment, perceived competence, effort, value/usefulness, felt pressure and tension, and perceived choice while performing a given activity, thus yielding six subscale scores. Recently, a seventh subscale has been added to tap the experiences of relatedness, although the validity of this subscale has yet to be established. The **interest/enjoyment subscale is considered the self-report measure of intrinsic motivation**; thus, although the overall questionnaire is called the Intrinsic Motivation Inventory, it is only the one subscale that assesses intrinsic motivation, per se. As a result, the interest/enjoyment subscale often has more items on it that do the other subscales. The perceived choice and perceived competence concepts are theorized to be positive predictors of both self-report and behavioral measures of intrinsic motivation, and pressure/tension is theorized to be a negative predictor of intrinsic motivation. Effort is a separate variable that is relevant to some motivation questions, so is used it its relevant. The value/usefulness subscale is used in internalization studies (e.g., Deci et al, 1994), the idea being that people internalize and become self-regulating with respect to activities that they experience as useful or valuable for themselves. Finally, the relatedness subscale is used in studies having to do with interpersonal interactions, friendship formation, and so on.

The IMI consists of varied numbers of items from these subscales, all of which have been shown to be factor analytically coherent and stable across a variety of tasks, conditions, and settings. The general criteria for inclusion of items on subscales have been a factor loading of at least 0.6 on the appropriate subscale, and no cross loadings above 0.4. Typically, loadings substantially exceed these criteria. Nonetheless, we recommend that investigators perform their own factor analyses on new data sets. Past research suggests that order effects of item presentation appear to be negligible, and the inclusion or exclusion of specific subscales appears to have no impact on the others. Thus, it is rare that all items have been used in a particular experiment. Instead, experimenters have chosen the subscales that are relevant to the issues they are exploring.

The IMI items have often been modified slightly to fit specific activities. Thus, for example, an item such as ÒI tried very hard to do well at this activityÓ can be changed to ÒI tried very hard to do well on these puzzlesÓ or Ò...in learning this materialÓ without effecting its reliability or validity. As one can readily tell, there is nothing subtle about these items; they are quite face-valid. However, in part, because of their straightforward nature, caution is needed in interpretation. We have found, for example, that correlations between self-reports of effort or interest and behavioral indices of these dimensions are quite modest--often around 0.4. Like other self-report measures, there is always the need to appropriately interpret how and why participants report as they do. Ego-involvements, self-presentation styles, reactance, and other psychological dynamics must be considered. For example, in a study by Ryan, Koestner, and Deci (1991), we found that when participants were ego involved, the engaged in pressured persistence during a free choice period and this behavior did not correlate with the

self-reports of interest/enjoyment. In fact, we concluded that to be confident in oneÕs assessment of intrinsic motivation, one needs to find that the free-choice behavior and the self-reports of interest/enjoyment are significantly correlated.

Another issue is that of redundancy. Items within the subscales overlap considerably, although randomizing their presentation makes this less salient to most participants. Nonetheless, shorter versions have been used and been found to be quite reliable. The incremental R for every item above 4 for any given factor is quite small. Still, it is very important to recognize that multiple item subscales consistently outperform single items for obvious reasons, and they have better external validity.

On The Scale page, there are five sections. First, the full 45 items that make up the 7 subscales are shown, along with information on constructing your own IMI and scoring it. Then, there are four specific versions of the IMI that have been used in past studies. This should give you a sense of the different ways it has been used. These have different numbers of items and different numbers of subscales, and they concern different activities. First, there is a standard, 22-item version that has been used in several studies, with four subscales: interest/enjoyment, perceived competence, perceived choice, and pressure/tension. Second, there is a short 9-item version concerned with the activity of reading some text material; it has three subscales: interest/enjoyment, perceived competence, and pressure/tension. Then, there is the 25-item version that was used in the internalization study, including the three subscales of value/usefulness, interest/enjoyment, and perceived choice. Finally, there is a 29-item version of the interpersonal relatedness questionnaire that has five subscales: relatedness, interest/enjoyment, perceived choice, pressure/tension, and effort.

Finally, McAuley, Duncan, and Tammen (1987) did a study to examine the validity of the IMI and found strong support for its validity.

References

Deci, E. L., Eghrari, H., Patrick, B. C., & Leone, D. (1994). Facilitating internalization: The self-determination theory perspective. *Journal of Personality*, *62*, 119-142.

McAuley, E., Duncan, T., & Tammen, V. V. (1987). Psychometric properties of the Intrinsic Motivation Inventory in a competitive sport setting: A confirmatory factor analysis. *Research Quarterly for Exercise and Sport*, *60*, 48-58.

Plant, R. W., & Ryan, R. M. (1985). Intrinsic motivation and the effects of self-consciousness, self-awareness, and ego-involvement: An investigation of internally-controlling styles. *Journal of Personality*, *53*, 435-449.

Ryan, R. M. (1982). Control and information in the intrapersonal sphere: An extension of cognitive evaluation theory. *Journal of Personality and Social Psychology*, *43*, 450-461.

Ryan, R. M., Connell, J. P., & Plant, R. W. (1990). Emotions in non-directed text learning. *Learning and Individual Differences*, *2*, 1-17.

Ryan, R. M., Koestner, R., & Deci, E. L. (1991). Varied forms of persistence: When free-choice behavior is not intrinsically motivated. *Motivation and Emotion*, *15*, 185-205.

Ryan, R. M., Mims, V., & Koestner, R. (1983). Relation of reward contingency and interpersonal context to intrinsic motivation: A review and test using cognitive evaluation theory. *Journal of Personality and Social Psychology*, *45*, 736-750.

The Scales

THE POST-EXPERIMENTAL INTRINSIC MOTIVATION INVENTORY

(Below are listed all 45 items that can be used depending on which are needed.)

For each of the following statements, please indicate how true it is for you, using the following scale:

1 2 3 4 5 6 7

not at all true — somewhat true — very true

Interest/Enjoyment

I enjoyed doing this activity very much
This activity was fun to do.
I thought this was a boring activity. (R)
This activity did not hold my attention at all. (R)
I would describe this activity as very interesting.
I thought this activity was quite enjoyable.
While I was doing this activity, I was thinking about how much I enjoyed it.

Perceived Competence

I think I am pretty good at this activity.
I think I did pretty well at this activity, compared to other students.
After working at this activity for awhile, I felt pretty competent.
I am satisfied with my performance at this task.
I was pretty skilled at this activity.
This was an activity that I couldnÕt do very well. (R)

Effort/Importance

I put a lot of effort into this.
I didnÕt try very hard to do well at this activity. (R)

I tried very hard on this activity.
It was important to me to do well at this task.
I didnÕt put much energy into this. (R)

Pressure/Tension
I did not feel nervous at all while doing this. (R)
I felt very tense while doing this activity.
I was very relaxed in doing these. (R)
I was anxious while working on this task.
I felt pressured while doing these.

Perceived Choice
I believe I had some choice about doing this activity.
I felt like it was not my own choice to do this task. (R)
I didnÕt really have a choice about doing this task. (R)
I felt like I had to do this. (R)
I did this activity because I had no choice. (R)
I did this activity because I wanted to.
I did this activity because I had to. (R)

Value/Usefulness
I believe this activity could be of some value to me.
I think that doing this activity is useful for _____
I think this is important to do because it can _____
I would be willing to do this again because it has some value to me.
I think doing this activity could help me to _____
I believe doing this activity could be beneficial to me.
I think this is an important activity.

Relatedness
I felt really distant to this person. (R)
I really doubt that this person and I would ever be friends. (R)
I felt like I could really trust this person.
IÕd like a chance to interact with this person more often.
IÕd really prefer not to interact with this person in the future. (R)
I donÕt feel like I could really trust this person. (R)
It is likely that this person and I could become friends if we interacted a lot.
I feel close to this person.

Constructing the IMI for your study. First, decide which of the variables (factors) you want to use, based on what theoretical questions you are addressing. Then, use the items from those factors, randomly ordered. If you use the value/usefulness items, you will need to complete the three items as appropriate. In other words, if you were studying whether the person believes an activity is useful for improving concentration, or becoming a

better basketball player, or whatever, then fill in the blanks with that information. If you do not want to refer to a particular outcome, then just truncate the items with its being useful, helpful, or important.

Scoring information for the IMI. To score this instrument, you must first reverse score the items for which an (R) is shown after them. To do that, subtract the item response from 8, and use the resulting number as the item score. Then, calculate subscale scores by averaging across all of the items on that subscale. The subscale scores are then used in the analyses of relevant questions.

* * * * * * * * * * * *

The following is a 22 item version of the scale that has been used in some lab studies on intrinsic motivation. It has four subscales: interest/enjoyment, perceived choice, perceived competence, and pressure/tension. The interest/enjoyment subscale is considered the self-report measure of intrinsic motivation; perceived choice and perceived competence are theorized to be positive predictors of both self-report and behavioral measures of intrinsic motivation. Pressure tension is theorized to be a negative predictor of intrinsic motivation. Scoring information is presented after the questionnaire itself.

TASK EVALUATION QUESTIONNAIRE

For each of the following statements, please indicate how true it is for you, using the following scale:

1	2	3	4	5	6	7
not at all true			somewhat true			very true

1. While I was working on the task I was thinking about how much I enjoyed it.
2. I did not feel at all nervous about doing the task.
3. I felt that it was my choice to do the task.
4. I think I am pretty good at this task.
5. I found the task very interesting.
6. I felt tense while doing the task.
7. I think I did pretty well at this activity, compared to other students.

8. Doing the task was fun.

9. I felt relaxed while doing the task.

10. I enjoyed doing the task very much.

11. I didnÕt really have a choice about doing the task.

12. I am satisfied with my performance at this task.

13. I was anxious while doing the task.

14. I thought the task was very boring.

15. I felt like I was doing what I wanted to do while I was working on the task.

16. I felt pretty skilled at this task.

17. I thought the task was very interesting.

18. I felt pressured while doing the task.

19. I felt like I had to do the task.

20. I would describe the task as very enjoyable.

21. I did the task because I had no choice.

22. After working at this task for awhile, I felt pretty competent.

Scoring information. Begin by reverse scoring items # 2, 9, 11, 14, 19, 21. In other words, subtract the item response from 8, and use the result as the item score for that item. This way, a higher score will indicate more of the concept described in the subscale name. Thus, a higher score on pressure/tension means the person felt more pressured and tense; a higher score on perceived competence means the person felt more competent; and so on. Then calculate subscale scores by averaging the items scores for the items on each subscale. They are as follows. The (R) after an item number is just a reminder that the item score is the reverse of the participantÕs response on that item.

Interest/enjoyment: 1, 5, 8, 10, 14(R), 17, 20
Perceived competence: 4, 7, 12, 16, 22
Perceived choice: 3, 11(R), 15, 19(R), 21(R)
Pressure/tension: 2(R), 6, 9(R), 13, 18

The subscale scores can then be used as dependent variables, predictors, or mediators, depending on the research questions being addressed.

* * * * * * * * * * * *

TEXT MATERIAL QUESTIONNAIRE I

For each of the following statements, please indicate how true it is for your, using the following scale as a guide:

	1 2 3 4 5 6 7	
not at all true	somewhat true	very true

1. While I was reading this material, I was thinking about how much I enjoyed it.
2. I did not feel at all nervous while reading.
3. This material did not hold my attention at all.
4. I think I understood this material pretty well.
5. I would describe this material as very interesting.
6. I think I understood this material very well, compared to other students.
7. I enjoyed reading this material very much.
8. I felt very tense while reading this material.
9. This material was fun to read.

Scoring information. Begin by reverse scoring items # 2 and 3. In other words, subtract the item response from 8, and use the result as the item score for that item. This way, a higher score will indicate more of the

concept described in the subscale name. Then calculate subscale scores by averaging the items scores for the items on each subscale. They are shown below. The (R) after an item number is just a reminder that the item score is the reverse of the participantÕs response on that item.

Interest/enjoyment: 1, 3(R), 5, 7, 9
Perceived competence: 4, 6,
Pressure/tension: 2(R), 8

* * * * * * * * * * * *

The next version of the questionnaire was used for a study of internalization with an uninteresting computer task (Deci et al., 1994).

ACTIVITY PERCEPTION QUESTIONNAIRE

The following items concern your experience with the task. Please answer all items. For each item, please indicate how true the statement is for you, using the following scale as a guide:

1	2	3	4	5	6	7
not at all true			somewhat true			very true

1. I believe that doing this activity could be of some value for me.
2. I believe I had some choice about doing this activity.
3. While I was doing this activity, I was thinking about how much I enjoyed it.
4. I believe that doing this activity is useful for improved concentration.
5. This activity was fun to do.
6. I think this activity is important for my improvement.
7. I enjoyed doing this activity very much.
8. I really did not have a choice about doing this activity.

9. I did this activity because I wanted to.

10. I think this is an important activity.

11. I felt like I was enjoying the activity while I was doing it.

12. I thought this was a very boring activity.

13. It is possible that this activity could improve my studying habits.

14. I felt like I had no choice but to do this activity.

15. I thought this was a very interesting activity.

16. I am willing to do this activity again because I think it is somewhat useful.

17. I would describe this activity as very enjoyable.

18. I felt like I had to do this activity.

19. I believe doing this activity could be somewhat beneficial for me.

20. I did this activity because I had to.

21. I believe doing this activity could help me do better in school.

22. While doing this activity I felt like I had a choice.

23. I would describe this activity as very fun.

24. I felt like it was not my own choice to do this activity.

25. I would be willing to do this activity again because it has some value for me.

Scoring information. Begin by reverse scoring items # 8, 12, 14, 18, 20, and 24 by subtracting the item response from 8 and using the result as the item score for that item. Then calculate subscale scores by averaging the items scores for the items on each subscale. They are shown below. The (R) after an item number is just a reminder that the item score is the reverse of the participantÕs response on that item.

Interest/enjoyment:	3, 5, 7, 11, 12(R), 15, 17, 23
Value/usefulness:	1, 4, 6, 10, 13, 16, 19, 21, 25
Perceived choice:	2, 8(R), 9, 14(R), 18(R), 20(R), 22, 24(R)

* * * * * * * * * * * *

SUBJECT IMPRESSIONS QUESTIONNAIRE

The following sentences describe thoughts and feelings you may have had regarding the other person who participated in the experiment with you. For each of the following statement please indicate how true it is for you, using the following scale as a guide:

1	2	3	4	5	6	7
not at all true			somewhat true			very true

1. While I was interacting with this person, I was thinking about how much I enjoyed it.
2. I felt really distant to this person.
3. I did not feel at all nervous about interacting with this person.
4. I felt like I had choice about interacting with this person.
5. I would describe interacting with this person as very enjoyable.
6. I really doubt that this person and I would ever become friends.
7. I found this person very interesting.
8. I enjoyed interacting with this person very much.
9. I felt tense while interacting with this person.
10. I really feel like I could trust this person.
11. Interacting with this person was fun.
12. I felt relaxed while interacting with this person.
13. IÕd like a chance to interact more with this person.

14. I didnÕt really have a choice about interacting with this person.

15. I tried hard to have a good interaction with this person.

16. IÕd really prefer not to interact with this person in the future.

17. I was anxious while interacting with this person.

18. I thought this person was very boring.

19. I felt like I was doing what I wanted to do while I was interacting with this person.

20. I tried very hard while interacting with this person.

21. I donÕt feel like I could really trust this person.

22. I thought interacting with this person was very interesting.

23. I felt pressured while interacting with this person.

24. I think itÕs likely that this person and I could become friends.

25. I felt like I had to interact with this person.

26. I feel really close to this person.

27. I didnÕt put much energy into interacting with this person.

28. I interacted with this person because I had no choice.

29. I put some effort into interacting with this person.

Scoring information. Begin by reverse scoring items # 2, 3, 6, 12, 14, 16, 18, 21, 25, 27, and 28 by subtracting the item response from 8 and using the result as the item score for that item. Then calculate subscale scores by averaging the items scores for the items on each subscale. They are shown below. The (R) after an item number is just a reminder that the item score is the reverse of the participantÕs response on that item.

Relatedness:	2(R), 6(R), 10, 13, 16(R), 21(R), 24, 26
Interest/enjoyment:	1, 5, 7, 8, 11, 18(R), 22
Perceived choice:	4, 14(R), 19, 25(R), 28(R)
Pressure/tension:	3(R), 9, 12(R), 17, 23,
Effort:	15, 20, 27(R), 29