

Rafael Noleto Lima

Apoio ao Ensino de Algoritmos por meio da Demonstração de Códigos-Fonte Similares

São Luís

2018

Rafael Noleto Lima

Apoio ao Ensino de Algoritmos por meio da Demonstração de Códigos-Fonte Similares

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Maranhão
Centro de Ciências Exatas e da Tecnologia
Departamento de Informática

Orientador: Prof. Dr. Carlos de Salles Soares Neto

São Luís
2018

Rafael Noleto Lima

Apoio ao Ensino de Algoritmos por meio da Demonstração de Códigos-Fonte Similares/ Rafael Noleto Lima. – São Luís, 2018-
86p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Carlos de Salles Soares Neto

Trabalho de Conclusão de Curso – Universidade Federal do Maranhão
Centro de Ciências Exatas e da Tecnologia
Departamento de Informática , 2018.

1. ensino de algoritmos. 2. motivação. 3. programação. I. Prof. Dr. Carlos de Salles Soares Neto. II. Universidade Federal do Maranhão
Centro de Ciências Exatas e da Tecnologia
Departamento de Informática III. Apoio ao Ensino de Algoritmos por meio da Demonstração de Códigos-Fonte Similares

Rafael Noleto Lima

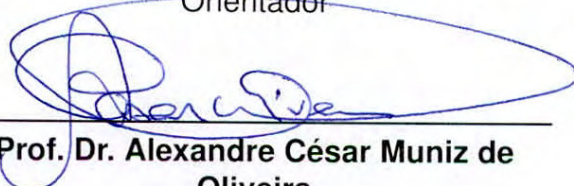
Apoio ao Ensino de Algoritmos por meio da Demonstração de Códigos-Fonte Similares

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. São Luís, 21 de dezembro de 2018:



Prof. Dr. Carlos de Salles Soares Neto
Departamento de Informática - UFMA
Orientador



**Prof. Dr. Alexandre César Muniz de
Oliveira**
Departamento de Informática - UFMA
Examinador



**Prof. Msc. Dilson José Lins Rabêlo
Júnior**
UNASUS - UFMA
Examinador

São Luís
2018

*Este trabalho é dedicado a todos
que acreditaram que este dia chegaria.*

Agradecimentos

A Deus, aos deuses e às deusas, por terem me dado forças para suportar as adversidades da vida.

À minha família, por ser o alicerce nos momentos mais difíceis.

À Universidade Federal do Maranhão, por ter me proporcionado anos de intenso aprendizado, permitindo a minha evolução como pessoa, como estudante e como profissional.

Ao meu orientador Carlos de Salles Soares Neto, pelo apoio incondicional e por ter acreditado em mim, me incentivando a continuar mesmo quando nem eu mesmo acreditava. Sua forma exemplar de dar aulas e cativar os alunos é um modelo a ser seguido.

Às pessoas queridas que, provavelmente, eu nunca teria conhecido se não fosse por intermédio desta longa jornada chamada UFMA, em especial o Bruno, a Cássia, o Leandro, o Tássio e o Flávio.

A todos que, direta ou indiretamente, fizeram parte da minha formação, muito obrigado.

Resumo

Os cursos da área de Computação enfrentam o desafio de combater a alta evasão e desistência de alunos, sobretudo em relação às disciplinas de programação. A dificuldade em assimilar conceitos relacionados a algoritmos desmotiva os estudantes e é imprescindível buscar formas de resgatar o interesse do aluno, seja por meio de novas abordagens no ensino ou por ferramentas tecnológicas disponíveis como as competições online. Desta forma, este estudo visa experimentar metodologias inovadoras com a utilização do juiz online Judge e o desenvolvimento de uma página virtual que sirva como repositório de códigos-fonte de soluções de problemas similares aos cobrados nas competições realizadas em sala, estimulando assim a capacidade de construir conhecimento novo a partir de conceitos prévios. Inspirada na abordagem Problem Based Learning, a disciplina foi ministrada com base em resolução de problemas de competições, com torneios sendo organizados entre os alunos semanalmente. Tendo a competitividade como aliada à motivação, aplicou-se um formulário de avaliação para colher informações a respeito das expectativas dos alunos em relação à disciplina. Ao final, o formulário foi aplicado novamente e as respostas sobre motivação foram comparadas em gráficos.

Palavras-chaves: ensino de algoritmos. motivação. programação.

Abstract

Computing courses face the challenge of combating high student evasion and dropouts, especially related to programming subjects. When students find it hard to assimilate concepts related to algorithms, they get demotivated and it is important for teachers to find ways to reobtain the interest of the student, whether through new teaching techniques or tools such as online competitions. Therefore, this study aims to use innovative methodologies such as the use of an online judge named Judge and the development of a virtual page which serves as a source code repository for problem-solving solutions similar to those assigned in class, thus stimulating the ability to construct new knowledge from previous concepts.

Inspired by the Problem Based Learning, which is based on problem solving algorithm competitions, tournaments have being organized among students weekly. By having competitiveness allied with motivation, an evaluation form was applied to measure students' expectations towards the course. By the end of the semester, the form was re-applied and the answers were compared in charts.

Key-words: algorithm teaching. motivation. programming.

Lista de ilustrações

Figura 1 – Página principal de problemas no UVa Online Judge	23
Figura 2 – Página de submissão de códigos no UVa Online Judge	24
Figura 3 – Página personalizada de um usuário do uHunt	25
Figura 4 – Problemas do assunto Grafo selecionados pelo uHunt	26
Figura 5 – Página de um problema no uDebug	27
Figura 6 – Problemas reunidos por categoria no URI Online Judge	28
Figura 7 – Página principal do SPOJ exibindo problemas	29
Figura 8 – Página principal do CodeForces exibindo problemas	30
Figura 9 – Página principal do CodeForces exibindo torneios	30
Figura 10 – Página inicial do Jutge	36
Figura 11 – Página de problemas do <i>Jutge</i>	38
Figura 12 – Página de torneios (<i>Exams/Contests</i>) do Jutge	39
Figura 13 – Formulário de criação de um torneio no Jutge	40
Figura 14 – Ícones que podem representar uma questão no Jutge	40
Figura 15 – Lista de submissões em um torneio no Jutge	41
Figura 16 – Página principal do blog codigosufma	42
Figura 17 – Página do blog codigosufma exibindo as soluções disponíveis	43
Figura 18 – Respostas às afirmações relacionadas à Autoeficácia	46
Figura 19 – Respostas às afirmações relacionadas à Estratégias de Aprendizagem Ativa	47
Figura 20 – Respostas às afirmações relacionadas ao Valor do Aprendizado de Algoritmos 2	48
Figura 21 – Respostas às afirmações relacionadas ao Objetivo de desempenho	49
Figura 22 – Respostas às afirmações relacionadas ao Objetivo de realização pessoal	50
Figura 23 – Respostas às afirmações relacionadas ao Estímulo do ambiente de aprendizagem	51
Figura 24 – Respostas às afirmações relacionadas às Expectativas quanto ao ambiente de aprendizagem	52
Figura 25 – Comparação entre os resultados da Autoeficácia	58
Figura 26 – Comparação entre os resultados das Estratégias de Aprendizagem Ativa	59
Figura 27 – Comparação entre os resultados do Valor de Aprendizado de Algoritmos 2	60
Figura 28 – Comparação entre os resultados do Objetivo de desempenho	61
Figura 29 – Comparação entre os resultados do Objetivo de realização pessoal	62

Figura 30 – Comparação entre os resultados do Estímulo do ambiente de aprendizagem	63
Figura 31 – Comparação entre os resultados das Expectativas com o ambiente de aprendizagem	64

Lista de tabelas

Tabela 1 – Comparações entre os Ambientes Virtuais	31
Tabela 2 – Possíveis vereditos emitidos pelo Jutge	37
Tabela 3 – Classificação de Respostas ao Formulário de Motivação	45
Tabela 4 – Afirmções relacionadas à Autoeficácia	46
Tabela 5 – Afirmções relacionadas às Estratégias de Aprendizagem Ativa	47
Tabela 6 – Afirmções relacionadas ao Valor do Aprendizado de Algoritmos 2	48
Tabela 7 – Afirmções relacionadas ao Objetivo de desempenho	49
Tabela 8 – Afirmções relacionadas ao Objetivo de realização pessoal	50
Tabela 9 – Afirmções relacionadas ao Estímulo do ambiente de aprendizagem	51
Tabela 10 – Afirmções relacionadas às Expectativas quanto ao ambiente de aprendizagem	52
Tabela 11 – Visualização da Lista de Códigos	53
Tabela 11 – Visualização da Lista de Códigos (continuação)	54
Tabela 12 – Visualização de Código-Fonte	54
Tabela 12 – Visualização de Código-Fonte (continuação)	55
Tabela 12 – Visualização de Código-Fonte (continuação)	56
Tabela 13 – Quantidade de códigos acessados por aluno	56

Lista de abreviaturas e siglas

ACM	Association for Computing Machinery
ICPC	International Collegiate Programming Contest
PBL	Problem Based-Learning
PHP	PHP Hypertext Preprocessor
SPOJ	Sphere Online Judge
SQL	Structured Query Language
URI	Universidade Regional Integrada
UVa	Universidad de Valladolid

Sumário

1	INTRODUÇÃO	21
1.1	Justificativa	21
1.2	Objetivos	22
1.2.1	Objetivo Geral	22
1.2.2	Objetivos Específicos	22
2	TRABALHOS RELACIONADOS	23
2.1	Ambientes Virtuais para Prática de Programação	23
2.1.1	UVa Online Judge	23
2.1.2	Ambiente uHunt	24
2.1.3	Ambiente uDebug	26
2.1.4	URI Online Judge	27
2.1.5	Sphere Online Judge (SPOJ)	28
2.1.6	CodeForces	29
2.2	Novas abordagens no ensino de Algoritmos	32
3	FERRAMENTAS METODOLÓGICAS ADOTADAS	35
3.1	Algoritmos 2: um experimento controlado	35
3.2	Judge.org - avaliador online	36
3.2.1	Como criar um problema no Judge	38
3.2.2	Como criar um torneio no Judge	38
3.2.3	Uso de torneios no Judge na disciplina Algoritmos 2	40
3.3	Blog codigosufma	42
4	DADOS COLETADOS	45
4.1	Formulário de Avaliação de Motivação	45
4.2	Respostas da primeira aplicação do formulário	46
4.2.1	Autoeficácia	46
4.2.2	Estratégias de Aprendizagem Ativa	47
4.2.3	Valor do Aprendizado de Algoritmos 2	48
4.2.4	Objetivo de desempenho	49
4.2.5	Objetivo de realização pessoal	50
4.2.6	Estímulo do ambiente de aprendizagem	51
4.2.7	Expectativas quanto ao ambiente de aprendizagem (Judge)	52
4.3	Dados de utilização do <i>blog</i> codigosufma pelos alunos	53

4.3.1	Quantas vezes os alunos entraram no <i>site</i> e visualizaram a Lista de Códigos?	53
4.3.2	Quantas vezes os alunos entraram no <i>site</i> e visualizaram um Código-Fonte?	54
4.4	Análise entre as duas aplicações do formulário	56
4.4.1	Respostas subjetivas sobre o Jutge e o <i>blog</i>	57
4.4.2	Comparação de resultados: Autoeficácia	58
4.4.3	Comparação de resultados: Estratégias de Aprendizagem Ativa	59
4.4.4	Comparação de resultados: Valor de Aprendizado de Algoritmos 2	60
4.4.5	Comparação de resultados: Objetivo de desempenho	61
4.4.6	Comparação de resultados: Objetivo de realização pessoal	62
4.4.7	Comparação de resultados: Estímulo do ambiente de aprendizagem	63
4.4.8	Comparação de resultados: Expectativas com o ambiente de aprendizagem	64
5	CONCLUSÃO	65
5.1	Discussão dos Resultados	65
5.2	Trabalhos futuros	66
	REFERÊNCIAS	67
	APÊNDICE A – FORMULÁRIO DE MOTIVAÇÃO	69
	APÊNDICE B – BANCO DE LOGS COMPLETO DO BLOG CODI-GOSUFMA	71
	APÊNDICE C – LISTA DE ARQUIVOS NECESSÁRIOS PARA A CRIAÇÃO DE UM PROBLEMA NO JUTGE	75
	APÊNDICE D – ALGUMAS QUESTÕES CRIADAS NO JUTGE	79

1 Introdução

A disciplina de Algoritmos evidencia-se como um dos pilares do curso de Ciência da Computação, razão pela qual é apresentada aos alunos desde os semestres iniciais de sua formação. Segundo o Parecer CNE/CES 136/2012, emitido pela Câmara de Educação Superior do Conselho Nacional da Educação, órgão vinculado ao Ministério da Educação, que trata das Diretrizes Curriculares Nacionais para os cursos de graduação em Computação, “os cursos de bacharelado e licenciatura em Computação devem formar profissionais que revelem pelo menos as competências e habilidades comuns para identificar problemas que tenham solução algorítmica” (CNE/CES, 2002).

Em contrapartida, estudos mostram que uma das razões para os altos índices de evasão e desistência em disciplinas de programação são as dificuldades encontradas pelos alunos no processo de assimilação de conceitos relacionados a algoritmos, tornando esse aprendizado uma fonte de medo e frustração (CASTRO et al., 2003).

Desta forma, faz-se necessário buscar novas estratégias de ensino de algoritmos com o objetivo de despertar o interesse para a disciplina, contribuindo assim para um aumento na motivação dos alunos nos cursos de Computação.

1.1 Justificativa

Ao analisar os livros didáticos que trabalham algoritmos para a Computação, é possível perceber três formas principais de representar um algoritmo: seja por um algoritmo genérico, geralmente de forma textual, descrita em linguagem natural; uma gráfica chamada de Diagrama de Blocos; e, por fim, outra próxima à escrita de um código em linguagem de programação chamada de Pseudocódigo (TIMMERMANN, 2015).

Ao definir a metodologia de avaliação desta disciplina por meio da demonstração de códigos para resolução de problemas com solução algorítmica, a proposta deste projeto é ir além da abordagem de pseudocódigos e desenvolver uma técnica de ensino que permita ao aluno compreender e assimilar conceitos de algoritmos ao ter acesso a códigos-fonte que solucionem questões semelhantes às propostas em sala de aula, buscando, portanto, aumentar a motivação, a participação e o engajamento dos estudantes na disciplina de Algoritmos 2.

1.2 Objetivos

A seguir são definidos os objetivos gerais e específicos deste projeto.

1.2.1 Objetivo Geral

Esta monografia tem como objetivo geral contribuir com uma metodologia complementar ao ensino de algoritmos, tendo um repositório de exemplos de códigos-fonte como material de apoio.

1.2.2 Objetivos Específicos

Para a análise e alcance do objetivo geral, os objetivos específicos deste trabalho são:

- Identificar a variação no índice de motivação dos alunos na disciplina com a utilização da demonstração de códigos-fonte;
- Analisar os acessos dos alunos aos códigos-fonte de acordo com os tipos de problema;
- Verificar se há aumento na frequência de acessos dos alunos ao material de apoio durante o período das avaliações.

2 Trabalhos Relacionados

Este Capítulo apresenta diversos trabalhos e ferramentas relacionadas que se destinam a servir como um ambiente virtual de aprendizado para o ensino de algoritmos. O Capítulo apresenta, também, alguns estudos com disciplinas introdutórias de algoritmos.

2.1 Ambientes Virtuais para Prática de Programação

2.1.1 UVa Online Judge

O UVa Online Judge¹ é o mais tradicional portal *web* com problemas computacionais criado para o treinamento de alunos para o ICPC (International Collegiate Programming Contest). O site atualmente possui centenas de problemas e serviu de base para diversos ambientes de suporte integrados.

A [Figura 1](#) apresenta a tela principal onde estão reunidos os problemas do UVa Online Judge. O sistema permite o envio de soluções em diversas linguagens de programação como C, C++, Java, Pascal e Python, conforme mostra a [Figura 2](#):

Figura 1 – Página principal de problemas no UVa Online Judge

The screenshot shows the UVa Online Judge homepage. At the top, there is a navigation bar with the UVa logo and a banner for Embratel. Below the banner, there is a search bar and a 'Browse Problems' button. The main content area is titled 'Browse Problems' and contains a table of problem sets. The table has three columns: 'Title', 'Total Submissions / Solving %', and 'Total Users / Solving %'. The table lists various problem sets, including 'Problem Set Volumes (100...1999)', 'Contest Volumes (10000...)', 'Interactive Problems', 'Programming Challenges (Skiena & Revilla)', 'ACM-ICPC World Finals', 'ACM-ICPC Dhaka Site Regional Contests', 'Western and Southwestern European Regionals', 'Prominent Problemsetters', 'Rujia Liu's Presents', 'AOAPC I: Beginning Algorithm Contests (Rujia Liu)', 'AOAPC I: Beginning Algorithm Contests -- Training Guide (Rujia Liu)', 'AOAPC II: Beginning Algorithm Contests (Second Edition) (Rujia Liu)', 'Competitive Programming: Increasing the Lower Bound of Programming Contests (Steven & Felix Halim)', 'Competitive Programming 2: This increases the lower bound of Programming Contests. Again (Steven & Felix Halim)', and 'Competitive Programming 3: The New Lower Bound of Programming Contests (Steven & Felix Halim)'. At the bottom of the table, there is a pagination control showing '<< Start < Prev Next > End >>' and a 'Display # 5' dropdown menu.

¹ (UVa Online Judge <<https://www.uvaonlinejudge.org>>, Acesso em: dez/2018)

Figura 2 – Página de submissão de códigos no UVa Online Judge

O UVa Online Judge é um ambiente interessante para o treinamento de programadores com certo nível de experiência pois permite estudar e aplicar novas técnicas de programação com a solução de problemas práticos. No entanto, o *site* não parece muito adequado para quem está aprendendo a programar, já que não possui uma forma de apresentar os problemas classificados por níveis de dificuldade ou por assunto. Conforme visto na [Figura 1](#), o agrupamento dos problemas é feito de acordo com as competições de programação realizadas, o que torna o nível elevado demais para um iniciante em algoritmos.

2.1.2 Ambiente uHunt

O ambiente uHunt² é um complemento ao *site* UVa Online Judge e foi desenvolvido por Felix Halim, motivado por sua própria curiosidade em encontrar problemas algorítmicos para resolver. Esta ferramenta corrige a lacuna do UVa e traz os problemas agrupados por temas e níveis de dificuldade, além de informar quais possuem maior número de submissões aceitas, o que facilita ao programador iniciante encontrar e resolver problemas que sejam mais fáceis.

A [Figura 3](#), a seguir, mostra a página personalizada de um usuário do uHunt. No

² (uHunt <<https://uhunt.onlinejudge.org/>>, Acesso em: dez/2018)

canto superior direito, há uma janela de conversa em que é possível tirar dúvidas com outros participantes cadastrados. Abaixo, a plataforma mostra estatísticas relativas ao usuário, como últimos problemas resolvidos, quantidade de submissões aceitas e um gráfico com a evolução do programador de acordo com o número de questões feitas ao passar dos anos.

Figura 3 – Página personalizada de um usuário do uHunt

The screenshot shows the uHunt user interface. At the top left is the uHunt logo with the tagline 'hunt problems that matter'. Below it, a brief description states: 'uHunt is a complementary tool for UVa online-judge that keeps statistics, provide selections of problems to solve, and exposes a web API for other web developers to build upon it. See a brief history of uHunt.' There are input fields for 'Your UVa username' and 'Search Problem Number'. A chat window on the right shows a conversation with users like Sadman_AKF, imadnsn, and others. Below the chat is a 'Live Submissions' table with columns for #, Problem Title, User, Verdict, Lang, Time, Best, Rank, and Submit Time. The table lists several submissions for problem 11172.

Rafael (rafanoletto) statistics

The statistics page for Rafael (rafanoletto) features several components:

- Progress over the Years:** A line graph showing the cumulative number of submissions from year 11 to 18, reaching a total of 349.
- Submissions Statistics:** A bar chart showing the distribution of submissions across different verdicts: AC (365), PE (4), WA (70), TL (23), ML (0), CE (17), RE (21), and OT (0).
- Last Submissions:** A table listing recent submissions with columns for Problem, Verdict, Lang, Time, Best, Rank, and Submit Time. Problems listed include 10305, 10048, 341, 10986, and 10986.
- Solved Submissions:** A long list of problem IDs that have been solved, starting from 100 and ending with 12577.
- Summary:** 'Solved : 349, Submissions : 500' and 'Tried but not yet solved : 23'.

O uHunt é uma ferramenta poderosa para ajudar estudantes de algoritmos pois permite ao usuário escolher questões compatíveis com o nível de experiência do programador, bem como selecionar aquelas de temas específicos, o que ajuda na fixação do tópico tratado em sala de aula.


Felix Halim, criador do site, escreveu em parceria com seu irmão, Steven Halim, o livro Competitive Programming Exercises, que utiliza os problemas do UVa Online Judge para ensinar algoritmos e técnicas de programação. A obra, que já está em sua

terceira edição, aborda as questões dividindo-as por categorias: Introdução, Estruturas de Dados e Bibliotecas, Paradigmas da Programação, Grafos, Problemas Matemáticos, Processamento de Strings, Geometria Computacional, Tópicos Avançados e Tópicos Raros.

Para exemplificar, a Figura 4 ilustra uma seleção de questões relacionadas ao assunto Grafo, trabalhadas no livro.

Figura 4 – Problemas do assunto Grafo selecionados pelo uHunt

Competitive Programming Exercises



Steven Halim and I published the [Competitive Programming](#) book which is targeted to help regular computer science students to quickly get up and running for the [ACM ICPC](#) as well as [IOI](#). The book discusses the types of problems that are frequently occurs in programming contests. The exercises have been integrated to this uHunt tool so that you can keep track of your progress. To get started, select a chapter from the table on the right. Each chapter has starred problems (i.e., a must try problem). Happy solving :)

FB Page | Info | Buy
Edition: 1st, 2nd, 3rd

3rd Edition's Exercises (switch to: 1st, 2nd, 3rd)		
Book Chapters	Starred ★	ALL
1. Introduction	53%	44%
2. Data Structures and Libraries	35%	26%
3. Problem Solving Paradigms	21%	15%
4. Graph	15%	13%
5. Mathematics	38%	27%
6. String Processing	28%	18%
7. (Computational) Geometry	5%	8%
8. More Advanced Topics	0%	0%
9. Rare Topics	16%	14%

Graph Traversal (6/18 = 33%)

Just Graph Traversal (0/3)

- 11831 - Sticker Collector Robot ★ | discuss Lev 3 **Tried (2)**
- 11906 - Knight in a War Grid ★ | discuss Lev 3 --- ? ---
- 12442 - Forwarding Emails ★ | discuss Lev 3 --- ? ---

Flood Fill/Finding Connected Components (2/3)

- 1103 - Ancient Messages ★ | discuss Lev 3 --- ? ---
- 11094 - Continents ★ | discuss Lev 2 ✓ 0.0005/1288
- 11953 - Battleships ★ | discuss Lev 2 ✓ 0.0005/487(1)

Topological Sort (2/3)

- 872 - Ordering ★ | discuss Lev 3 --- ? ---
- 10305 - Ordering Tasks ★ | discuss Lev 1 ✓ 0.0005/4542
- 11060 - Beverages ★ | discuss Lev 2 ✓ 0.0005/902

Bipartite Graph Check (1/3)

- 10004 - Bicoloring ★ | discuss Lev 1 ✓ 0.0005/4055
- 11080 - Place the Guards ★ | discuss Lev 3 --- ? ---
- 11396 - Claw Decomposition ★ | discuss Lev 3 --- ? ---

Finding Articulation Points/Bridges (0/3)

- 315 - Network ★ | discuss Lev 2 --- ? ---
- 796 - Critical Links ★ | discuss Lev 2 --- ? ---
- 10765 - Doves and bombs ★ | discuss Lev 3 --- ? ---

Finding Strongly Connected Components (1/3)

- 247 - Calling Circles ★ | discuss Lev 2 --- ? ---
- 11504 - Dominos ★ | discuss Lev 2 --- ? ---
- 11838 - Come and Go ★ | discuss Lev 3 ✓ 0.0405/742

Minimum Spanning Tree (2/6 = 33%)

Standard (1/3)

- 11228 - Transportation system. ★ | discuss Lev 3 --- ? ---
- 11631 - Dark roads ★ | discuss Lev 2 **Tried (1)**
- 11747 - Heavy Cycle Edges ★ | discuss Lev 3 ✓ 0.0005/556

Variants (1/3)

- 10048 - Audiophobia ★ | discuss Lev 2 ✓ 0.0005/653
- 10369 - Arctic Network ★ | discuss Lev 2 --- ? ---
- 10600 - ACM Contest and Blacko... ★ | discuss Lev 2 --- ? ---

All-Pairs Shortest Paths (0/6 = 0%)

Standard (0/3)

- 821 - Page Hopping ★ | discuss Lev 2 --- ? ---
- 10171 - Meeting Prof. Miguel... ★ | discuss Lev 3 --- ? ---
- 11463 - Commandos ★ | discuss Lev 3 --- ? ---

Variants (0/3)

Single-Source Shortest Paths (SSSP) (3/15 = 20%)

On Unweighted Graph: BFS, Easier (2/3)

- 429 - Word Transformation ★ | discuss Lev 2 **Tried (1)**
- 924 - Spreading The News ★ | discuss Lev 2 ✓ 0.0005/450
- 10553 - Bombs! NO they are Min... ★ | discuss Lev 2 ✓ 0.2705/1260

On Unweighted Graph: BFS, Harder (0/3)

- 314 - Robot ★ | discuss Lev 3 --- ? ---
- 11101 - Mall Mania ★ | discuss Lev 4 --- ? ---
- 12160 - Unlock the Lock ★ | discuss Lev 3 --- ? ---

On Weighted Graph: Dijkstra's, Easier (1/3)

- 929 - Number Maze ★ | discuss Lev 2 --- ? ---
- 1112 - Mice and Maze ★ | discuss Lev 3 ✓ 0.0005/1145
- 10986 - Sending email ★ | discuss Lev 2 **Tried (4)**

On Weighted Graph: Dijkstra's, Harder (0/3)

- 10801 - Lift Hopping ★ | discuss Lev 3 --- ? ---
- 11492 - Babel ★ | discuss Lev 3 --- ? ---
- 12047 - Highest Paid Toll ★ | discuss Lev 4 --- ? ---

On Graph with Negative Weight Cycle: Bellman Ford's (0/3)

- 558 - Wormholes ★ | discuss Lev 2 --- ? ---
- 10449 - Traffic ★ | discuss Lev 4 --- ? ---
- 10557 - XYZZY ★ | discuss Lev 3 --- ? ---

2.1.3 Ambiente uDebug

A plataforma uDebug³ é uma comunidade em que os usuários podem contribuir com dicas sobre os problemas e oferecer casos de teste críticos que auxiliam no momento de verificar se o seu algoritmo está correto. Além dos problemas presentes no UVa Online Judge, o uDebug também traz ajuda sobre problemas de outros juízes online, como os da ACM-ICPC⁴, da Google Code Jam⁵ e do Facebook Hacker Cup⁶.

³ (uDebug <<https://www.udebug.com>>, Acesso em: dez/2018)

⁴ (ACM-ICPC <<https://icpc.baylor.edu/>>, Acesso em: dez/2018)

⁵ (Google Code Jam <<https://codingcompetitions.withgoogle.com/codejam>>, Acesso em: dez/2018)

⁶ (Facebook Hacker Cup <<http://www.facebook.com/hackercup>>, Acesso em: dez/2018)

A Figura 5, a seguir, mostra a página referente a um problema. Neste caso, cinco usuários adicionaram casos de teste e, mais abaixo no campo *Accepted Output*, o sistema mostra qual a saída correta correspondente à entrada que consta no campo *Input* e que deve ser comparada com a saída que o programa do usuário retorna.

Figura 5 – Página de um problema no uDebug

The screenshot shows the uDebug interface for a problem. At the top, there are links for 'Problems', 'Contributors', and 'Sign In'. Below the header, there are three main sections:

- Select Input (5):** A table listing five users with their input cases, dates, and votes. Each row includes a 'Sign Up to Vote' button.
- Input:** A text area containing the input cases for the selected users.
- Accepted Output:** A text area showing the correct output for the input cases.
- Your Output:** A text area for the user to enter their own output, with instructions to run code and compare.

Buttons for 'Add Input', 'Delete Input', 'Copy Input', 'Get Accepted Output', 'Copy Output', 'Clear', and 'Compare Outputs' are visible throughout the interface.

User	Date	Votes
1 brianfry713	03 May 2016 10:58:30	148
2 morris821028	03 May 2016 10:58:30	113
3 Ryuuk	08 Sep 2016 23:22:34	69
4 jakepezaro	12 Dec 2016 13:54:22	63
5 testitem qlstudio	26 Nov 2016 00:55:00	59

```

Input
20 20
9999 9999
1 9999
340 3000
3000 340
500 101
1 1
9999 9998

Accepted Output
20 20 8
9999 9999 92
1 9999 262
340 3000 217
3000 340 217
500 101 144
1 1 1
9999 9998 92

Your Output
1. Run your code with the same input as above.
2. Paste your output here.
3. Press "Compare Outputs".
  
```

O uDebug oferece aos usuários a possibilidade de confrontar casos de teste críticos, ajudando, portanto, a validar o algoritmo desenvolvido. Desta forma, o estudante ganha experiência com os problemas e começa a pensar de maneira menos trivial, trabalhando o refinamento de seu código para que esteja preparado para lidar com exemplos de entrada e casos de teste não convencionais.

2.1.4 URI Online Judge

O URI Online Judge⁷ é um repositório de questões, semelhante ao UVa Online Judge, que conta atualmente com quase 2000 exercícios, divididos em categorias que

⁷ (URI Online Judge <<http://www.urionlinejudge.com.br/>>, Acesso em: dez/2018)

vão desde o nível básico até o avançado. Esta plataforma foi desenvolvida pelo professor Neilor Turin e pelo aluno Jean Bez, ambos da Universidade Regional Integrada, na cidade de Erechim, ao norte do Rio Grande do Sul.

A Figura 6 apresenta a página de categorias dos problemas. Além de classificações já conhecidas como Strings e Grafos, o site evidencia uma categoria especial chamada Iniciante, dedicada a problemas básicos para quem está iniciando na programação.

Figura 6 – Problemas reunidos por categoria no URI Online Judge

CATEGORIAS	
SELECIONE UMA DAS 9 GRANDES CATEGORIAS DE PROBLEMAS PARA COMEÇAR A RESOLVER.	
1	INICIANTE Problemas básicos para quem está iniciando na programação... 308 PROBLEMAS
2	AD-HOC Problemas de Simulação, Datas e Ad-Hoc no geral... 636 PROBLEMAS
3	STRINGS Palíndromos, Frequência, Ad-Hoc, LCS, Manipulação de Strings... 122 PROBLEMAS
4	ESTRUTURAS E BIBLIOTECAS Filas, Pilhas, Ordenação, Mapas... 141 PROBLEMAS
5	MATEMÁTICA Sistemas Numéricos, Número Primos, BigInteger... 211 PROBLEMAS
6	PARADIGMAS Programação Dinâmica, Busca Binária, Gulosos, Backtracking... 186 PROBLEMAS
7	GRAFOS Flood Fill, MST, SSSP, DAG, Fluxo Máximo, Árvores... 213 PROBLEMAS
8	GEOMETRIA COMPUTACIONAL Pontos e Linhas, Polígonos... 70 PROBLEMAS
9	SQL Linguagens de Consulta: Seleção, Inserção, Atualização, Criação 34 PROBLEMAS
AUTORES	Todos os problemas disponíveis agrupados por autor. 234 AUTORES
LISTAR TODOS	Todos os problemas em um só lugar. 1904 PROBLEMAS
ORIGENS	Todos os problemas agrupados por competições ou eventos. 162 ORIGENS

TOP 20

- Maycon Alves
- Gabriel Duarte
- Luiz Joaquim
- Erick Leonardo de...
- Gustavo Policarpo
- Gabriel Leonardo ...
- Thalyson Nepomuceno
- Luis Fernando Ver...
- Mtheh (IUBAT)
- Sleeping
- Wyllian Brito
- Diego Rangel
- Eduardo Theodoro ...
- Thaddeus Hieronymus
- [Traveling Balloo...
- Ricardo Oliveira
- Renan Tashiro
- Rodolfo Riyoei Goya
- Marcello Marques
- Feodorv

17/12/18 03:36

Com um visual atraente e uma extensa base de questões escritas em português, o URI Online Judge se mostra uma ótima alternativa ao estrangeiro UVa Online Judge. Outra possibilidade disponibilizada pelo URI Online Judge é a criação de *contests*, que são torneios semelhantes às maratonas de programação organizadas pelo ICPC (International Collegiate Programming Contest).

2.1.5 Sphere Online Judge (SPOJ)

O Sphere Online Judge (SPOJ)⁸ é um juiz *online* administrado pela empresa polonesa Sphere Research Labs e bastante utilizado para treinos de seletivas de ma-

⁸ (Sphere Online Judge (SPOJ) <<https://www.spoj.com/>>, Acesso em: dez/2018)

ratonas de programação. O *site* conta com um fórum onde os participantes discutem formas de resolver as questões e uma de suas grandes vantagens é a capacidade de aceitar submissões em mais de 40 linguagens de programação, dentre elas as tradicionais C, C++, Java e Python e a linguagem esotérica *Brainfuck*.

Na [Figura 7](#), é possível ver a página principal do SPOJ Brasil exibindo um conjunto de questões ordenadas pelo número de pessoas que já solucionaram. Outra opção de ordenação é classificando as que tiveram o maior percentual de submissões aceitas. Apesar de sua aparência simples, o *site* possui uma versão em português e seu fórum de dúvidas é uma boa ferramenta para ajudar os estudantes.

Figura 7 – Página principal do SPOJ exibindo problemas

The screenshot shows the SPOJ Brasil website interface. At the top, there's a header with the SPOJ logo and 'SPOJ Brasil' text. Below the header, there are tabs for 'All problems', 'TODO problems', and 'Unsolved problems'. A list of programming languages is displayed: AIIDA95, ASM32, ASM32-GCC, MAWK, GAWK, BASH, BC, BF, C-CLANG, C, CSHARP, CPP, C++, 4.3.2, CPP14-CLANG, CPP14, C99, CLPS, COBOL, COFFEE, LISP, sbcl, LISP, clisp, D-DMD, D-CLANG, D, DART, ELIXIR, FANTOM, FORTH, FORTRAN, GO, GOSU, GRV, HASK, ICON, ICK, JAR, JAVA, JS-MONKEY, JS-RHINO, KTLN, LUA, NEM, NICE, NIM, NODEJS, OBJC, OBJC-CLANG, OCAML, OCT, PAS-FPC, PAS-GPC, PERL, PHP, PICO, PIKE, PRLG-swi, PROLOG, PYTHON, PYPY, PYTHON3, R, RACKET, RUBY, RUST, SCM, guile, SCM, qobi, CHICKEN, SED, ST, SQLITE, SWIFT, UNLAMBDA, VB.NET, WHITESPACE.

The main section is titled 'Lista de problemas seletivos problems'. It features a table with the following columns: ID, S, NAME, CÓDIGO, usuários, and ACC %. The table lists 17 problems, each with a status icon (a red checkmark) and a percentage of accepted submissions.

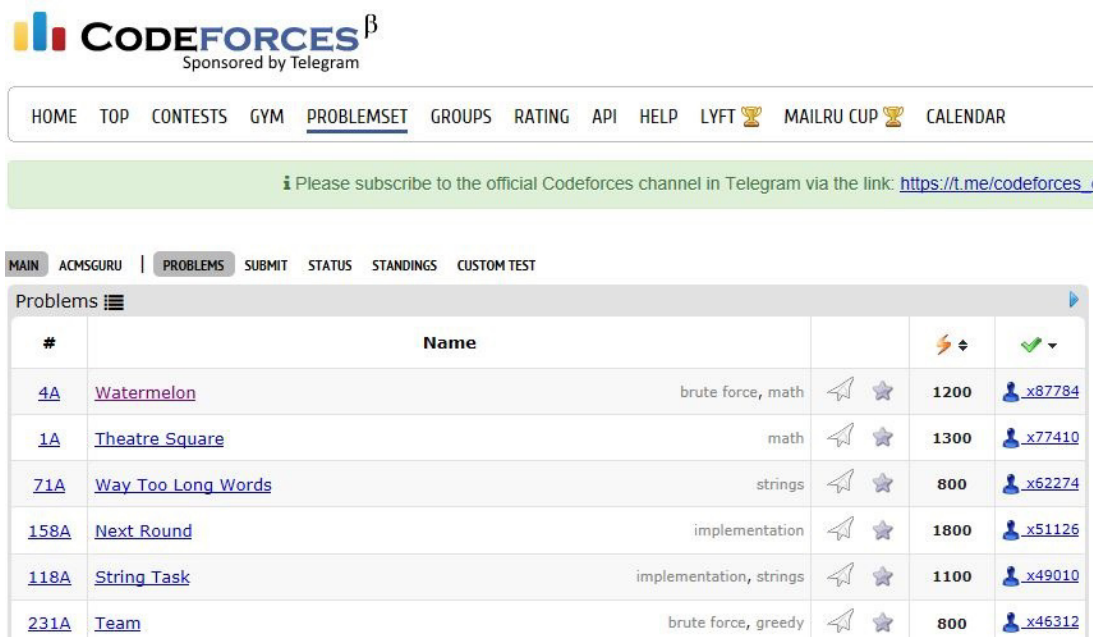
ID	S	NAME	CÓDIGO	usuários	ACC %
1831	✓	f91	F91	4121	36.78
1734	✓	Quem vai ser reprovado	PLACAR	2667	18.68
2839	✓	Popularidade	POPULAR	2663	45.81
2280	✓	Encontre o telefone	ENCOTEL	2539	36.55
2284	✓	Palavras primas	PAPRIMAS	1971	29.77
2844	✓	Você pode dizer 11	ONZE	1794	21.29
1745	✓	Recuperação	RECUPERA	1605	25.95
2846	✓	Paridade	PARIDADE	1511	39.32
1763	✓	Sudoku	SUDOIME	1351	29.42
1821	✓	O Bolo de Apostas	BAPOSTAS	1122	26.05
1850	✓	Conte os Fatores	CFATORES	1093	37.22
2836	✓	Moedas	MOEDAS	1032	41.60
1737	✓	Mesa da Sra Montagny!	MESA	957	22.95
2281	✓	Rumo aos 9s	RUMO9S	913	36.41

2.1.6 CodeForces

O CodeForces⁹ é um juiz online criado e mantido por um grupo de competidores de maratonas da Universidade de Saratov, na Rússia. Com foco principal em competições de programação, cada usuário é classificado por um sistema de *rating* que determina seu nível dentro do *site* e as questões são classificadas pela dificuldade, com *tags* atribuídas a cada uma, permitindo ao estudante buscar problemas que envolvam um assunto específico, conforme [Figura 8](#).

⁹ (CodeForces <<http://codeforces.com/>>, Acesso em: dez/2018)

Figura 8 – Página principal do CodeForces exibindo problemas



The screenshot shows the CodeForces homepage. At the top, there is a navigation bar with links: HOME, TOP, CONTESTS, GYM, **PROBLEMSET**, GROUPS, RATING, API, HELP, LYFT, MAILRU CUP, and CALENDAR. Below the navigation bar, there is a green banner with a message: "Please subscribe to the official Codeforces channel in Telegram via the link: <https://t.me/codeforces>".

Below the banner, there is a secondary navigation bar with links: MAIN, ACMSGURU, **PROBLEMS**, SUBMIT, STATUS, STANDINGS, and CUSTOM TEST. The main content area is titled "Problems" and displays a table of problems.

#	Name	Tags	Difficulty	Rating	Author
4A	Watermelon	brute force, math	1200	x87784	
1A	Theatre Square	math	1300	x77410	
71A	Way Too Long Words	strings	800	x62274	
158A	Next Round	implementation	1800	x51126	
118A	String Task	implementation, strings	1100	x49010	
231A	Team	brute force, greedy	800	x46312	

Regularmente, são organizados torneios locais e mundais virtualmente (Figura 9), com direito a camisetas aos participantes e até prêmios em dinheiro. Recentemente, a empresa Lyft, maior concorrente do Uber nos Estados Unidos, criou o torneio "Lyft Level 5 Challenge 2018"¹⁰, hospedado pelo CodeForces e que distribuiu prêmios e oportunidades de estágio a seus vencedores.

Figura 9 – Página principal do CodeForces exibindo torneios

Contest history

Past contests

Name	Writers	Start	Length	Results	Author
Codeforces Round #527 (Div. 3) Enter > Virtual participation >	MikeMirzayanov Vovuh	Dec/18/2018 11:35 ^{UTC-3}	02:00	Preliminary results open hacking: 08:15:44	x9322
Avito Cool Challenge 2018 Enter > Virtual participation >	fateice fizzq2002 quailty yanQval yjq_naive	Dec/16/2018 11:35 ^{UTC-3}	02:30	Final standings	x6333
Educational Codeforces Round 56 (Rated for Div. 2) Enter > Virtual participation >	Ajosteen BledDest PikMike Vovuh adedalic	Dec/15/2018 11:35 ^{UTC-3}	02:00	Final standings	x7951
Codeforces Round #526 (Div. 1) Enter > Virtual participation >	Dimon egor.jifar maxim.shuklin savkinds2089	Dec/10/2018 13:35 ^{UTC-3}	02:00	Final standings	x675
Codeforces Round #526 (Div. 2) Enter > Virtual participation >	Dimon egor.jifar maxim.shuklin savkinds2089	Dec/10/2018 13:35 ^{UTC-3}	02:00	Final standings	x6483
2018-2019 Russia Open High School Programming Contest (Unrated, Online Mirror, ICPC Rules, Teams Preferred) Enter > Virtual participation >		Dec/09/2018 05:05 ^{UTC-3}	05:00	Final standings	x2159

¹⁰ <http://codeforces.com/blog/entry/967> (Blog CodeForces, 2010)

Em julho de 2018, ocorreu, também, a Microsoft Q# Coding Contest¹¹, em colaboração com o CodeForces, em que a empresa procurava talentos em programação quântica.

Mantendo o foco principal voltado às competições de programação, o CodeForces permite que o estudante visualize as questões de torneios anteriores para treinar e ainda possibilita que sejam consultados os casos de entrada e saída e os códigos aceitos de outros usuários.

Ao exibir a solução de um código, o CodeForces corrobora, inclusive, a ideia deste trabalho, demonstrando que é válido para os alunos aprenderem novas formas de resolver problemas algorítmicos, analisando soluções anteriores.

Tabela 1 – Comparações entre os Ambientes Virtuais

Ambiente	É possível começar a programar?	Organiza os problemas por dificuldade e/ou assunto?	Dá acesso ao código fonte de problemas similares?
UVa Online Judge	Não, os problemas exigem conhecimento prévio de algoritmos.	Utilizando o uHunt, sim.	Não.
URI Online Judge	Sim, há uma seção feita com problemas para iniciantes.	Sim, há divisão clara entre assuntos.	Não.
SPOJ	Sim, é possível mas é preciso procurar para encontrar os mais fáceis.	Não, a divisão é feita apenas por competições.	Não.
CodeForces	Sim, os programadores têm acesso a problemas compatíveis com seu nível no site.	Sim, há um esquema de tags que permite a fácil classificação e seleção dos problemas.	Sim, há a possibilidade de consultar os códigos de grandes programadores em torneios já finalizados.

¹¹ (MICROSOFT QUANTUM TEAM, 2018)

2.2 Novas abordagens no ensino de Algoritmos

No estudo de (OLIVEIRA, 2017), foi proposta uma abordagem de ensino semipresencial de algoritmos apoiada por um juiz corretor de questões *online*, aliado a um ambiente virtual de aprendizagem. Embora parte dos alunos que participaram da pesquisa não se mostraram satisfeitos com a experiência virtual da disciplina pela ausência do professor, a maioria dos estudantes entrevistados considerou o juiz *online* aplicado à turma como uma ferramenta importante pois promoveu e incentivou a prática de exercícios de programação.

Em relação à abordagem de (OLIVEIRA, 2017), o presente estudo certifica-se da importância da figura do professor pois, em consonância ao pensamento de (BERBEL, 2011), é papel do professor ser o intermediador do processo de desenvolvimento, contribuindo, também, para a promoção de autonomia dos alunos.

Neste aspecto, é fundamental que o professor se aproprie de metodologias pedagógicas que facilitem seu aprendizado e se afastem do modelo de ensino técnico-formal e abstrato, característico da disciplina de algoritmos (AVIZ JÚNIOR, 2007).

Segundo (PEREIRA JÚNIOR; RAPKIEWICZ, 2004), o aluno se torna mais interessado em um ambiente descontraído e colaborativo do que em um ambiente instrucionista, no qual o professor preocupa-se apenas em mostrar conhecimento.

Desta forma, as aulas de Algoritmos 2, ministradas pelo professor Carlos de Salles Soares Neto, alternavam-se entre expositivas com demonstração de conceitos e práticas com competições de programação, em uma abordagem construtivista do conhecimento. Esta metodologia será abordada em maiores detalhes na seção 3.1.

O Aprendizado Significativo, do teórico David P. Ausubel, serviu de base para o estudo de (BARBOSA, 2011), aplicando essa metodologia no ensino de algoritmos. (BARBOSA, 2011) esclarece que, além de ser cognitivista por tentar explicar o processo de cognição, o Aprendizado Significativo é também construtivista por assumir que o processo de apreensão de conhecimento é evolutivo, ou seja, é construído com base em etapas e conceitos já conhecidos.

Esta abordagem tem ainda mais relevância ao compararmos com o estudo de (TUAN; CHIN; SHIEH, 2005), a respeito da motivação de estudantes no processo de aprendizado científico. Os autores desenvolveram um questionário que analisa a motivação, através de aspectos inerentes à personalidade e comportamento do aluno como auto-eficácia, objetivo de realização pessoal e estímulo do ambiente de aprendizagem.

Um dos quesitos que influencia a motivação, segundo o questionário desenvolvido por (TUAN; CHIN; SHIEH, 2005), é a utilização de estratégias de aprendizagem

ativa, entre elas, a capacidade de relacionar conceitos novos com experiências e conhecimentos anteriores, assim como preconiza a Teoria do Aprendizado Significativo (AUSUBEL, 1963).

Outro exemplo de abordagem construtivista é a Aprendizagem Baseada em Problemas (PBL)¹², que traz para a sala de aula a interdisciplinaridade, seguindo a concepção de ensino atual, prevista na Lei de Diretrizes e Bases da Educação. Com o PBL, o foco do ensino passa a ser o aluno, que deixa de exercer o papel de receptor passivo das informações transmitidas por seus professores (SOLEDADE, 2018).

Desta forma, este trabalho se propõe a desenvolver com os alunos a construção do conhecimento por Aprendizado Significativo, estimulando-os a estabelecer estratégias de aprendizagem ativa, relacionando conceitos novos (ou seja, a resolução de novos problemas) a conhecimentos prévios (no caso, os códigos-fonte de problemas já resolvidos). Além disto, com base na PBL, este trabalho pretende analisar como os alunos comportam-se em relação ao processo de resolução de problemas.

¹² *Problem-Based Learning*

3 Ferramentas Metodológicas Adotadas

Este capítulo apresenta a metodologia adotada no presente estudo e as ferramentas utilizadas para auxiliar durante o processo.

3.1 Algoritmos 2: um experimento controlado

A disciplina de Algoritmos 2 trata de conceitos avançados de programação e estruturas de dados. No âmbito do curso de Ciência da Computação, na Universidade Federal do Maranhão, faz parte do grupo das eletivas, o que significa que os alunos têm a prerrogativa de escolher se desejam cursá-la, uma vez que a mesma não compõe a grade curricular obrigatória.

A ementa da disciplina é composta pelos seguintes temas: técnicas de desenvolvimento de algoritmos (enumeradores de permutações), dividir e conquistar; backtracking; programação dinâmica; algoritmos gulosos; branch-and-bound; algoritmos probabilísticos; metaheurísticas de busca.

No semestre letivo de 2018.2, o professor Carlos de Salles Soares Neto foi o responsável por ministrar as aulas e organizou a disciplina nos seguintes tópicos: algoritmos ad-hoc, permutações e técnicas de dividir e conquistar, que formaram a primeira nota; em seguida, *backtracking* e programação dinâmica foram os assuntos cobrados na segunda avaliação; e a terceira nota foi relativa ao assunto de Teoria e Algoritmos de Grafos.

Todas as avaliações foram realizadas por meio de torneios (*contests*) realizados na plataforma Judge, que será abordada na seção a seguir. A cada semana, um torneio era criado com questões novas criadas ou adaptadas de outras provenientes do UVA Online Judge. Algumas das questões utilizadas nos torneios desta disciplina compõem o [Apêndice D](#) deste trabalho.

As aulas aconteciam às segundas e quartas-feiras entre 15h50 e 17h20. Às segundas-feiras, as aulas ocorriam em um dos laboratórios de informática do Centro de Ciências Exatas e da Tecnologia da UFMA, o que possibilitava a aplicação dos torneios. Cada aluno teria à sua disposição um computador para praticar os algoritmos tratados em sala, resolvendo questões relativas àquele torneio.

Já às quartas-feiras, a disciplina era ensinada por meio de aulas expositivas em que o professor apresentava o conteúdo, mostrando a aplicação dos algoritmos

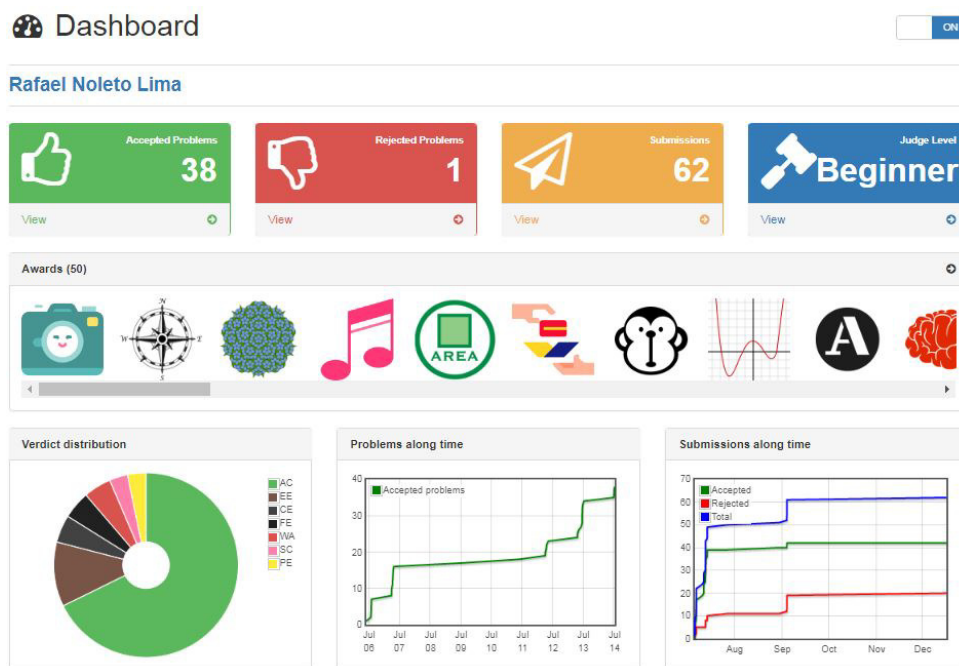
em questões semelhantes e comentando aquelas pertencentes ao torneio.

Um fator motivacional importante, presente na forma em que a disciplina foi conduzida, é a competitividade estimulada em sala de aula. Segundo pesquisa realizada por (IRION; PELEGRINO; BOTELHO, 2016), alunos que participam de competições envolvem-se mais com a disciplina, demonstrando crescimento em seus resultados, além de aprenderem o conteúdo de forma mais dinâmica do que alunos que apenas assistem a aulas expositivas.

3.2 Jutge.org - avaliador online

Conforme mencionado na seção anterior, durante a disciplina, foi utilizado o *site* Jutge.org¹ (Figura 10), que é uma ferramenta educacional para alunos e professores que possui um avaliador online de problemas.

Figura 10 – Página inicial do Jutge



Assim como outras plataformas como o UVa Online Judge e o Sphere Online Judge (SPOJ), o Jutge reúne problemas que devem ser resolvidos por meio do desenvolvimento de um algoritmo. Estes problemas descrevem situações cotidianas e/ou fictícias e o usuário tem acesso a parte de uma série de casos de teste e as respectivas saídas que seu programa deve retornar.

O usuário, então, submete sua solução algorítmica desenvolvida e o juiz online, que possui um conjunto secreto de casos de teste, compara a saída do programa do

¹ (Jutge.org <<https://www.jutge.org/>>, Acesso em: dez/2018)

usuário com a saída que a solução oficial retorna.

Para cada solução submetida, o Judge analisa se a submissão apresentada pelo aluno soluciona corretamente o problema proposto, confrontando com a saída da solução oficial, além de avaliar quesitos técnicos como o tempo de execução do algoritmo e a memória utilizada, que devem respeitar os limites definidos pela questão. A [Tabela 2](#) traz os vereditos emitidos pelo juiz online do Judge com as respectivas explicações.

Tabela 2 – Possíveis vereditos emitidos pelo Judge

Sigla	Significado	Explicação
AC	Accepted	Resposta Aceita: a resposta que o programa do usuário retorna é compatível com a solução oficial.
WA	Wrong Answer	Resposta errada: a resposta que o programa do usuário retorna não é compatível com a solução oficial.
EE	Execution Error	Erro em Tempo de Execução: o programa do usuário apresenta um erro durante a execução, como, por exemplo, acesso inválido de memória ou divisão por zero.
PE	Presentation Error	Erro de Apresentação: a resposta que o programa do usuário retorna pode (ou não) estar totalmente correta, porém apresenta espaços ou quebras de linha a mais.
TLE	Time Limit Exceeded	Tempo Limite Excedido: o processamento do programa do usuário ultrapassa o limite de tempo estabelecido pela questão.
MLE	Memory Limit Exceeded	Memória Limite Excedida: o processamento do programa do usuário ultrapassa o limite de memória estabelecido pela questão.

O Judge possui dois tipos de perfil de usuário: estudante e professor. A [Figura 10](#) apresenta a tela inicial do Judge após o login de um usuário do tipo estudante. Nela, há informações de quantos problemas o aluno já resolveu, quantas submissões foram aceitas e rejeitadas e qual seu nível atual no site. Além disso, há gráficos que mostram a evolução do aluno em relação aos problemas aceitos e rejeitados com o passar do tempo.

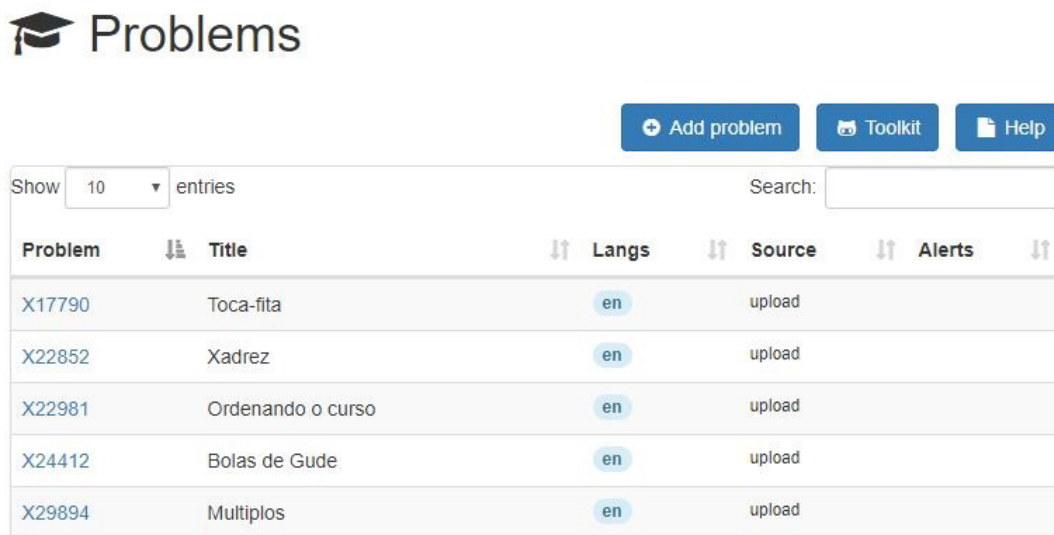
O usuário do tipo professor tem acesso a funcionalidades específicas no site.

Além de criar turmas, incluir seus alunos e realizar atividades envolvendo os problemas que estão disponíveis no site, o perfil de professor também pode redigir novos problemas e adicioná-los ao Jutge.

3.2.1 Como criar um problema no Jutge

O ambiente Jutge possui um repositório de questões demonstrativas na plataforma Github, com documentação bem definida e que auxilia no processo da criação de novos problemas. A maneira mais simples e rápida de inserir um problema é criar cada um dos arquivos que o *Jutge* exige, compactar tudo em uma pasta zipada e realizar o upload na seção correspondente.

Figura 11 – Página de problemas do *Jutge*



Problem	Title	Langs	Source	Alerts
X17790	Toca-fita	en	upload	
X22852	Xadrez	en	upload	
X22981	Ordenando o curso	en	upload	
X24412	Bolas de Gude	en	upload	
X29894	Multiplos	en	upload	

A [Figura 11](#) mostra o módulo do Jutge relativo aos problemas em um perfil do tipo professor (*instructor*). Além da lista com aqueles que o próprio professor já incluiu, há um ícone chamado “*Add problem*”, no qual novos problemas podem ser adicionados. Além da possibilidade da pasta zipada, a inclusão também pode ser feita por meio de links no Dropbox ou no Github.


No [Apêndice C](#), estão descritos os arquivos necessários para a criação de um problema no Jutge, segundo sua própria documentação:

3.2.2 Como criar um torneio no Jutge

A [Figura 12](#), na página seguinte, mostra a seção *Exams/Contests*. A forma de criar os testes (*Exams*) e torneios (*Contests*) no Jutge é a mesma. A diferença entre as duas modalidades de avaliação é que o torneio estabelece um *ranking* dos alunos,

de acordo com a velocidade em que resolveram as questões, de forma análoga às disputas de maratonas de programação.

Figura 12 – Página de torneios (*Exams/Contests*) do Judge

 Exams [+ Add exam](#)

Show entries Search:

Title	Expected start time	Course	Type	Visible submissions
Grafos #3 (10/12/2018)	2018-12-11 06:31:26	Algoritmos 2 - Ufma 2018.2	Contest	Yes
Grafos #1 (26/11/2018)	2018-11-27 10:00:00	Algoritmos 2 - Ufma 2018.2	Contest	Yes
Programação Dinâmica #2 (12/11/2018)	2018-11-12 03:30:00	Algoritmos 2 - Ufma 2018.2	Contest	Yes
Programação Dinâmica #1 (05/11/2018)	2018-11-05 03:30:00	Algoritmos 2 - Ufma 2018.2	Contest	Yes
Avaliação 2 - Divisão e Conquista	2018-10-09 02:00:00	Algoritmos 2 - Ufma 2018.2	Contest	Yes
Exercício 3 - Divisão e Conquista	2018-09-25 02:00:00	Algoritmos 2 - Ufma 2018.2	Contest	Yes
Exercício 2 - Soma de inteiros	2018-09-11 07:00:26	Algoritmos 2 - Ufma 2018.2	Contest	Yes
Avaliação 1 - Ad hoc	2018-09-04 17:00:00	Algoritmos 2 - Ufma 2018.2	Contest	Yes
Exercício 1 - Ad Hoc	2018-08-30 12:00:00	Algoritmos 2 - Ufma 2018.2	Contest	Yes

Showing 1 to 9 of 9 entries [Previous](#) [1](#) [Next](#)


Ao selecionar a opção "Add exam", o professor deve preencher o formulário da [Figura 13](#) com os dados gerais do torneio, como Id, Título, Horário de Início, Duração e Compiladores permitidos, entre outros.

Após a criação do torneio com os dados presentes na [Figura 13](#), devemos inserir os problemas e adicionar os estudantes que participarão. Nas maratonas de programação, cada problema é representado por um balão colorido. O Judge utiliza-se dessa característica das maratonas e permite que cada problema seja simbolizado por um ícone. No total, há 974 ícones disponíveis incluindo balões, doces, letras, números e até Pokémons, conforme [Figura 14](#).

Um detalhe sobre estes ícones é que eles brilham quando o usuário resolve corretamente uma questão. Este aspecto lúdico presente no Judge pode se tornar um atrativo especial para o ensino de algoritmos para crianças, assemelhando-se a plataformas baseadas em jogos como o *site* Hour of Code², utilizado com sucesso no trabalho de (CARLOS; GODINHO; GOMIDE, 2018).

² (Hour of Code <<https://hourofcode.com/>>, Acesso em: dez/2018)

Figura 13 – Formulário de criação de um torneio no Jutge

 Add exam


Id

Title

Place

Description Preview

Instructions Preview

Expected start time 

Running time (minutes)

Contest mode No

Anonymous contest No

Figura 14 – Ícones que podem representar uma questão no Jutge




3.2.3 Uso de torneios no Jutge na disciplina Algoritmos 2

Utilizar os torneios do Jutge em sala de aula foi uma experiência semelhante a simular a dinâmica realizada durante maratonas de programação. Nas maratonas de programação, times compostos por três estudantes tentam resolver durante 5 horas

o maior número possível dos 10 ou mais problemas que são entregues no início da competição. Estes estudantes têm à sua disposição apenas um computador e material impresso (livros, listagens, manuais) para vencer a batalha contra o relógio e os problemas propostos.

Assim, a cada semana, nas aulas de segunda-feira no Laboratório de Informática, os alunos receberam um desafio contendo entre dois a quatro problemas cuja solução deveria ser submetida por meio do Jutge, que corrigiria as questões e emitiria um dos vereditos possíveis, conforme descritos na [Tabela 2](#). Abaixo, a [Figura 15](#) mostra uma lista de submissões dos alunos durante um torneio realizado na turma.

Figura 15 – Lista de submissões em um torneio no Jutge

 Submissions

Problem	User	Verdict	Compiler	Time in
X64437_en	Jorge Lucas Silva Cavalcante	EE invalid memory reference	GCC	2018-12-01 19:47:49
X58731_en	Welton Marinho de Souza	AC	GCC	2018-12-01 19:28:34
X64437_en	Jorge Lucas Silva Cavalcante	EE invalid memory reference	GCC	2018-12-01 19:18:04
X58731_en	Welton Marinho de Souza	PE	GCC	2018-12-01 19:06:14
X23854_en	Pedro Thiago Cutrim dos Santos	AC	GCC	2018-12-01 17:36:46
X64437_en	Jorge Lucas Silva Cavalcante	EE invalid memory reference	GCC	2018-12-01 17:05:12
X61944_en	Pedro Thiago Cutrim dos Santos	AC	GCC	2018-12-01 16:24:35
X58731_en	Luis Alves dos Santos Neto	AC	GCC	2018-12-01 16:14:45
X58731_en	Luis Alves dos Santos Neto	PE	GCC	2018-12-01 16:06:36
X58731_en	Luis Alves dos Santos Neto	EE invalid memory reference	GCC	2018-12-01 16:00:56

Showing 41 to 50 of 122 entries

Previous 1 ... 4 **5** 6 ... 13 Next

Esta simulação de maratona estimula os alunos a treinarem a rapidez de raciocínio para resolução dos problemas, além de incentivar o aumento na velocidade em que programam, visto que a pontuação de cada resposta correta é proporcional ao tempo que levaram para resolver. Quanto mais rápido a questão foi respondida, maior a pontuação do aluno.

Embora outros sites como o CodeForces e o A2OJ sejam amplamente conhecidos e recomendados para a criação de disputas online, alguns fatores foram fundamentais para a aplicação do Jutge na disciplina de Algoritmos 2. Enquanto aqueles sites utilizam problemas de seus arquivos e também de outros repositórios, o Jutge permite a criação de problemas novos, o que garante que uma questão adicionada ao Jutge seja inédita, considerando aquele contexto.

Além disso, soluções de problemas de juizes renomados como o UVa Online Judge são facilmente encontrados em uma busca pela *internet*. Sendo assim, se as avaliações fossem feitas com base nessas questões, não seria possível prevenir que o aluno burlasse o processo de aprendizado, utilizando códigos prontos que não fossem de sua autoria. No [Apêndice D](#), há exemplos de questões adaptadas para o Judge.

Outro fator importante é que, dentro de um torneio organizado pelo professor no Judge, ele pode ter acesso aos códigos submetidos pelos alunos para resolução de cada questão. Estes códigos possuem grande poder didático. O professor poderia, por exemplo, resguardando o sigilo de cada submissão, discutir em sala de aula, em conjunto com os alunos, qual o motivo de determinado código não retornar a resposta correta, ou analisar um algoritmo enviado e verificar porque está falhando em certos casos de teste, estimulando a participação da turma numa abordagem construtivista da disciplina.

3.3 Blog codigosufma

Para auxiliar os alunos durante os torneios organizados pelo Judge, foi desenvolvido o *blog* <<http://codigosufma.online>>, onde foram disponibilizados códigos que solucionavam corretamente problemas presentes no UVa Online Judge.

A [Figura 16](#) mostra a página principal do *blog*, que foi desenvolvido com base no WordPress, um sistema livre de gestão de conteúdo para internet, baseado em PHP com banco de dados SQL, atualmente uma das ferramentas mais utilizadas para conteúdo online³. Como forma de contribuição para o meio acadêmico, o código-fonte da implementação do *site* poderá vir a ser disponibilizado para consultas.

Figura 16 – Página principal do blog codigosufma



A cada semana, novas soluções foram adicionadas à página de acordo com os assuntos estudados em sala de aula e cobrados no respectivo torneio. Dessa forma, os alunos utilizariam este repositório de códigos como fonte de estudos para se familiarizar com os algoritmos e a utilização destes no contexto da resolução de questões agrupadas por assunto, conforme visto na [Figura 17](#).

Para controlar o acesso às soluções, foram geradas senhas individuais para cada aluno. Cada vez que um estudante visitou o site e selecionou um dos códigos,

³ (Wordpress <<https://pt.wikipedia.org/wiki/WordPress>>, Acesso em: dez/2018)

Figura 17 – Página do blog codigosufma exibindo as soluções disponíveis

Seja bem-vindo, Rafael Noletto Lima.

Selecione um dos códigos abaixo para consultar:

BFS - Busca em Largura

- ◉ 10004 - Bicolorable [\[UVA\]](#)
- ◉ 10067 - Playing with Wheels [\[UVA\]](#)
- ◉ 10653 - Bombs! NO they are Mines!! [\[UVA\]](#)
- ◉ 383 - Shipping Routes [\[UVA\]](#)
- ◉ 417 - Word Index [\[UVA\]](#)
- ◉ 429 - Word Transformation [\[UVA\]](#)

DFS - Busca em Profundidade

- ◉ 10009 - All Roads Lead Where? [\[UVA\]](#)
- ◉ 11838 - Come and Go [\[UVA\]](#)
- ◉ 260 - Il Gioco dell'X [\[UVA\]](#)

Caminho Minimo - Dijkstra

- ◉ 341 - Non-Stop Travel [\[UVA\]](#)

Minimum Spanning Tree

- ◉ 908 - Re-connecting Computer Sites [\[UVA\]](#)

Shortest Paths - Floyd-Warshall

- ◉ 1112 - Mice and Maze [\[UVA\]](#)

Ordenacao Topologica

- ◉ 10305 - Ordering Tasks [\[UVA\]](#)

o sistema salvou em seu banco de dados um *log* contendo as informações do aluno, o horário de visita ao site e o código visualizado, em tabelas SQL, conforme será explorado mais adiante na [seção 4.3](#).

Desta forma, foi possível coletar dados importantes acerca da utilização dos estudantes ao blog como, por exemplo, quais os tipos de problemas mais visitados, qual o assunto que os alunos mais tiveram interesse e em que época da disciplina a página foi mais visitada. Estes dados serão apresentados no próximo Capítulo.

4 Dados Coletados

Neste Capítulo são apresentados os dados coletados com o formulário baseado no estudo de avaliação de motivação de (TUAN; CHIN; SHIEH, 2005). Além disto, são exibidos os *logs* de acesso dos usuários ao *blog* *codigosufma*¹.

4.1 Formulário de Avaliação de Motivação

Antes da divulgação do *blog* aos alunos da disciplina de Algoritmos 2, foi aplicado, com 9 dos 11 alunos matriculados, um formulário de avaliação de motivação no dia 26/11/2018. A ideia era descobrir como estavam as impressões da turma em relação à disciplina, à utilização do Jutge e o que eles achavam da ideia de ter um repositório de códigos à disposição.

O formulário original (TUAN; CHIN; SHIEH, 2005) contava com 35 (trinta e cinco) afirmações que foram adaptadas para o contexto da disciplina de Algoritmos 2. Ao final, foram adicionadas 5 (cinco) afirmações relativas ao Jutge e ao *blog*, totalizando 40 (quarenta) afirmações. Para cada afirmação, o aluno deveria responder em que grau concorda ou discorda do que está sendo afirmado, dentre uma escala de 1 a 5, conforme tabela abaixo.

Tabela 3 – Classificação de Respostas ao Formulário de Motivação

Nota	Código	Significado
1	DT	Discordo Totalmente
2	DP	Discordo Parcialmente
3	NTO	Não Tenho Opinião
4	CP	Concordo Parcialmente
5	CT	Concordo Totalmente

As 35 primeiras afirmações constantes do formulário, derivadas de (TUAN; CHIN; SHIEH, 2005), estão agrupadas em tópicos, conforme as características que influenciam a motivação do aluno, quais sejam: a autoeficácia, o uso de estratégias de aprendizagem ativa, o valor do aprendizado da disciplina, os objetivos de desempenho e de realização pessoal do aluno e o estímulo do ambiente de aprendizagem. As cinco afirmações adicionadas refletem as expectativas do aluno em relação ao ambiente de aprendizagem, no caso, o Jutge e o *blog* *codigosufma*.

O formulário completo, entregue aos alunos, está disponível no [Apêndice A](#).

¹ (Blog "*codigosufma*"<<http://codigosufma.online/>>, Acesso em: dez/2018)

4.2 Respostas da primeira aplicação do formulário

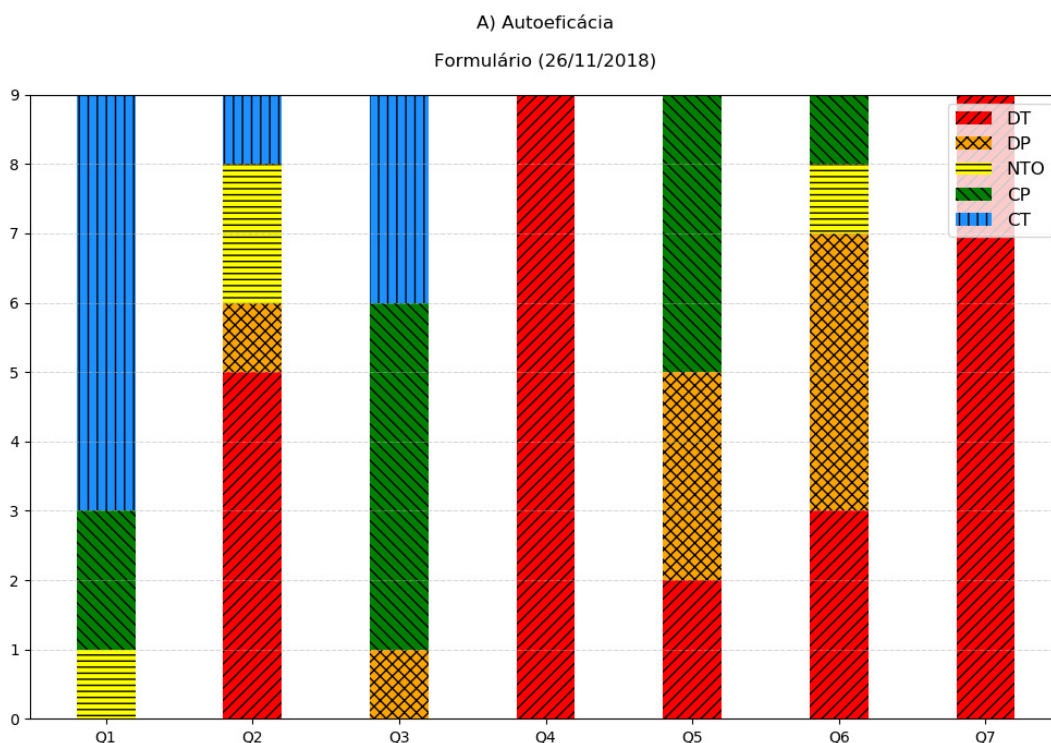
Nesta seção, são apresentadas, em sequência, as afirmações correspondentes a cada tópico e os gráficos que representam a quantidade de respostas dos alunos para cada opção.

4.2.1 Autoeficácia

Tabela 4 – Afirmações relacionadas à Autoeficácia

Autoeficácia
Q1. Independente de o conteúdo da disciplina de Algoritmos 2 ser difícil ou fácil, eu tenho certeza que consigo compreendê-lo.
Q2. Eu não me sinto confiante para compreender conceitos difíceis da disciplina de Algoritmos 2.
Q3. Eu tenho certeza de que posso ir bem nos testes de Algoritmos 2.
Q4. Não importa o quanto eu me esforce, eu não consigo aprender os conceitos de Algoritmos 2.
Q5. Quando os problemas de Algoritmos 2 são muito difíceis, eu desisto ou apenas faço as partes fáceis.
Q6. Durante os testes de Algoritmos 2, eu prefiro pedir ajuda para outras pessoas ao invés de pensar por mim mesmo.
Q7. Quando eu acho um conteúdo de Algoritmos 2 difícil, eu nem tento aprendê-lo.

Figura 18 – Respostas às afirmações relacionadas à Autoeficácia

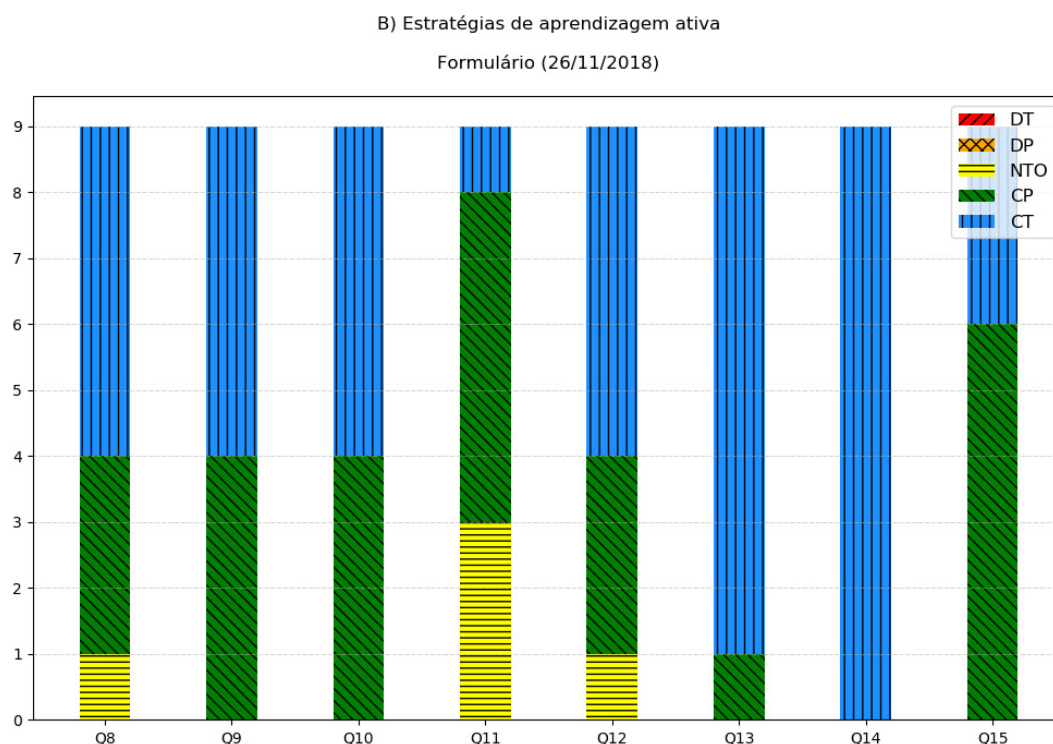


4.2.2 Estratégias de Aprendizagem Ativa

Tabela 5 – Afirmações relacionadas às Estratégias de Aprendizagem Ativa

Estratégias de Aprendizagem Ativa
Q8. Quando eu aprendo novos conceitos de algoritmos, eu me esforço para compreendê-lo.
Q9. Quando eu aprendo novos conceitos de Algoritmos 2, eu os relaciono com minhas experiências anteriores.
Q10. Quando eu não compreendo um conceito de Algoritmos 2, eu procuro meios relevantes que possam me ajudar.
Q11. Quando eu não compreendo um conceito de Algoritmos 2, eu procuro discutir com o professor ou outros colegas para facilitar minha compreensão.
Q12. Durante o processo de aprendizagem, eu tento fazer conexões entre os conceitos que aprendo.
Q13. Quando eu cometo um erro em um algoritmo, eu tento descobrir o porquê.
Q14. Quando encontro conceitos de Algoritmos 2 que eu não compreendo, ainda assim, me esforço para aprendê-los.
Q15. Quando novos conceitos que eu aprendo entram em conflito com meus conhecimentos anteriores, eu tento compreender o porquê.

Figura 19 – Respostas às afirmações relacionadas à Estratégias de Aprendizagem Ativa

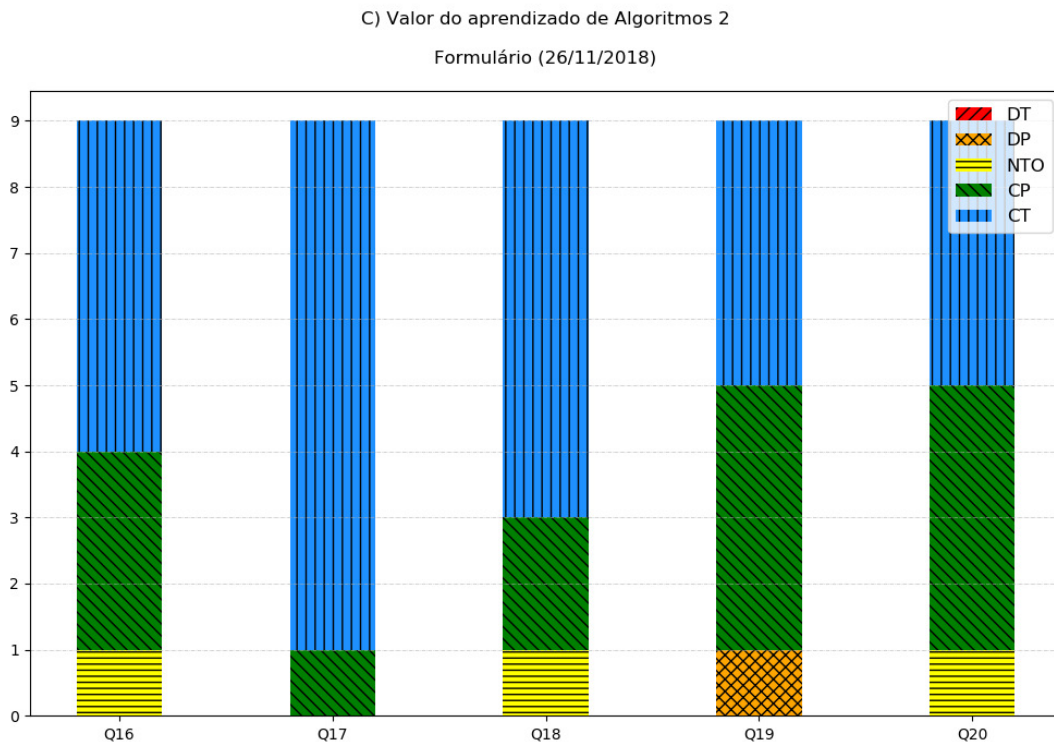


4.2.3 Valor do Aprendizado de Algoritmos 2

Tabela 6 – Afirmações relacionadas ao Valor do Aprendizado de Algoritmos 2

Valor do Aprendizado de Algoritmos 2
16. Eu penso que aprender os conceitos de Algoritmos 2 é importante porque eu posso usá-los no meu dia a dia.
17. Eu penso que aprender os conceitos de Algoritmos 2 é importante porque estimula meu raciocínio.
18. Em Algoritmos 2, eu penso que é importante aprender a resolver problemas práticos.
19. Em Algoritmos 2, eu penso que é importante participar de atividades de desenvolvimento em grupo.
20. Enquanto estou aprendendo Algoritmos 2, é importante conseguir satisfazer minha própria curiosidade.

Figura 20 – Respostas às afirmações relacionadas ao Valor do Aprendizado de Algoritmos 2

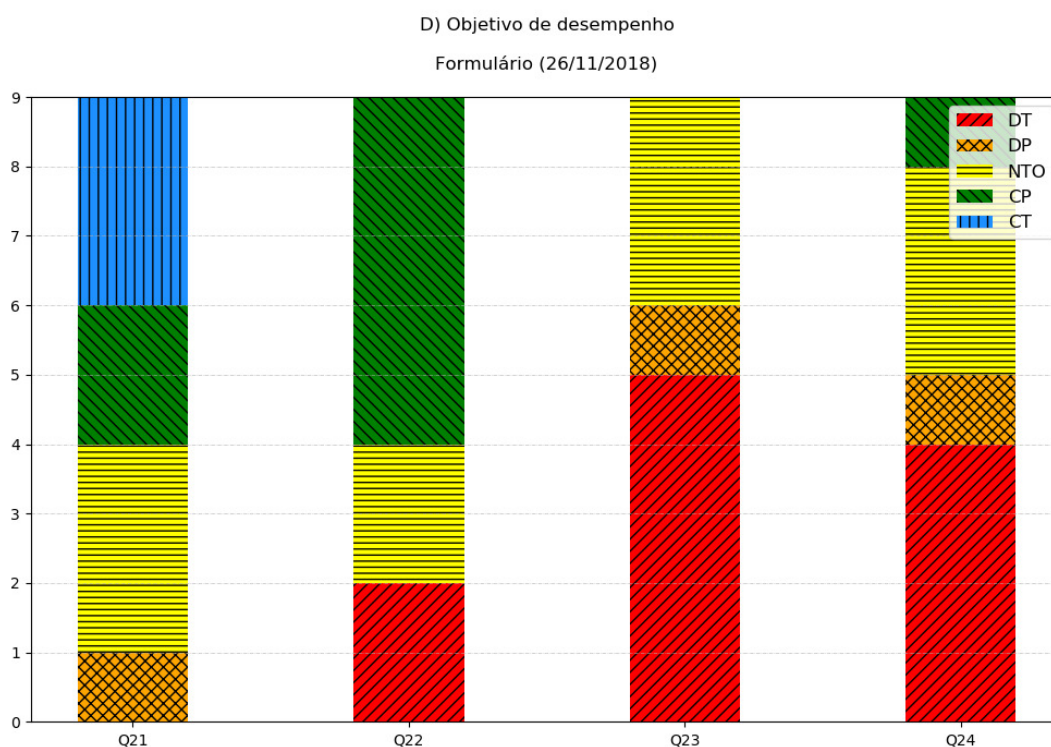


4.2.4 Objetivo de desempenho

Tabela 7 – Afirmações relacionadas ao Objetivo de desempenho

Objetivo de desempenho
21. Eu participo dos testes de Algoritmos 2 para ter uma boa nota.
22. Eu participo dos testes de Algoritmos 2 para ter melhor desempenho que outros estudantes.
23. Eu participo dos testes de Algoritmos 2 para que outros estudantes pensem que eu sou inteligente.
24. Eu participo dos testes de Algoritmos 2 para que o professor preste atenção em mim.

Figura 21 – Respostas às afirmações relacionadas ao Objetivo de desempenho

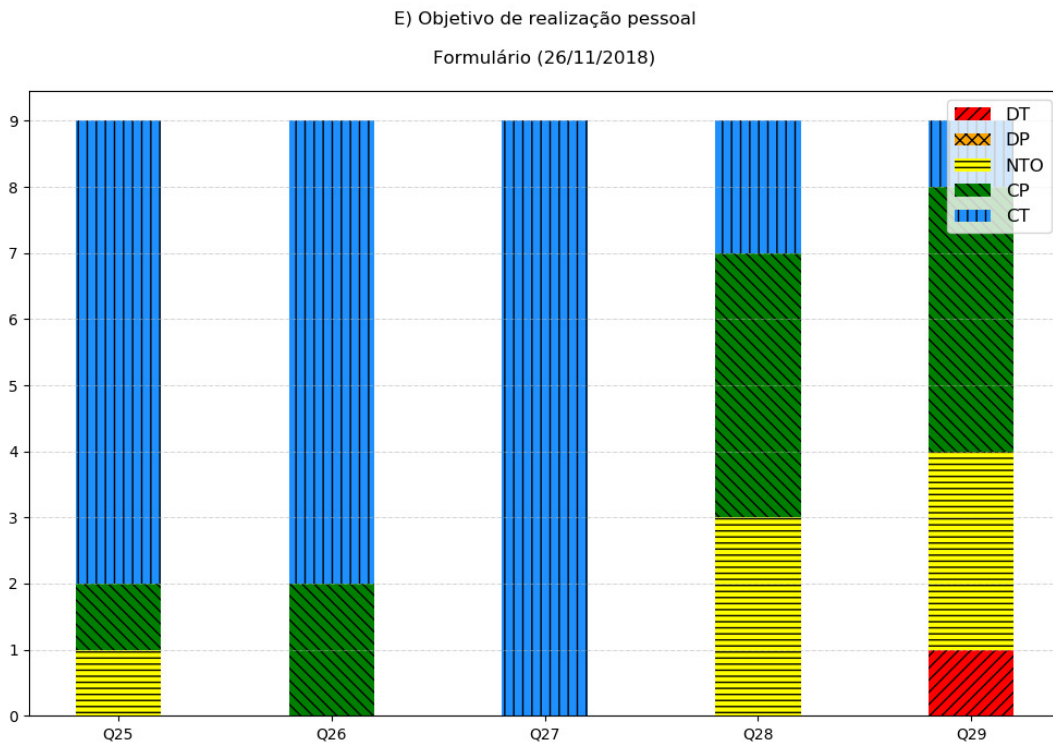


4.2.5 Objetivo de realização pessoal

Tabela 8 – Afirmações relacionadas ao Objetivo de realização pessoal

Objetivo de realização pessoal
25. Durante um teste de Algoritmos 2, eu me sinto mais realizado quando obtenho uma boa pontuação.
26. Durante um teste de Algoritmos 2, eu me sinto mais realizado quando me sinto confiante sobre o conteúdo.
27. Durante um teste de Algoritmos 2, eu me sinto mais realizado quando sou capaz de solucionar um problema difícil.
28. Durante uma aula de Algoritmos 2, eu me sinto mais realizado quando o professor aceita minhas ideias.
29. Durante uma aula de Algoritmos 2, eu me sinto mais realizado quando outros estudantes aceitam minhas ideias.

Figura 22 – Respostas às afirmações relacionadas ao Objetivo de realização pessoal

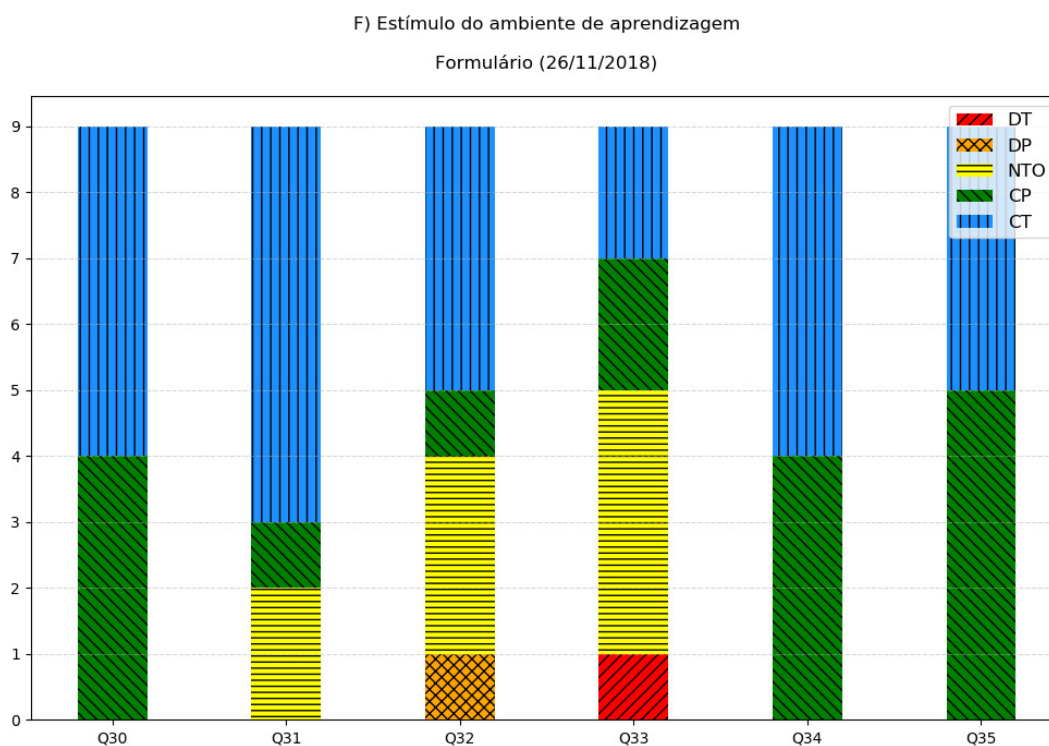


4.2.6 Estímulo do ambiente de aprendizagem

Tabela 9 – Afirmações relacionadas ao Estímulo do ambiente de aprendizagem

Estímulo do ambiente de aprendizagem
30. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque o conteúdo é empolgante e dinâmico.
31. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque o professor usa diferentes estratégias de ensino.
32. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque o professor não me pressiona muito.
33. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque o professor presta atenção em mim.
34. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque os problemas são desafiadores.
35. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque a turma participa de debates.

Figura 23 – Respostas às afirmações relacionadas ao Estímulo do ambiente de aprendizagem



4.2.7 Expectativas quanto ao ambiente de aprendizagem (Jutge)

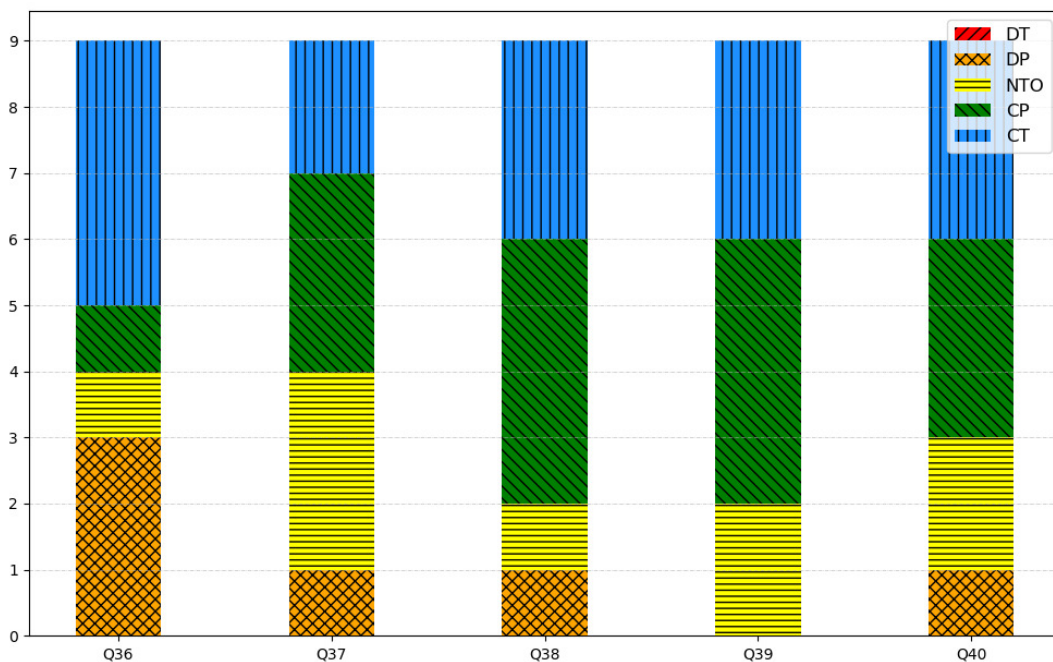
Tabela 10 – Afirmações relacionadas às Expectativas quanto ao ambiente de aprendizagem

Expectativas quanto ao ambiente de aprendizagem (Jutge)
36. Eu gosto de fazer os testes realizados na plataforma Jutge.org.
37. Os testes realizados na plataforma Jutge.org me motivaram a estudar mais.
38. Durante os testes de Algoritmos 2, eu me sinto motivado mesmo quando não entendo os conceitos referentes ao problema.
39. Durante os testes de Algoritmos 2, eu me sinto motivado quando encontro a solução de um problema semelhante.
40. Durante os testes de Algoritmos 2, eu me sentirei mais motivado, se tiver acesso a soluções de problemas semelhantes.

Figura 24 – Respostas às afirmações relacionadas às Expectativas quanto ao ambiente de aprendizagem

G) Expectativas quanto ao ambiente de aprendizagem (Jutge)

Formulário (26/11/2018)



4.3 Dados de utilização do *blog* codigosufma pelos alunos

Logo após a primeira aplicação do formulário em 26/11/2018, todos os 11 alunos matriculados na disciplina receberam por e-mail uma senha individual de acesso ao *blog* codigosufma² e puderam iniciar a explorar os códigos-fonte lá presentes, a fim de ajudá-los diante dos torneios relativos ao assunto Grafos. A cada acesso ao *site* ou consulta a código, eram salvos *logs* no banco de dados com as informações do nome do aluno, horário e página que foi visitada.

Dessa forma, ao fazer consultas na linguagem SQL, foi possível extrair informações interessantes, resultando nas tabelas a seguir. Por questões de privacidade, os alunos serão referenciados apenas através do ID que receberam no sistema. O resultado de uma busca completa³ no banco de dados com todos os logs pode ser encontrado no [Apêndice B](#).

4.3.1 Quantas vezes os alunos entraram no *site* e visualizaram a Lista de Códigos?

Tabela 11 – Visualização da Lista de Códigos

4

ID_logs	data_logs	ID_aluno	link_post
1	2018-11-26 16:59:33	9	Lista de Codigos
3	2018-11-26 17:01:29	13	Lista de Codigos
9	2018-11-26 17:23:14	12	Lista de Codigos
15	2018-11-26 21:41:34	9	Lista de Codigos
18	2018-11-28 18:23:45	7	Lista de Codigos
19	2018-11-28 21:32:41	7	Lista de Codigos
22	2018-11-28 22:04:29	12	Lista de Codigos
29	2018-11-29 09:57:27	7	Lista de Codigos
31	2018-11-29 22:13:45	11	Lista de Codigos
33	2018-11-30 12:36:48	7	Lista de Codigos
36	2018-12-01 18:16:11	7	Lista de Codigos
39	2018-12-02 11:42:59	12	Lista de Codigos
45	2018-12-03 16:46:33	11	Lista de Codigos
47	2018-12-03 16:52:37	9	Lista de Codigos
50	2018-12-03 17:29:50	12	Lista de Codigos
63	2018-12-04 13:23:22	9	Lista de Codigos

² (Blog "codigosufma" <<http://codigosufma.online/>>, Acesso em: dez/2018)

³ SELECT ID_logs, data_logs, ID_aluno, link_post FROM userlogs AS l, codigos AS c, userpass AS u WHERE l.fk_ID_aluno = u.ID_aluno AND l.fk_filename_cods = c.filename_cods ORDER BY l.ID_logs ASC

Tabela 11 – Visualização da Lista de Códigos (continuação)

ID_logs	data_logs	ID_aluno	link_post
65	2018-12-04 17:09:03	9	Lista de Codigos
67	2018-12-04 17:26:13	9	Lista de Codigos
69	2018-12-05 00:05:14	4	Lista de Codigos
71	2018-12-05 11:37:37	4	Lista de Codigos
73	2018-12-05 16:01:33	4	Lista de Codigos
75	2018-12-05 18:09:49	11	Lista de Codigos
77	2018-12-07 03:25:35	4	Lista de Codigos
78	2018-12-07 16:28:14	13	Lista de Codigos
82	2018-12-07 18:51:27	4	Lista de Codigos
84	2018-12-07 19:35:13	4	Lista de Codigos
86	2018-12-17 15:19:05	12	Lista de Codigos

Foram registrados 27 acessos. Os alunos que entraram no site mais vezes foram, em ordem decrescente: Aluno4 e Aluno9 com 6 acessos; Aluno7 e Aluno12, com 5 acessos cada; Aluno11, com 3 acessos e Aluno13, com 2 acessos.

4.3.2 Quantas vezes os alunos entraram no *site* e visualizaram um Código-Fonte?

Tabela 12 – Visualização de Código-Fonte

5

ID_logs	data_logs	ID_aluno	link_post
2	2018-11-26 16:59:42	9	10004 - Bicolorable (BFS)
4	2018-11-26 17:01:51	13	1112 - Mice and Maze (Shortest-Paths)
5	2018-11-26 17:02:07	13	10004 - Bicolorable (BFS)
6	2018-11-26 17:02:47	9	10067 - Playing with Wheels (BFS)
7	2018-11-26 17:03:15	9	908 - Re-connecting Computer Sites (MST)
8	2018-11-26 17:03:17	13	10067 - Playing with Wheels (BFS)
10	2018-11-26 17:32:31	13	10653 - Bombs! NO they are Mines!! (BFS)
11	2018-11-26 17:32:44	13	429 - Word Transformation (BFS)
12	2018-11-26 17:33:02	13	10067 - Playing with Wheels (BFS)
13	2018-11-26 17:33:08	13	11838 - Come and Go (DFS)
14	2018-11-26 17:33:34	13	10009 - All Roads Lead Where? (DFS)
16	2018-11-26 21:41:43	9	10067 - Playing with Wheels (BFS)
17	2018-11-26 21:42:26	9	429 - Word Transformation (BFS)

Tabela 12 – Visualização de Código-Fonte (continuação)

ID_logs	data_logs	ID_aluno	link_post
20	2018-11-28 21:35:10	7	10009 - All Roads Lead Where? (DFS)
21	2018-11-28 21:35:32	7	10004 - Bicolorable (BFS)
23	2018-11-28 22:04:40	12	10067 - Playing with Wheels (BFS)
24	2018-11-28 22:05:03	12	10004 - Bicolorable (BFS)
25	2018-11-28 22:05:20	12	10653 - Bombs! NO they are Mines!! (BFS)
26	2018-11-28 22:05:37	12	383 - Shipping Routes (BFS)
27	2018-11-28 22:05:58	12	417 - Word Index (BFS)
28	2018-11-28 22:06:10	12	429 - Word Transformation (BFS)
30	2018-11-29 09:57:48	7	1112 - Mice and Maze (Shortest-Paths)
32	2018-11-29 22:14:09	11	10004 - Bicolorable (BFS)
34	2018-11-30 12:47:23	7	10653 - Bombs! NO they are Mines!! (BFS)
35	2018-11-30 12:48:21	7	10336 - Rank the Languages (FF)
37	2018-12-01 18:16:48	7	908 - Re-connecting Computer Sites (MST)
38	2018-12-01 18:17:22	7	10653 - Bombs! NO they are Mines!! (BFS)
40	2018-12-02 11:43:20	12	10004 - Bicolorable (BFS)
41	2018-12-02 11:45:52	12	908 - Re-connecting Computer Sites (MST)
42	2018-12-02 12:08:56	12	11838 - Come and Go (DFS)
43	2018-12-02 12:09:00	12	10009 - All Roads Lead Where? (DFS)
44	2018-12-02 12:09:07	12	429 - Word Transformation (BFS)
46	2018-12-03 16:46:50	11	10305 - Ordering Tasks (TOPSORT)
48	2018-12-03 16:52:45	11	10305 - Ordering Tasks (TOPSORT)
49	2018-12-03 16:52:47	9	10305 - Ordering Tasks (TOPSORT)
51	2018-12-03 17:29:58	12	10305 - Ordering Tasks (TOPSORT)
52	2018-12-03 17:30:07	12	1112 - Mice and Maze (Shortest-Paths)
53	2018-12-03 17:30:11	12	908 - Re-connecting Computer Sites (MST)
54	2018-12-03 17:30:18	12	572 - Oil Deposits (FF)
55	2018-12-03 17:30:29	12	469 - Wetlands of Florida (FF)
56	2018-12-03 17:30:36	12	10336 - Rank the Languages (FF)
57	2018-12-03 17:30:41	12	341 - Non-Stop Travel (DIJ)
58	2018-12-03 17:30:54	12	11838 - Come and Go (DFS)
59	2018-12-03 17:31:00	12	10009 - All Roads Lead Where? (DFS)
60	2018-12-03 17:31:05	12	429 - Word Transformation (BFS)
61	2018-12-03 17:46:31	12	417 - Word Index (BFS)
62	2018-12-03 17:46:40	12	10305 - Ordering Tasks (TOPSORT)
64	2018-12-04 13:23:33	9	341 - Non-Stop Travel (DIJ)

Tabela 12 – Visualização de Código-Fonte (continuação)

ID_logs	data_logs	ID_aluno	link_post
66	2018-12-04 17:09:11	9	341 - Non-Stop Travel (DIJ)
68	2018-12-04 17:26:24	9	341 - Non-Stop Travel (DIJ)
70	2018-12-05 00:05:30	4	10009 - All Roads Lead Where? (DFS)
72	2018-12-05 11:40:47	4	10305 - Ordering Tasks (TOPSORT)
74	2018-12-05 16:01:44	4	341 - Non-Stop Travel (DIJ)
76	2018-12-05 18:10:17	11	341 - Non-Stop Travel (DIJ)
79	2018-12-07 16:28:42	13	10067 - Playing with Wheels (BFS)
80	2018-12-07 16:28:57	13	341 - Non-Stop Travel (DIJ)
81	2018-12-07 16:29:14	13	10004 - Bicolorable (BFS)
83	2018-12-07 18:51:49	4	10305 - Ordering Tasks (TOPSORT)
85	2018-12-07 19:35:56	4	10305 - Ordering Tasks (TOPSORT)
87	2018-12-17 15:19:32	12	10067 - Playing with Wheels (BFS)

Agrupando os 60 resultados em uma nova consulta⁶, temos:

Tabela 13 – Quantidade de códigos acessados por aluno

ID_aluno	ContCodigosPorAluno
12	24
13	11
9	9
7	7
4	5
11	4

4.4 Análise entre as duas aplicações do formulário

Antes do término da disciplina de Algoritmos 2, no dia 17/12/2018, algumas semanas após a divulgação do *blog* codigosufma⁷, foi solicitado aos alunos que respondessem a uma versão atualizada daquele formulário aplicado anteriormente. Dos 9 alunos que responderam na primeira ocasião, apenas 3 participaram da reaplica-

⁶ SELECT u.ID_aluno, Count(u.ID_aluno) AS ContCodigosPorAluno FROM userlogs AS l, codigos AS c, userpass AS u WHERE l.fk_ID_aluno = u.ID_aluno AND l.fk_filename_cods = c.filename_cods AND c.fk_tipo_cods != 'LOGIN' GROUP BY u.ID_aluno ORDER BY ContCodigosPorAluno DESC

⁷ ([Blog "codigosufma"](http://codigosufma)<<http://codigosufma.online/>>, Acesso em: dez/2018)

ção do formulário. Serão analisados os resultados desse estudo, qualitativamente, utilizando-se os dados destes três formulários. Estes alunos serão identificados por um identificador secreto, o ID a qual pertence no Banco de Dados do *blog*, portanto os gráficos são apresentados na ordem: Aluno2, Aluno3 e Aluno9.

4.4.1 Respostas subjetivas sobre o Jutge e o *blog*

Além das 40 afirmações, os alunos tiveram a oportunidade de responder duas questões subjetivas a respeito do Jutge e do *blog*. Estas foram suas respostas sobre o Jutge:

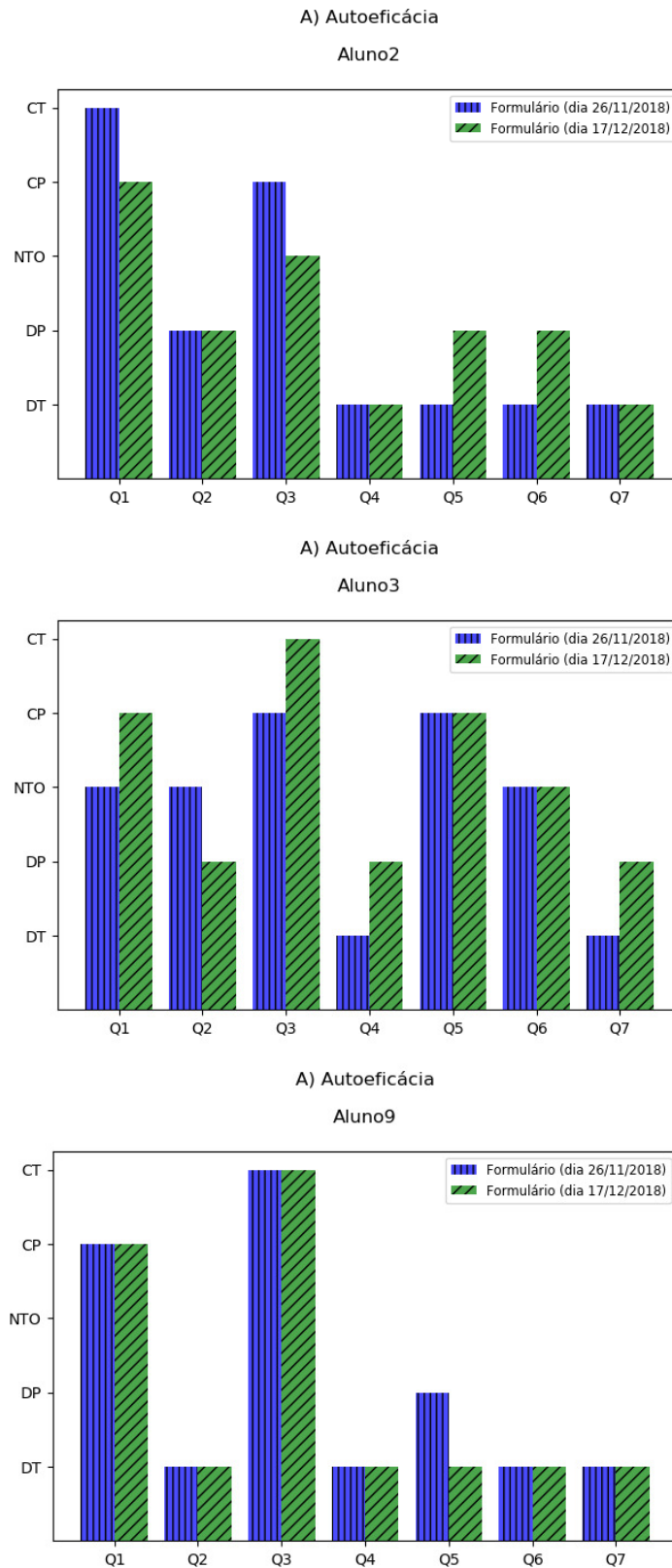
- "A utilização da plataforma para a correção trouxe uma maior imparcialidade na correção, e também me fez ter uma maior preocupação quanto a tempo e memória."
- "Uma boa plataforma para a disciplina. Possui ranking. Poderia haver uma porcentagem de erro, como no uri online."
- "Achei o uso da plataforma muito bom. Porém o desenvolvimento de certas questões se torna muito difícil para quem não tem prática, o que tornou o tempo para fazer as questões bem curto, já que durante a semana é complicado fazer essas, e nem sempre houve prolongamento para o final de semana."

Estas foram suas respostas a respeito do *blog*:

- "Não utilizei muito, mas me ajudou em um problema que não estava conseguindo criar uma lógica para solução."
- "facilitou o reúso (*sic*) de algoritmos."
- "Tomei conhecimento no site nos últimos momentos da disciplina, logo não pude usa-la adequadamente"

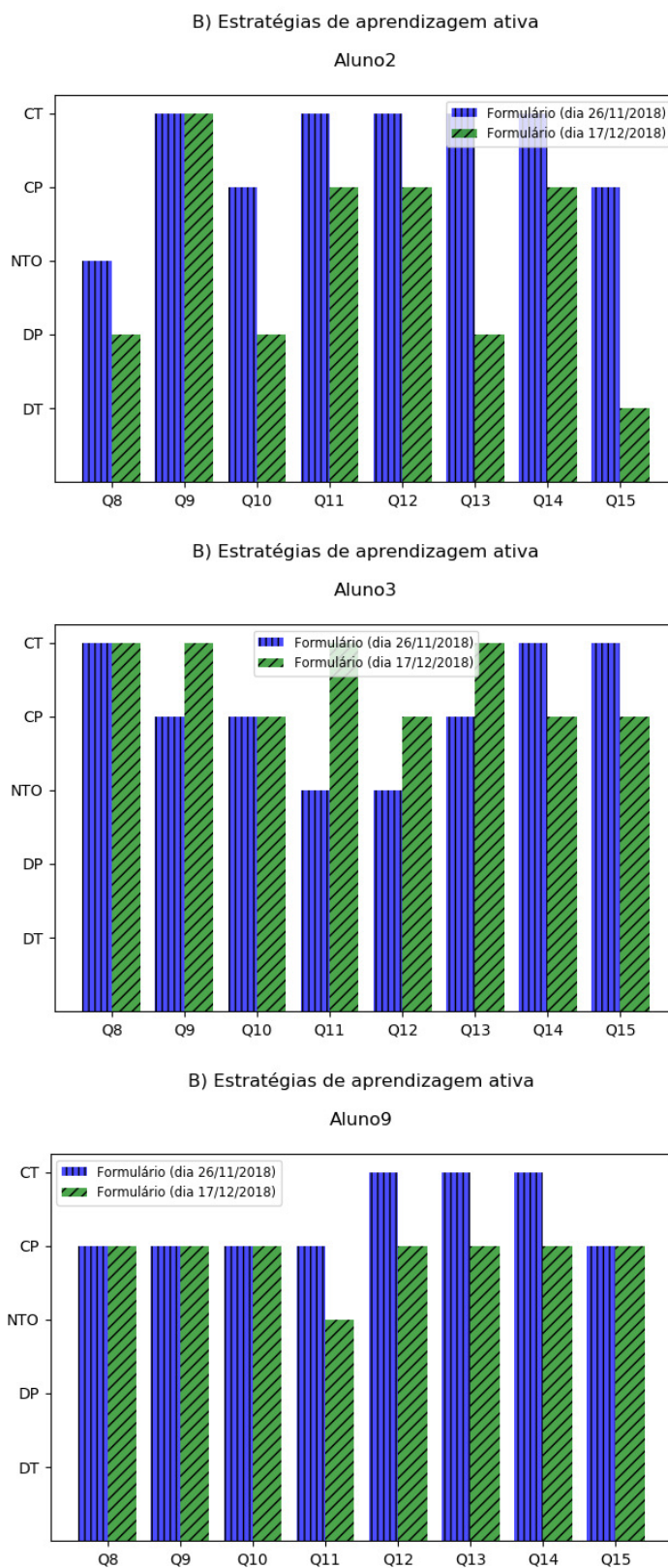
4.4.2 Comparação de resultados: Autoeficácia

Figura 25 – Comparação entre os resultados da Autoeficácia



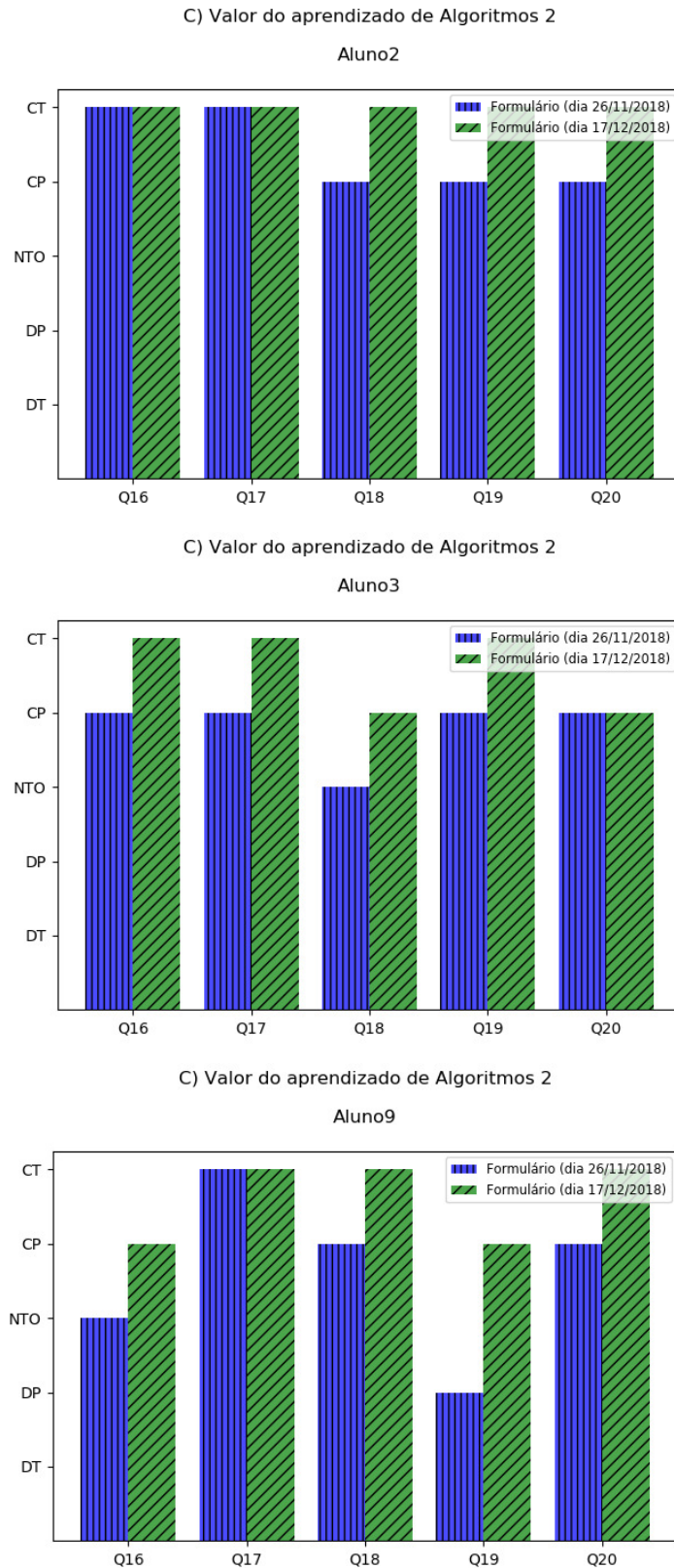
4.4.3 Comparação de resultados: Estratégias de Aprendizagem Ativa

Figura 26 – Comparação entre os resultados das Estratégias de Aprendizagem Ativa



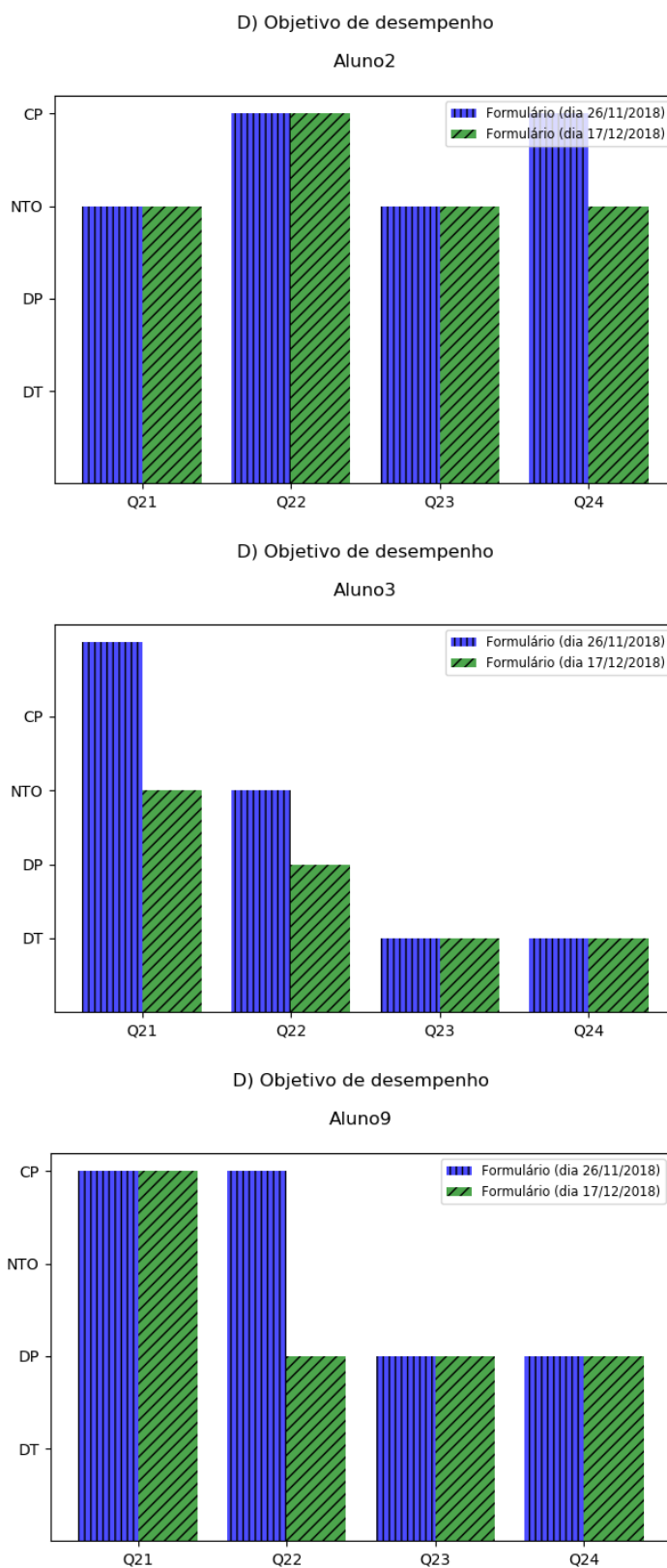
4.4.4 Comparação de resultados: Valor de Aprendizado de Algoritmos 2

Figura 27 – Comparação entre os resultados do Valor de Aprendizado de Algoritmos 2



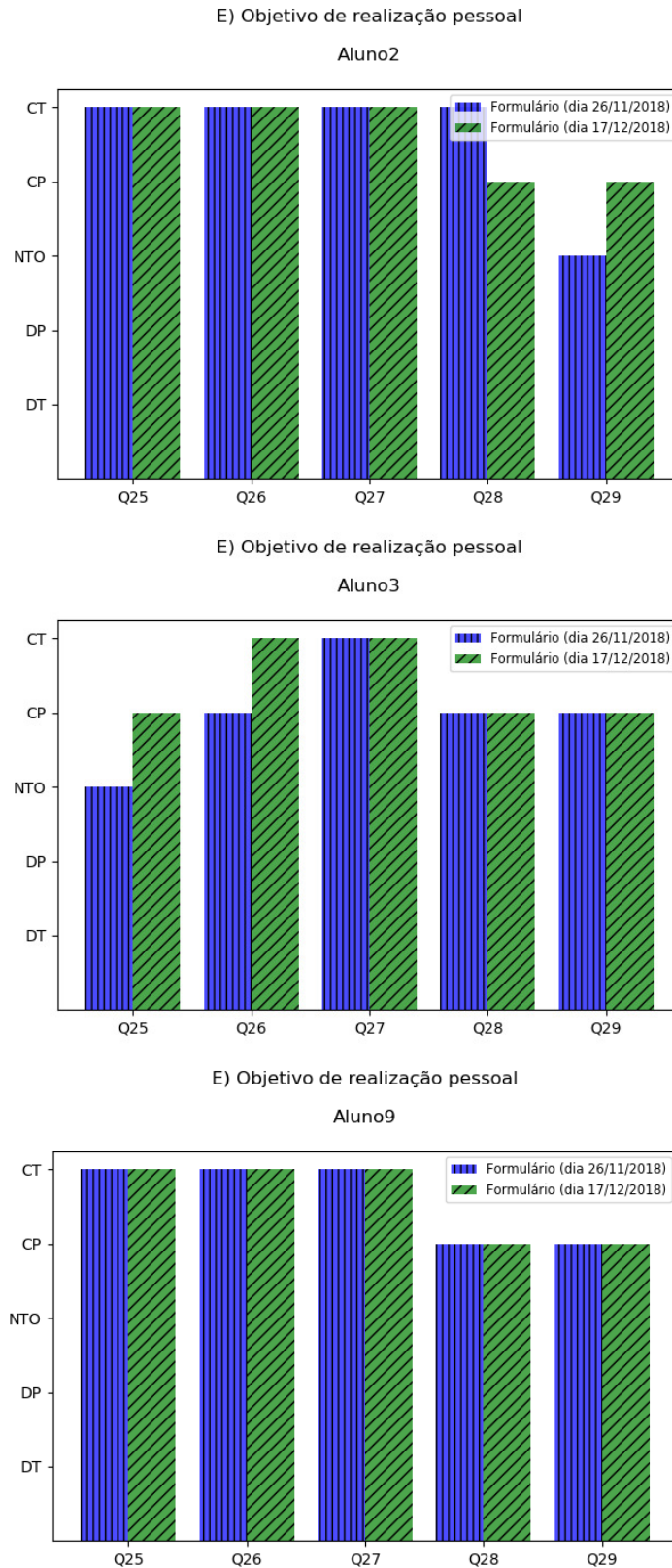
4.4.5 Comparação de resultados: Objetivo de desempenho

Figura 28 – Comparação entre os resultados do Objetivo de desempenho



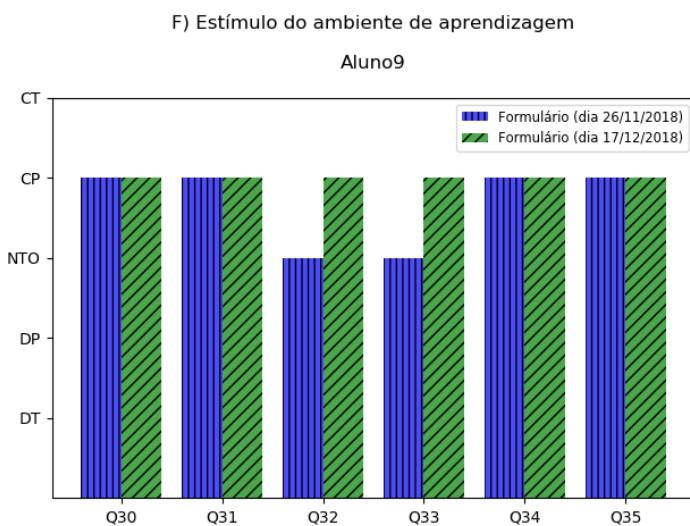
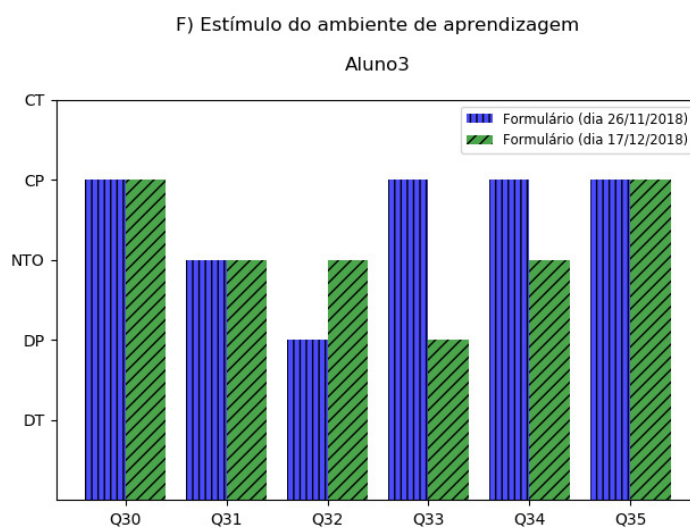
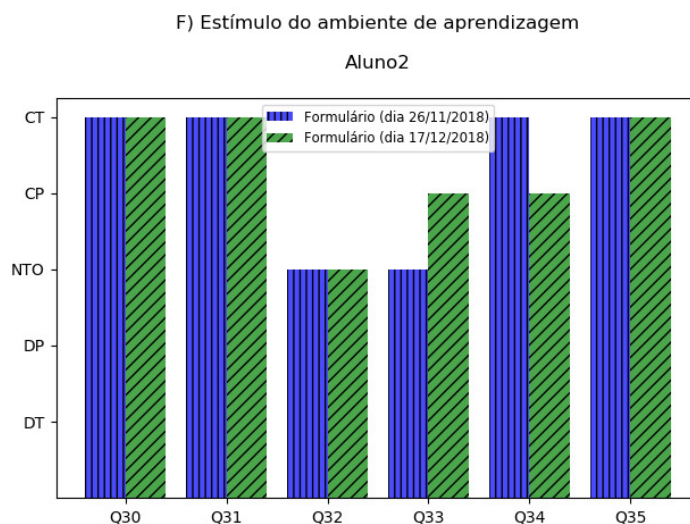
4.4.6 Comparação de resultados: Objetivo de realização pessoal

Figura 29 – Comparação entre os resultados do Objetivo de realização pessoal



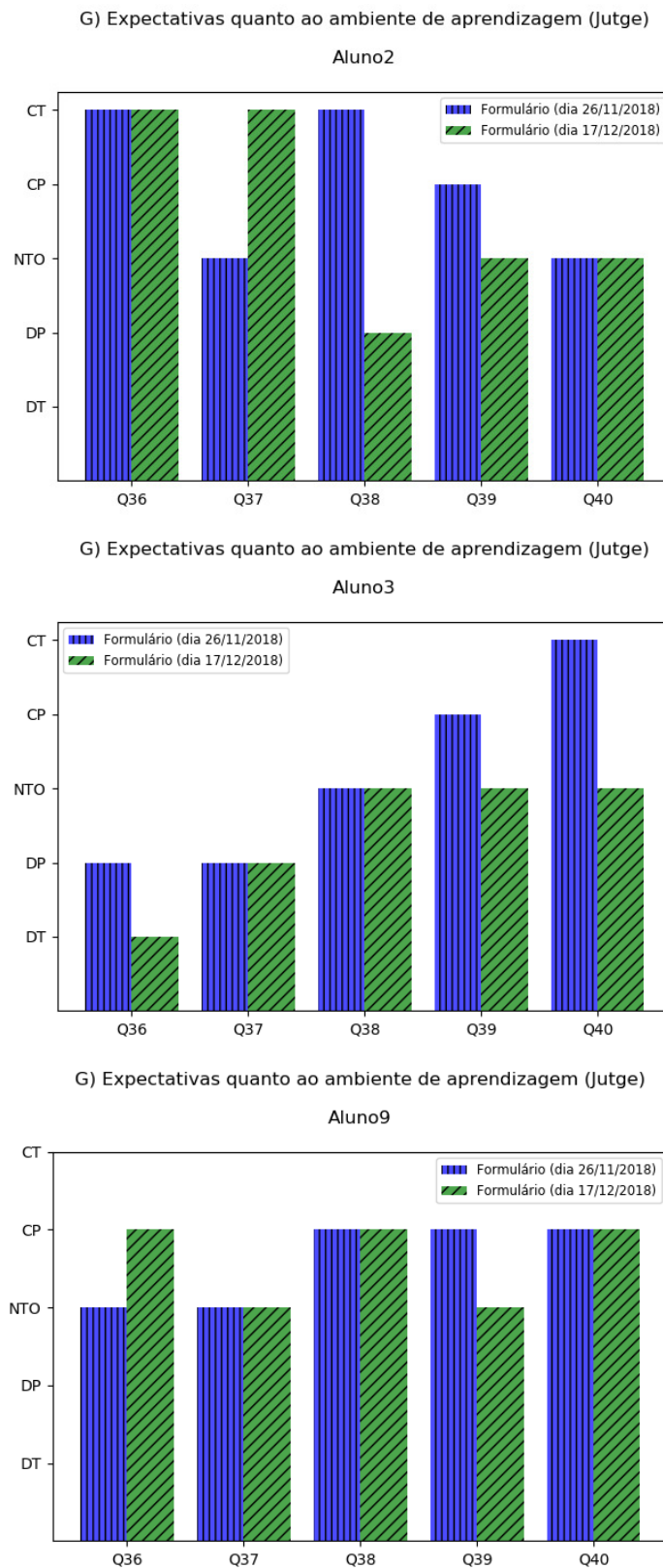
4.4.7 Comparação de resultados: Estímulo do ambiente de aprendizagem

Figura 30 – Comparação entre os resultados do Estímulo do ambiente de aprendizagem



4.4.8 Comparação de resultados: Expectativas com o ambiente de aprendizagem

Figura 31 – Comparação entre os resultados das Expectativas com o ambiente de aprendizagem



5 Conclusão

Esta monografia traz uma contribuição significativa para a Ciência da Computação pois reforça e comprova algumas das abordagens atuais que tratam sobre o ensino de algoritmos. Diante das ferramentas tecnológicas disponíveis hoje em dia, como os juízes *online* e os repositórios de problemas, é possível montar uma dinâmica que torne as aulas de programação mais estimulantes para os alunos, por meio de uma metodologia construtivista do conhecimento em que o professor é parte mediadora do processo.

5.1 Discussão dos Resultados

Dos 11 alunos matriculados, logo percebeu-se que apenas 9 estiveram comprometidos e engajados com a matéria desde o princípio. O fato de ser uma disciplina não-obrigatória pode ter contribuído para a desistência de alguns deles. De qualquer maneira, os torneios estavam sendo realizados pelo Judge e havia boa participação da turma durante a resolução de problemas em sala de aula.

O juiz *online* Judge, que teve papel fundamental na forma de avaliação da disciplina, se mostrou uma ferramenta poderosa para o ensino de algoritmos, devido à sua ampla gama de funcionalidades. A possibilidade de adicionar problemas, a consulta aos códigos-fonte dos alunos e a criação dos torneios possibilitam ao professor explorar novas maneiras de se pensar, estimulando a rapidez do raciocínio crítico e lógico-matemático dos estudantes.

Quanto ao *blog*, que iniciou simultaneamente aos torneios do assunto Teoria de Grafos, houve divulgação por parte do monitor da disciplina através de e-mail, bem como anúncio do professor durante a aula, porém não pareceu surtir efeito. Isso pode ter ocorrido devido à existência de uma biblioteca própria chamada "*Grafo.cpp*" organizada pelo professor e disponível aos alunos, que reunia os mais conhecidos algoritmos referentes a Grafo. Diante de uma biblioteca tão completa e suficiente, os alunos podem ter dado preferência a manter a utilização dela, que era abordada e revisada sempre em sala de aula, do que ir em busca de outros códigos-fonte desconhecidos.

A movimentação geral no *site* foi baixa. Apenas 6 alunos utilizaram a ferramenta e destes, apenas um, o *Aluno9*, respondeu a ambos os formulários de avaliação de motivação. Apesar disso, as respostas do *Aluno9* transmitem resultados bastante satisfatórios.

O destaque fica por conta do valor de aprendizado de Algoritmos 2. Enquanto

no primeiro formulário, o referido estudante afirmou que discordava parcialmente da afirmação Q19 "*Em Algoritmos 2, eu penso que é importante participar de atividades de desenvolvimento em grupo*", na segunda aplicação ele respondeu que concordava parcialmente. Isso pode ser consequência dos torneios realizados durante a sala de aula e da motivação que a competitividade causou nos alunos, visto que a mesma reação foi observada no *Aluno2* e no *Aluno3*. Ambos apresentaram um crescimento nesse quesito, como observa-se na [Figura 27](#).

Outro aspecto motivacional que teve melhorias foi o objetivo da realização pessoal. Os alunos sentiram-se realizados ao conseguir uma boa pontuação e houve aumento na confiança para resolver um problema, conforme [Figura 29](#).

5.2 Trabalhos futuros

Como forma de aprimorar este estudo, é recomendável que seja feita uma replicação desta experiência em uma turma futura, aplicando, desta vez, o formulário no início da disciplina enquanto os alunos ainda estão sem opiniões prévias, trazendo um comparativo mais fiel entre a motivação no início da disciplina e no final dela.

Da mesma forma, para garantir uma maior interação dos alunos com o *blog* repositório de códigos-fonte, o professor pode repensar as aulas expositivas e abordar uma visão de PBL, por exemplo, em que o aprendizado é baseado em problemas. No lugar de montar uma biblioteca *Grafo.cpp* em que o aluno tem acesso direto a vários protótipos de funções comuns da área, poderia ser mais interessante utilizar elementos da vida real, resolver problemas diversos e incentivar que o próprio aluno crie a sua biblioteca particular.

De maneira geral, mesmo sem ter o formulário como instrumento embasador da opinião do aluno, foi notório perceber que a turma estava mais envolvida com a disciplina, participando mais das aulas e principalmente dos torneios realizados pelo Judge. Houve melhora nas notas com o passar da disciplina e o clima na sala de aula, durante os torneios, demonstrava que a turma sentia-se motivada a aprender, fomentando, assim, uma competitividade sadia onde, no final, o vencedor do torneio ganha, mas todos aprendem e se desenvolvem juntos.

Referências

- ACM-ICPC <<https://icpc.baylor.edu/>>. Acesso em: dez/2018. Citado na página 26.
- AUSUBEL, D. P. The psychology of meaningful verbal learning. New York, 1963. Citado na página 33.
- AVIZ JÚNIOR, A. A. de. A aprendizagem de algoritmos: Uma experiência no curso de Tecnologia em Informática do CEFET-PA. Universidade Federal do Pará, Belém, p. 92, 2007. Citado na página 32.
- BARBOSA, L. da S. Aprendizado significativo aplicado ao ensino de algoritmos. Universidade Federal do Rio Grande do Norte, Natal, 2011. Citado na página 32.
- BERBEL, N. A. N. As metodologias ativas e a promoção da autonomia de estudantes. In: UNIVERSIDADE ESTADUAL DE LONDRINA. *Semina: Ciências Sociais e Humanas*. [S.l.], 2011. v. 32, p. 25–40. ISSN 1679-0383. Citado na página 32.
- Blog CodeForces. *Problems' Materials Publishing*. 2010. Disponível em: <<http://codeforces.com/blog/entry/967>>. Citado na página 30.
- Blog "codigosufma" <<http://codigosufma.online/>>. Acesso em: dez/2018. Citado 3 vezes nas páginas 45, 53 e 56.
- CARLOS, L.; GODINHO, J.; GOMIDE, J. Um relato de experiência da escola de verão de programação para crianças. In: *Anais do Workshop de Informática na Escola*. [S.l.: s.n.], 2018. v. 24, n. 1, p. 41. Citado na página 39.
- CASTRO, T. H. C. de et al. Utilizando programação funcional em disciplinas introdutórias de computação. 2003. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2002/0015.pdf>>. Acesso em: 10 dez. 18. Citado na página 21.
- CodeForces <<http://codeforces.com/>>. Acesso em: dez/2018. Citado na página 29.
- CONSELHO NACIONAL DE EDUCAÇÃO/CÂMARA DE EDUCAÇÃO SUPERIOR. *Diretrizes Curriculares Nacionais para os cursos de graduação em Computação*. [S.l.], 2002. Disponível em: <<http://portal.mec.gov.br/observatorio-da-educacao/323-secretarias-112877938/orgaos-vinculados-82187207/12991-diretrizes-curriculares-cursos-de-graduacao>>. Acesso em: 10 dez. 2018. Citado na página 21.
- Facebook Hacker Cup <<http://www.facebook.com/hackercup>>. Acesso em: dez/2018. Citado na página 26.
- Google Code Jam <<https://codingcompetitions.withgoogle.com/codejam>>. Acesso em: dez/2018. Citado na página 26.
- Hour of Code <<https://hourofcode.com/>>. Acesso em: dez/2018. Citado na página 39.

IRION, C.; PELEGRINO, D. H.; BOTELHO, M. P. da S. A motivação através da competitividade-a busca pela qualidade da educação no estudo da computação. *SIED: EnPED-Simpósio Internacional de Educação a Distância e Encontro de Pesquisadores em Educação a Distância*, 2016. Citado na página 36.

Judge.org <<https://www.judge.org/>>. Acesso em: dez/2018. Citado na página 36.

MICROSOFT QUANTUM TEAM. *Challenge your skills in the Microsoft Q Coding Contest – Summer 2018*. 2018. Disponível em: <<https://cloudblogs.microsoft.com/quantum/2018/06/27/challenge-your-skills-in-the-microsoft-q-coding-contest-summer-2018/>>. Citado na página 31.

OLIVEIRA, A. S. Uma proposta de ensino semipresencial de programação apoiada por juiz on-line e ambiente virtual de aprendizagem móvel. São Cristóvão, Brasil, 2017. Citado na página 32.

PEREIRA JÚNIOR, J. C. R.; RAPKIEWICZ, C. E. O processo de ensino-aprendizagem de fundamentos de programação: Uma visão crítica da pesquisa no Brasil. 2004. Citado na página 32.

SOLEDADE, M. *Aprendizagem Baseada em Problemas (PBL), o que é?* 2018. Disponível em: <<https://silabe.com.br/blog/aprendizagem-baseada-em-problemas-pbl/>>. Acesso em: dez. 2018. Citado na página 33.

Sphere Online Judge (SPOJ) <<https://www.spoj.com/>>. Acesso em: dez/2018. Citado na página 28.

TIMMERMANN, G. L. K. O desafio de aprender e ensinar algoritmos: Mediações que professores e alunos estabelecem com o conteúdo no ensino superior. 2015. Disponível em: <<http://bibliodigital.unijui.edu.br:8080/xmlui/handle/123456789/4989>>. Acesso em: 10 dez. 2018. Citado na página 21.

TUAN, H.-L.; CHIN, C.-C.; SHIEH, S.-H. The development of a questionnaire to measure students' motivation towards science learning. v. 27, n. 6, p. 639–654, 2005. Citado 2 vezes nas páginas 32 e 45.

uDebug <<https://www.udebug.com/>>. Acesso em: dez/2018. Citado na página 26.

uHunt <<https://uhunt.onlinejudge.org/>>. Acesso em: dez/2018. Citado na página 24.

URI Online Judge <<http://www.urionlinejudge.com.br/>>. Acesso em: dez/2018. Citado na página 27.

UVa Online Judge <<https://www.uvaonlinejudge.org>>. Acesso em: dez/2018. Citado na página 23.

Wordpress <<https://pt.wikipedia.org/wiki/WordPress>>. Acesso em: dez/2018. Citado na página 42.

APÊNDICE A – Formulário de Motivação

Este questionário contém afirmações sobre sua motivação ao participar da disciplina de Algoritmos 2. Você deve expressar sua opinião sobre cada afirmação. Não existe um "certo" ou "errado". O que importa é a sua opinião. Pense sobre como cada afirmação descreve melhor sua motivação em participar nas aulas de Algoritmos 2.

Faça um círculo no número correspondente a:

1. Se você discorda totalmente da afirmação (DT)
2. Se você discorda parcialmente da afirmação (DP)
3. Se você não tem opinião sobre a afirmação (NTO)
4. Se você concorda parcialmente com a afirmação (CP)
5. Se você concorda totalmente com a afirmação (CT)

Preste atenção para responder todas as questões. Se você mudar de ideia sobre uma resposta, apenas faça um "X" nela e circule outra.

Algumas afirmações são muito parecidas entre si. Não se prenda a isso. Simplesmente dê a sua opinião sobre todas as afirmações.

Nome completo: _____

A. Auto eficácia	DT	DP	NTO	CP	CT
1. Independente de o conteúdo da disciplina de Algoritmos 2 ser fácil ou difícil, eu tenho certeza que consigo compreendê-lo.	1	2	3	4	5
2. Eu não me sinto confiante para compreender conceitos difíceis da disciplina de Algoritmos 2.	1	2	3	4	5
3. Eu tenho certeza de que posso ir bem nos testes de Algoritmos 2.	1	2	3	4	5
4. Não importa o quanto eu me esforce, eu não consigo aprender os conceitos de Algoritmos 2.	1	2	3	4	5
5. Quando os problemas de Algoritmos 2 são muito difíceis, eu desisto ou apenas faço as partes fáceis.	1	2	3	4	5
6. Durante os testes de Algoritmos 2, eu prefiro pedir ajuda para outras pessoas ao invés de pensar por mim mesmo.	1	2	3	4	5
7. Quando eu acho um conteúdo de Algoritmos 2 difícil, eu nem tento aprendê-lo.	1	2	3	4	5
B. Estratégias de aprendizagem ativa	DT	DP	NTO	CP	CT
8. Quando eu aprendo novos conceitos de algoritmos, eu me esforço para compreendê-lo.	1	2	3	4	5
9. Quando eu aprendo novos conceitos de Algoritmos 2, eu os relaciono com minhas experiências anteriores.	1	2	3	4	5
10. Quando eu não compreendo um conceito de Algoritmos 2, eu procuro meios relevantes que possam me ajudar.	1	2	3	4	5
11. Quando eu não compreendo um conceito de Algoritmos 2, eu procuro discutir com o professor ou outros colegas para facilitar minha compreensão.	1	2	3	4	5
12. Durante o processo de aprendizagem, eu tento fazer conexões entre os conceitos que aprendo.	1	2	3	4	5
13. Quando eu cometo um erro em um algoritmo, eu tento descobrir o porquê.	1	2	3	4	5
14. Quando encontro conceitos de Algoritmos 2 que eu não compreendo, ainda assim, me esforço para aprendê-los.	1	2	3	4	5
15. Quando novos conceitos que eu aprendo entram em conflito com meus conhecimentos anteriores, eu tento compreender o porquê.	1	2	3	4	5
C. Valor do aprendizado de Algoritmos 2	DT	DP	NTO	CP	CT
16. Eu penso que aprender os conceitos de Algoritmos 2 é importante porque eu posso usá-los no meu dia a dia.	1	2	3	4	5
17. Eu penso que aprender os conceitos de Algoritmos 2 é importante porque estimula meu raciocínio.	1	2	3	4	5
18. Em Algoritmos 2, eu penso que é importante aprender a resolver problemas práticos.	1	2	3	4	5
19. Em Algoritmos 2, eu penso que é importante participar de atividades de desenvolvimento em grupo.	1	2	3	4	5
20. Enquanto estou aprendendo Algoritmos 2, é importante conseguir satisfazer minha própria curiosidade.	1	2	3	4	5
D. Objetivo de desempenho	DT	DP	NTO	CP	CT
21. Eu participo dos testes de Algoritmos 2 para ter uma boa nota.	1	2	3	4	5
22. Eu participo dos testes de Algoritmos 2 para ter melhor desempenho que outros estudantes.	1	2	3	4	5
23. Eu participo dos testes de Algoritmos 2 para que outros estudantes pensem que eu sou inteligente.	1	2	3	4	5
24. Eu participo dos testes de Algoritmos 2 para que o professor preste atenção em mim.	1	2	3	4	5
E. Objetivo de realização pessoal	DT	DP	NTO	CP	CT
25. Durante um teste de Algoritmos 2, eu me sinto mais realizado quando obtenho uma boa pontuação.	1	2	3	4	5
26. Durante um teste de Algoritmos 2, eu me sinto mais realizado quando me sinto confiante sobre o conteúdo.	1	2	3	4	5
27. Durante um teste de Algoritmos 2, eu me sinto mais realizado quando sou capaz de solucionar um problema difícil.	1	2	3	4	5
28. Durante uma aula de Algoritmos 2, eu me sinto mais realizado quando o professor aceita minhas ideias.	1	2	3	4	5
29. Durante uma aula de Algoritmos 2, eu me sinto mais realizado quando outros estudantes aceitam minhas ideias.	1	2	3	4	5
F. Estímulo do ambiente de aprendizagem	DT	DP	NTO	CP	CT
30. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque o conteúdo é empolgante e dinâmico.	1	2	3	4	5
31. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque o professor usa diferentes estratégias de ensino.	1	2	3	4	5
32. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque o professor não me pressiona muito.	1	2	3	4	5
33. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque o professor presta atenção em mim.	1	2	3	4	5
34. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque os problemas são desafiadores.	1	2	3	4	5
35. Eu me sinto motivado a participar dos testes de Algoritmos 2 porque a turma participa de debates.	1	2	3	4	5
36. Eu gosto de fazer os testes realizados na plataforma Jutge.org.	1	2	3	4	5
37. Os testes realizados na plataforma Jutge.org me motivaram a estudar mais.	1	2	3	4	5
38. Durante os testes de Algoritmos 2, eu me sinto motivado mesmo quando não entendo os conceitos referentes ao problema.	1	2	3	4	5
39. Durante os testes de Algoritmos 2, eu me sinto motivado quando encontro a solução de um problema semelhante.	1	2	3	4	5
40. Durante os testes de Algoritmos 2, eu me sentirei mais motivado, se tiver acesso a soluções de problemas semelhantes.	1	2	3	4	5

Escreva um pouco sobre o que você acha sobre os testes realizados na plataforma Jutge.org:

APÊNDICE B – Banco de logs completo do blog codigosufma

Resultado SQL

Host: null

Banco de dados: bdcodigos

Tempo de geração: 20/12/2018 às 12:16

Gerado por: phpMyAdmin 4.2.7.1 / MySQL 5.6.26-log

Query SQL: SELECT ID_logs, data_logs, u.ID_aluno, link_post FROM userlogs AS l, codigos AS c, userpass AS u WHERE l.fk_ID_aluno = u.ID_aluno AND l.fk_filename_cods = c.filename_cods ORDER BY `l`.`ID_logs` ASC;

Registros: 87

A seleção atual não contém uma única coluna. As funções edição em grade, checkbox, editar, copiar e apagar não estão disponíveis.

ID_logs	data_logs	ID_aluno	link_post
1	2018-11-26 16:59:33	9	Lista de Codigos
2	2018-11-26 16:59:42	9	10004 - Bicolorable (BFS)
3	2018-11-26 17:01:29	13	Lista de Codigos
4	2018-11-26 17:01:51	13	1112 - Mice and Maze (Shortest-Paths)
5	2018-11-26 17:02:07	13	10004 - Bicolorable (BFS)
6	2018-11-26 17:02:47	9	10067 - Playing with Wheels (BFS)
7	2018-11-26 17:03:15	9	908 - Re-connecting Computer Sites (MST)
8	2018-11-26 17:03:17	13	10067 - Playing with Wheels (BFS)
9	2018-11-26 17:23:14	12	Lista de Codigos
10	2018-11-26 17:32:31	13	10653 - Bombs! NO they are Mines!! (BFS)
11	2018-11-26 17:32:44	13	429 - Word Transformation (BFS)
12	2018-11-26 17:33:02	13	10067 - Playing with Wheels (BFS)
13	2018-11-26 17:33:08	13	11838 - Come and Go (DFS)
14	2018-11-26 17:33:34	13	10009 - All Roads Lead Where? (DFS)
15	2018-11-26 21:41:34	9	Lista de Codigos
16	2018-11-26 21:41:43	9	10067 - Playing with Wheels (BFS)
17	2018-11-26 21:42:26	9	429 - Word Transformation (BFS)
18	2018-11-28 18:23:45	7	Lista de Codigos
19	2018-11-28 21:32:41	7	Lista de Codigos
20	2018-11-28 21:35:10	7	10009 - All Roads Lead Where? (DFS)
21	2018-11-28 21:35:32	7	10004 - Bicolorable (BFS)
22	2018-11-28 22:04:29	12	Lista de Codigos
23	2018-11-28 22:04:40	12	10067 - Playing with Wheels (BFS)
24	2018-11-28 22:05:03	12	10004 - Bicolorable (BFS)
25	2018-11-28 22:05:20	12	10653 - Bombs! NO they are Mines!! (BFS)
26	2018-11-28 22:05:37	12	383 - Shipping Routes (BFS)
27	2018-11-28 22:05:58	12	417 - Word Index (BFS)
28	2018-11-28 22:06:10	12	429 - Word Transformation (BFS)
29	2018-11-29 09:57:27	7	Lista de Codigos

ID_logs	data_logs	ID_aluno	link_post
30	2018-11-29 09:57:48	7	1112 - Mice and Maze (Shortest-Paths)
31	2018-11-29 22:13:45	11	Lista de Codigos
32	2018-11-29 22:14:09	11	10004 - Bicolorable (BFS)
33	2018-11-30 12:36:48	7	Lista de Codigos
34	2018-11-30 12:47:23	7	10653 - Bombs! NO they are Mines!! (BFS)
35	2018-11-30 12:48:21	7	10336 - Rank the Languages (FF)
36	2018-12-01 18:16:11	7	Lista de Codigos
37	2018-12-01 18:16:48	7	908 - Re-connecting Computer Sites (MST)
38	2018-12-01 18:17:22	7	10653 - Bombs! NO they are Mines!! (BFS)
39	2018-12-02 11:42:59	12	Lista de Codigos
40	2018-12-02 11:43:20	12	10004 - Bicolorable (BFS)
41	2018-12-02 11:45:52	12	908 - Re-connecting Computer Sites (MST)
42	2018-12-02 12:08:56	12	11838 - Come and Go (DFS)
43	2018-12-02 12:09:00	12	10009 - All Roads Lead Where? (DFS)
44	2018-12-02 12:09:07	12	429 - Word Transformation (BFS)
45	2018-12-03 16:46:33	11	Lista de Codigos
46	2018-12-03 16:46:50	11	10305 - Ordering Tasks (TOPSORT)
47	2018-12-03 16:52:37	9	Lista de Codigos
48	2018-12-03 16:52:45	11	10305 - Ordering Tasks (TOPSORT)
49	2018-12-03 16:52:47	9	10305 - Ordering Tasks (TOPSORT)
50	2018-12-03 17:29:50	12	Lista de Codigos
51	2018-12-03 17:29:58	12	10305 - Ordering Tasks (TOPSORT)
52	2018-12-03 17:30:07	12	1112 - Mice and Maze (Shortest-Paths)
53	2018-12-03 17:30:11	12	908 - Re-connecting Computer Sites (MST)
54	2018-12-03 17:30:18	12	572 - Oil Deposits (FF)
55	2018-12-03 17:30:29	12	469 - Wetlands of Florida (FF)
56	2018-12-03 17:30:36	12	10336 - Rank the Languages (FF)
57	2018-12-03 17:30:41	12	341 - Non-Stop Travel (DIJ)
58	2018-12-03 17:30:54	12	11838 - Come and Go (DFS)
59	2018-12-03 17:31:00	12	10009 - All Roads Lead Where? (DFS)
60	2018-12-03 17:31:05	12	429 - Word Transformation (BFS)
61	2018-12-03 17:46:31	12	417 - Word Index (BFS)
62	2018-12-03 17:46:40	12	10305 - Ordering Tasks (TOPSORT)
63	2018-12-04 13:23:22	9	Lista de Codigos
64	2018-12-04 13:23:33	9	341 - Non-Stop Travel (DIJ)
65	2018-12-04 17:09:03	9	Lista de Codigos
66	2018-12-04 17:09:11	9	341 - Non-Stop Travel (DIJ)
67	2018-12-04 17:26:13	9	Lista de Codigos
68	2018-12-04 17:26:24	9	341 - Non-Stop Travel (DIJ)
69	2018-12-05 00:05:14	4	Lista de Codigos

ID_logs	data_logs	ID_aluno	link_post
70	2018-12-05 00:05:30	4	10009 - All Roads Lead Where? (DFS)
71	2018-12-05 11:37:37	4	Lista de Codigos
72	2018-12-05 11:40:47	4	10305 - Ordering Tasks (TOPSORT)
73	2018-12-05 16:01:33	4	Lista de Codigos
74	2018-12-05 16:01:44	4	341 - Non-Stop Travel (DIJ)
75	2018-12-05 18:09:49	11	Lista de Codigos
76	2018-12-05 18:10:17	11	341 - Non-Stop Travel (DIJ)
77	2018-12-07 03:25:35	4	Lista de Codigos
78	2018-12-07 16:28:14	13	Lista de Codigos
79	2018-12-07 16:28:42	13	10067 - Playing with Wheels (BFS)
80	2018-12-07 16:28:57	13	341 - Non-Stop Travel (DIJ)
81	2018-12-07 16:29:14	13	10004 - Bicolorable (BFS)
82	2018-12-07 18:51:27	4	Lista de Codigos
83	2018-12-07 18:51:49	4	10305 - Ordering Tasks (TOPSORT)
84	2018-12-07 19:35:13	4	Lista de Codigos
85	2018-12-07 19:35:56	4	10305 - Ordering Tasks (TOPSORT)
86	2018-12-17 15:19:05	12	Lista de Codigos
87	2018-12-17 15:19:32	12	10067 - Playing with Wheels (BFS)

APÊNDICE C – Lista de arquivos necessários para a criação de um problema no Jutge

Nome do arquivo	O que é	Exemplo
handler.yml	Arquivo do tipo YML com informações técnicas sobre o problema como o namespace a ser utilizado, se o problema contém função main, entre outros parâmetros.	<i>handler: std</i> <i>source_modifier: none</i>
problem.LANG.yml	Arquivo do tipo YML que guarda os metadados do problema. A sigla LANG representa dois caracteres que denotam o idioma do problema. Atualmente o Jutge aceita e reconhece apenas as opções 'ca' (catalão), 'de' (alemão), 'en' (inglês) e 'es' (espanhol).	<i>title: Soma de inteiros</i> <i>author: Felix Halim, UVA</i> <i>email:felix.halim@uva.com</i> <i>translator:Carlos de Salles</i> <i>translator_email: csal-</i> <i>les@deinf.ufma.br</i>

problem.LANG.tex	Arquivo na linguagem LaTeX que descreve o texto do problema por meio de tags próprias, conforme o exemplo ao lado. Caso o professor queira incluir o problema em mais de um idioma, ele deve incluir mais de um arquivo .txt, sendo cada um com sua sigla correspondente ('ca', 'de', 'en' ou 'es').	$\backslash Problem\{Soma\ de\ inteiros\}$ $\backslash Statement$ <i>Escreva um programa que calcule a soma de dois inteiros</i> $\backslash Input$ <i>A entrada consiste em dois números inteiros.</i> $\backslash Output$ <i>Imprima uma linha com a soma dos dois números.</i> $\backslash Sample$
sample[*].inp	Caso de teste (entrada) com a extensão .inp. Os casos que iniciam com 'sample' são públicos e são apresentados juntos com a descrição do problema na página dele. Atenção: para cada arquivo de entrada sample[*].inp deve haver o respectivo arquivo de saída sample[*].cor.	<i>Exemplos de nomes de arquivos:</i> <i>sample-1.inp</i> <i>sample-2.inp</i> <i>sample-3.inp</i>
sample[*].cor	Caso de teste (saída) com a extensão .cor. Os casos que iniciam com 'sample' são públicos e são apresentados juntos com a descrição do problema na página dele. Atenção: para cada arquivo de saída sample[*].cor deve haver o respectivo arquivo de entrada sample[*].inp.	<i>Exemplos de nomes de arquivos:</i> <i>sample-1.cor</i> <i>sample-2.cor</i> <i>sample-3.cor</i>

test[*].inp	<p>Caso de teste (entrada) com a extensão .inp. Os casos que iniciam com 'test' são secretos. Geralmente, são mais extensos e servem para testar se o algoritmo submetido pelo aluno está bem formulado para casos de teste críticos. Atenção: para cada arquivo de entrada test[*].inp deve haver o respectivo arquivo de saída test[*].cor.</p>	<p><i>Exemplos de nomes de arquivos:</i></p> <p><i>test-1.inp</i> <i>test-2.inp</i> <i>test-3.inp</i></p>
test[*].cor	<p>Caso de teste (saída) com a extensão .cor. Os casos que iniciam com 'test' são secretos. Geralmente, são mais extensos e servem para testar se o algoritmo submetido pelo aluno está bem formulado para casos de teste críticos. Atenção: para cada arquivo de saída test[*].cor deve haver o respectivo arquivo de entrada test[*].inp.</p>	<p><i>Exemplos de nomes de arquivos:</i></p> <p><i>test-1.cor</i> <i>test-2.cor</i> <i>test-3.cor</i></p>

solution.[c,cc, py]	Todo problema deve ser incluído junto com sua solução própria, que vai ser utilizada com os arquivos <i>sample</i> e <i>test</i> de entrada e saída para verificar se o problema está realmente válido, ou seja, se há um algoritmo que produza aquelas saídas <i>.cor</i> para cada arquivo correspondente <i>.inp</i> .	Apenas são aceitas soluções oficiais em uma das seguintes linguagens: C (.c), C++ (.cc) ou Python (.py).
---------------------	---	--

APÊNDICE D – Algumas questões criadas no Jutge

Caixas (Bolas de gude 3)

X22218_en

João e Maria terminaram de brincar com bolas de gude e agora devem guardá-las em caixas. As caixas têm dimensões idênticas, são numeradas de 1 a N e devem ser empilhadas da seguinte forma:

1. As caixas de número menor não podem ser colocadas em cima de caixas de número maior.
2. Cada caixa possui um peso e uma carga máxima. A soma dos pesos das caixas em cima de determinada caixa não pode ultrapassar sua carga máxima.

Escreva um programa que encontre o número máximo de caixas que podem ser empilhadas.

Input

A primeira linha de um caso de teste é um inteiro N ($1 \leq N \leq 1000$), seguido de N linhas, cada uma com dois inteiros (ambos ≤ 3000), que representam o peso e a carga máxima que uma caixa suporta, respectivamente. A entrada termina com o caso N = 0.

Output

Cada linha da saída representa a quantidade máxima de caixas que podem ser empilhadas.

Sample input

```
5
19 15
7 13
5 7
6 8
1 2
0
```

Sample output

```
4
```

Problem information

Author : Carlos de Salles, DEINF/UFMA

Generation : 2018-12-20 16:33:36

© Jutge.org, 2006–2018.

<https://jutge.org>

Campus 311

X29776_en

Logo em seu primeiro dia de aula no curso de Computação, José, primo de João e Maria, ficou impressionado com o tempo que demorou até chegar à UFMA. Além da distância entre seu bairro e a universidade, José observou que há muitos semáforos na cidade que atrasam o fluxo do trânsito durante o percurso.

Sendo assim, ele desenhou todo o trajeto que faz de sua casa até a UFMA através de rotas e quantificou o tempo que gasta em cada uma das avenidas.

Como você está em períodos mais avançados que José, escreva um programa que ajude-o a encontrar o melhor percurso para chegar à UFMA no menor tempo.

Input

A primeira linha de cada caso de teste traz $P (< 50)$, o número de pontos no mapa que José traçou. Cada uma das P linhas seguintes inicia com um número n , a quantidade de avenidas que saem do ponto p . Em seguida, há n pares de números x e y , representando um outro ponto q do mapa e o tempo que gasta no ônibus entre os pontos p e q .

Por fim, temos dois números A e B , que são os pontos representados pela posição atual de José e da UFMA.

No trajeto em que José faz, todas as avenidas são de mão dupla.

A entrada se encerra com $P = 0$.

Output

Para cada caso de teste, a saída deve obedecer o formato apresentado no exemplo:

Rota 'i': chegada em 'j' minutos., onde 'i' representa o caso de teste e 'j' é o tempo mínimo em que José chegaria na UFMA, se o Campus 311 seguisse a melhor rota.

Sample input

```
5
2 3 3 4 6
3 1 2 3 7 5 6
1 4 5
0
1 4 7
2 4

2
1 2 5
1 1 6
1 2

7
4 2 5 3 13 4 8 5 18
2 3 7 6 14
1 6 6
2 3 5 5 9
3 6 2 7 9 4 6
1 7 2
0
```

```
1 7
0
```

Sample output

Rota 1: chegada em 8 minutos.

Rota 2: chegada em 5 minutos.
Rota 3: chegada em 20 minutos.

Problem information

Author :

Generation : 2018-12-10 20:22:57

© *Jutge.org*, 2006–2018.

<https://jutge.org>

Ordenando o curso

X33397_en

José, o primo mais velho de João e Maria, entrou no curso de Ciência da Computação e, muito animado, foi na Coordenação se informar mais sobre as disciplinas. Lá, ele recebeu uma lista das disciplinas que deve cursar e um conjunto de definições de pré-requisitos. Como José é um aluno bastante aplicado, ele quer fazer as disciplinas na ordem de prioridade em que foram apresentadas na lista.

Diante disto, ajude José a definir uma ordem de disciplinas, respeitando as definições de pré-requisitos.

Input

A primeira linha de cada caso de teste traz dois números m e n que representam a lista de disciplinas e a quantidade de pré-requisitos. A seguir seguem m linhas, cada uma com o nome de uma disciplina (com no máximo 100 caracteres). As próximas n linhas representam uma condição onde " $disciplina_a disciplina_b$ " significa que a $disciplina_a$ deve ser feita antes da $disciplina_b$.

Output

Para cada caso de teste, imprima a sequência única de disciplinas que satisfaça os critérios estabelecidos.

Obs.: No segundo caso de teste, a sequência "calculo4 calculo1 calculo2 calculo3" obedece aos critérios dos pré-requisitos, porém não satisfaz a condição de prioridade na ordem da lista.

Sample input 1

```
10 6
Materia_A
Materia_B
Materia_C
Materia_D
Materia_E
Materia_F
Materia_G
Materia_H
Materia_I
Materia_J
Materia_E Materia_B
Materia_G Materia_E
Materia_A Materia_I
Materia_G Materia_F
Materia_G Materia_D
Materia_A Materia_F
0 0
```

Sample input 2

```
4 3
calculo1
calculo2
calculo3
```

Sample output 1

```
Materia_A
Materia_C
Materia_G
Materia_D
Materia_E
Materia_B
Materia_F
Materia_H
Materia_I
Materia_J
```

```
calculo4
calculo1 calculo2
calculo2 calculo3
calculo1 calculo3
0 0
```

Sample output 2

```
calculo1
```

Sample input 3

```
5 4
algoritmos_1
linguagem_c
paradigmas_de_programacao
estruturas_de_dados
algoritmos_2
algoritmos_1 linguagem_c
linguagem_c paradigmas_de_programacao
algoritmos_1 paradigmas_de_programacao
algoritmos_1 algoritmos_2
0 0
```

```
calculo2
calculo3
calculo4
```

Sample output 3

```
algoritmos_1
linguagem_c
paradigmas_de_programacao
estruturas_de_dados
algoritmos_2
```

Problem information

Author :

Generation : 2018-12-10 22:32:50

© *Jutge.org*, 2006–2018.

<https://jutge.org>

Pokébolas

X37265_en

Após jogar Pokémon Go por toda a UFMA, José, primo de João e Maria e calouro de Ciência da Computação, decidiu fazer um mapa com todos os Pokéstops da cidade e as distâncias entre elas. Ele percebeu que nem sempre há um ônibus que passe entre os Pokéstops $P1$ e $P2$, mas, sempre que existir tal ônibus, este também faz o percurso de volta.

José tem um cartão de passagem Nível 4, que sempre debita o mesmo valor máximo X . Na cidade rodam ônibus com tarifas diferentes, e José está preocupado em encontrar um caminho no qual o valor máximo de tarifa entre os trechos seja o menor possível.

Input

A entrada consiste em vários casos de teste. Cada caso de teste começa com três inteiros P , T e C , que representam o número de Pokéstops, a quantidade de Trechos na cidade e as Consultas que José deve realizar.

Output

Para cada caso de teste, imprima "*Caso #*" com o número do respectivo caso. Na linha seguinte, para cada consulta entre $P1$ e $P2$, imprima o menor valor que seja a tarifa máxima dentre todos os caminhos entre $P1$ e $P2$.

Caso não exista um caminho entre $P1$ e $P2$, José não poderá jogar, portanto imprima "*sem pokebolas*".

Exemplo: No primeiro caso de teste, o menor caminho entre os pontos 1 e 5 é 1-2-3-5 (60+50+90 = 200), com valor máximo de tarifa 90. No entanto, há um outro caminho que apresenta um menor valor máximo: 1-2-3-4-5 (60+50+80+60 = 250), cujo valor máximo de tarifa é 80. Este é o menor valor máximo entre todos os caminhos possíveis.

Sample input 1

```
7 6 3
1 2 60
2 3 50
3 4 80
3 5 90
4 5 60
6 7 100
1 5
3 5
1 7
0 0 0
```

Sample output 1

```
Caso #1
80
80
sem pokebolas
```

Sample input 2

```
3 3 3
1 2 74
1 3 89
2 3 91
1 2
1 3
2 3
0 0 0
```

Sample output 2

```
Caso #1
74
89
89
```

Problem information

Author :

Generation : 2018-12-10 20:26:20

© *Jutge.org*, 2006–2018.

<https://jutge.org>