

SORTIA

Um Jogo para Ensino de Algoritmo de Ordenação: Estudo de caso na Disciplina de Estrutura de Dados

Paulo Battistella
Aldo von Wangenheim
Christiane G. von Wangenheim

GQS – Grupo de Qualidade de Software
INCoD - Instituto Nacional de Ciência e Tecnologia para Convergência Digital

Ensino de Algoritmo de Ordenação

Plano de Ensino

Alocação dinâmica de memória

Variáveis estáticas e dinâmicas
Ponteiros
Passagem de parâmetros por referência, valor e nome

Estruturas lineares

Listas
Pilhas
Filas

Complexidade de algoritmos

Conceitos de Complexidade de Algoritmos
Métodos práticos para Análise da Complexidade de Estruturas e Algoritmos

Árvores

Árvore binária
Árvores binárias semibalanceadas (Árvore AVL, Árvore Red-Black)
Árvore semibalanceadas multivias (Árvore B e B+)
Árvores binárias multichaves (Árvore k-d)
Árvores semibalanceadas multivias multichaves (Árvores k-b-d e b-k-d)

Tabela de espalhamento (hash)

Tratamento de colisões
Funções de espalhamento

Métodos de ordenação

Conceitos básicos, implicações e premissas
Métodos de Complexidade Quadrática
Método por inserção
Método por seleção
Método da bolha
Métodos Avançados e de Complexidade $n \log n$
Método Quicksort

Método Heapsort

Estruturas de dados em arquivo

Acesso direto e Acesso sequencial
Organização de Arquivos de Índices
Métodos indexados seqüenciais, ISAM e VSAM
Indexação por Árvore
Indexação por Multilistas e Lista Invertida

Curso: Ciência da Computação (UFSC)

Disciplina: Estrutura de Dados

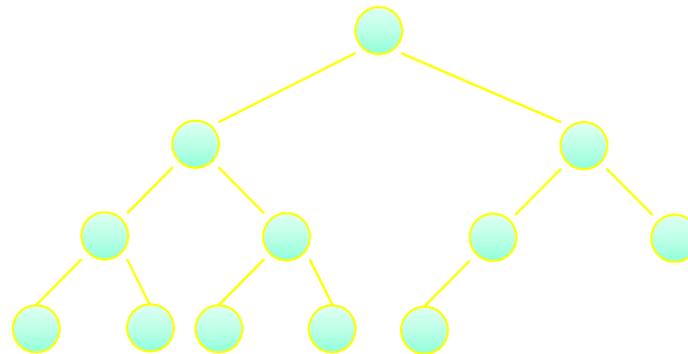
Ensino do *Heapsort*

O que é exigido do aluno?

1. Compreender o funcionamento e a estrutura lógica (código)
2. Aplicação do algoritmo

Funcionamento do *Heapsort*

Primeiro passo: Cria-se uma árvore-binária, justificada a esquerda, onde o nodo Pai é sempre maior que os nodos Filhos.

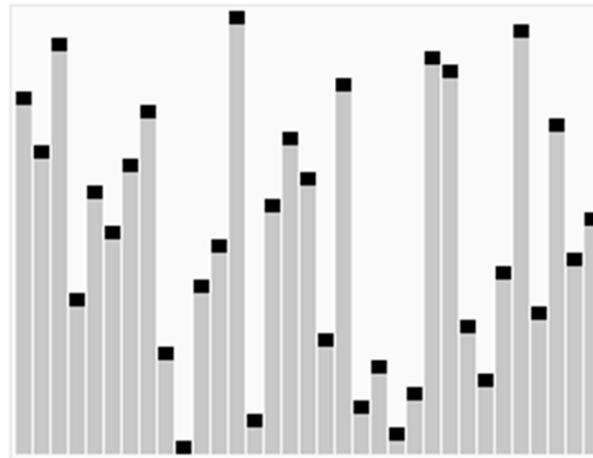


Justificada à
esquerda

Funcionamento do *Heapsort*

Segundo passo: Reanalisar a *árvore-heap*

- Mover o maior valor, que está na raiz, para a última posição
- Reduzir o tamanho da árvore (vetor)
- Reanalisar a árvore a partir da raiz, para manter o nodo Pai sempre com valor maior que os nodos Filhos



[<http://pt.wikipedia.org/wiki/Heapsort>]

**Como o algoritmo de ordenação *Heapsort*
era ensinado?**



Aulas Expositivas

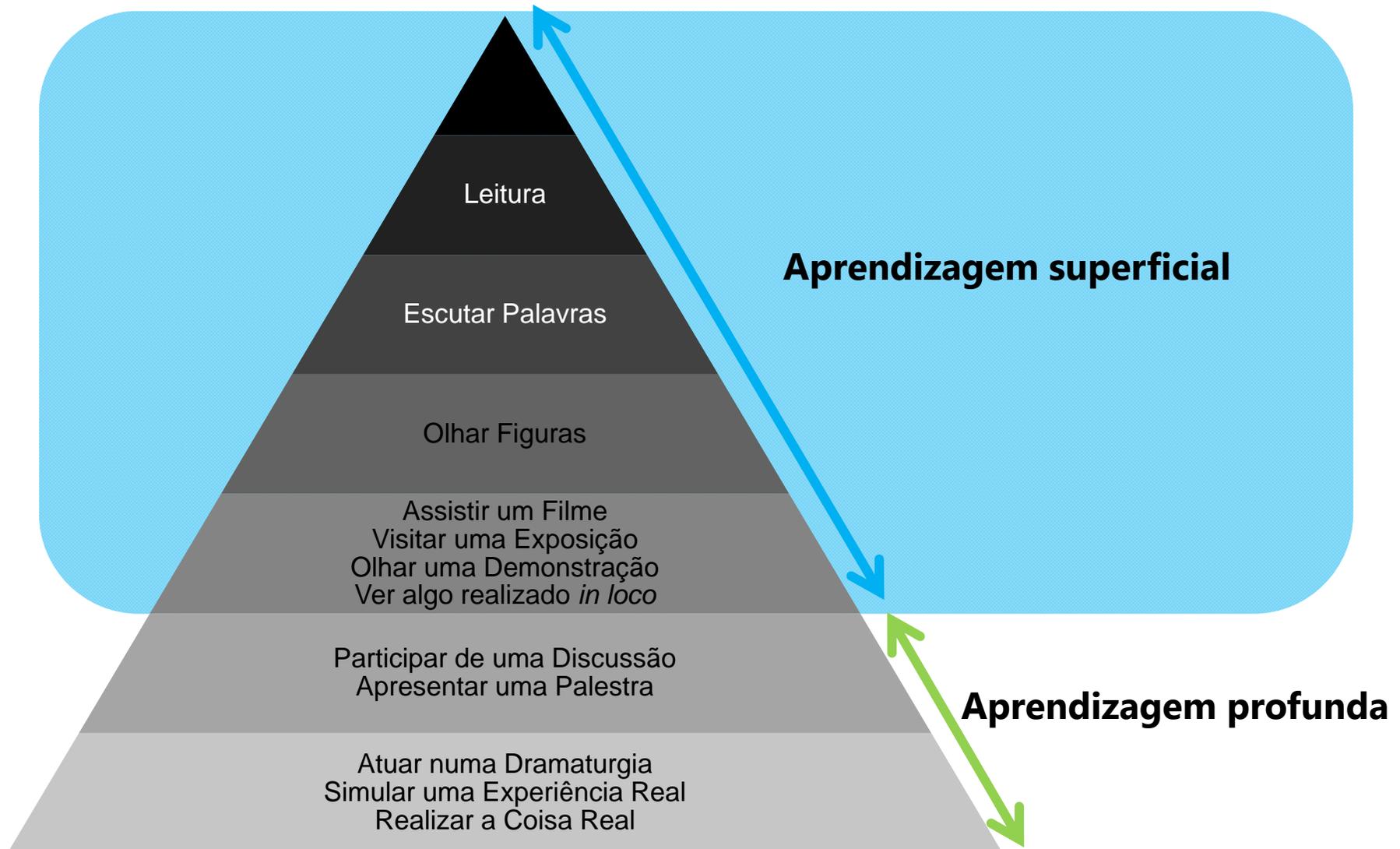




Problemas

- Alunos perdem o interesse
- A aprendizagem é difícil
- Dificuldade de relembrar e saber aplicar o conteúdo

Como proporcionar um ensino mais profundo?



Estratégias instrucionais



[<http://olc.spsd.sk.ca/de/pd/instr/index.html>]

Um **jogo educacional** pode ser definido como:

Jogo

competição (jogo) entre adversários (jogadores) que agem sob restrições (regras) para um objetivo (vencer).

Educacional

projetado especificamente para ensinar um determinado assunto.

[Baseado em C. C. Abt. Serious Games. University Press of America, 2002 & J. V. Dempsey et al. The Instructional Gaming Literature. Technical Report 96-1, University of South Alabama, 1996]

Descrição do jogo SORTIA

- **Objetivo de aprendizagem do jogo:** Reforçar, aprender e aplicar o processo de ordenação de um conjunto de números inteiros.
- **Gênero:** Simulação
- **Participantes:** *single-player*
- **Tipo:** Computador/online
- **Duração:** Em média 20 minutos
- **Feedback:** Ao finalizar o jogo, o jogador recebe *feedback* dos movimentos realizados corretamente e os movimentos errados.
- **Link:** <http://www.inf.ufsc.br/~awangenh/sorting/HeapGame/>

Formação da *Árvore-heap*

localhost/estruturadedados/projeto/

Jogo de Ordenação: Heapsort

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Variável i	92																				
Valores do Vetor	92	74	40	74	69	64	66	64	73	59	1	38	18	23	46	4	5	52	63	27	6
Variável j			40																		
aAux	92																				
N																					

CONTROLE

Iniciar Limpar tabuleiro Finalizar Ajustar Valores Remover a Raiz

RESULTADO

Regras do Jogo

Reanálise da *Árvore-heap*

localhost/estruturadedados/projeto/

Jogo de Ordenação: Heapsort

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Variável i	69																				
Valores do Vetor	69	59	40	64	52	64	66	27	63	6	1	38	18	23	46	4	5	73	74	74	92
Variável j					52																
aAux		59																			
N																		73			

CONTROLE				
Iniciar	Limpar tabuleiro	Finalizar	Ajustar Valores	Remover a Raiz
RESULTADO				
Regms do Jogo				

Final do jogo

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Variável i																					
Valores do Vetor	59	40	5	52	64	66	27	63	6	1	38	18	23	46	4	64	69	73	74	74	92
Variável j																					
aAux																					
N																					

Feedback do final do jogo

CONTROLE				
Iniciar	Limpar tabuleiro	Finalizar	Ajustar Valores	Remover a Raiz
RESULTADO	<pre>--RESULTADO DA ORDENAÇÃO HEAPSORT-- ERRO: a ordenação não foi realizada corretamente. 1) VETOR ORIGINAL ([40][73][38][4][1][92][46][5][74][59][69][64][18][23][66][64][74][52][63][27][6]) 2)VISUALIZAÇÃO DO VETOR ORDENADO CORRETAMENTE ([1][4][5][6][18][23][27][38][40][46][52][59][63][64][64][66][69][73][74][74][92]) 3)PASSOS CORRETOS PASSO:0 AJUSTAR - (74, 5); PASSO:1 AJUSTAR - (66, 46); PASSO:2 AJUSTAR - (69, 1); PASSO:3 AJUSTAR - (74, 4); PASSO:4 AJUSTAR - (64, 4); PASSO:5 AJUSTAR - (92, 38); PASSO:6 AJUSTAR - (64, 38); PASSO:7 AJUSTAR - (74, 73); PASSO:8 AJUSTAR - (74, 73); PASSO:9 AJUSTAR - (92, 40); *****PASSO ERRADO***** PASSO:10 REMOVER DA RAÍZ - (6, 92); A seguir serão mostrados os passos que deveriam ser realizados **ERRO**:10 AJUSTAR - (66, 40); **ERRO**:11 AJUSTAR - (46, 40); **ERRO**:12 REMOVER DA RAÍZ - (6, 92); **ERRO**:13 AJUSTAR - (74, 6); **ERRO**:14 AJUSTAR - (74, 6); **ERRO**:15 AJUSTAR - (73, 6); **ERRO**:16 AJUSTAR - (63, 6);</pre>			
	Regras do Jogo			

Regras do jogo

localhost/estruturadedados/projeto/regras.html

REGRAS DO JOGO HEAPSORT

Neste jogo vamos simular a execução do algoritmo Heapsort. Este algoritmo utiliza uma árvore-binária balanceada a esquerda para realizar a ordenação.

Os passos executados pelo algoritmo são:

- 1) Transformar o vetor em uma árvore-heap;
- 2) Inversão dos valores da raiz com o último valor do vetor, considerando que o maior valor do vetor está na raiz;
- 3) Transformar o vetor novamente em uma árvore-heap e realizar a inversão dos valores da raiz com o último elemento da árvore (última folha). Esta regra será repetida até o vetor estiver completamente ordenado.

Entendendo o tabuleiro

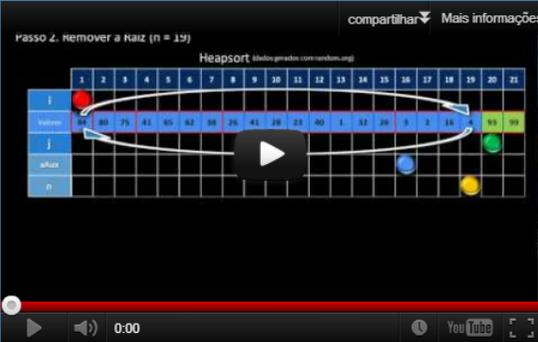
O tabuleiro possui 6 linhas, onde:

- 1) Na primeira linha estão os índices do vetor.
- 2) Na segunda linha temos o ponteiro "Variável i", a qual sempre será inicializada na posição "N" DIV 2" do vetor.
- 3) Na terceira linha temos os valores do vetor, os quais são escolhidos aleatoriamente.
- 4) Na quarta linha temos o ponteiro "Variável j" que é responsável pela verificação dos nós filhos.
- 5) Na quinta linha temos o ponteiro auxiliar "aAux" que identifica qual é o nó pai da subárvore.
- 6) Na sexta linha temos o tamanho do vetor, a qual chamamos de "N". Este vetor vai decrementar de 21 até 1 na medida que o vetor vai sendo ordenado.

Botões do jogo

- 1) "Iniciar": começar o jogo de ordenação.
- 2) "Limpar tabuleiro": limpa os valores do vetor e posiciona os ponteiros nas posições iniciais do jogo.
- 3) "Finalizar": finaliza o jogo e mostra o resultado da ordenação.
- 4) "Ajustar Valores": faz a inversão dos valores do ponteiro "aAux" e o ponteiro "Variável j".
- 5) "Remover a Raiz": faz a inversão do valor da raiz (primeira posição do vetor) com a última folha. E também reduz o tamanho do vetor (n-1).

Veja abaixo a simulação do Heapsort:



compartilhar Mais informações

Passo 2. Remover a Raiz (n = 17)

Heapsort (video-grafico.com/random.org)

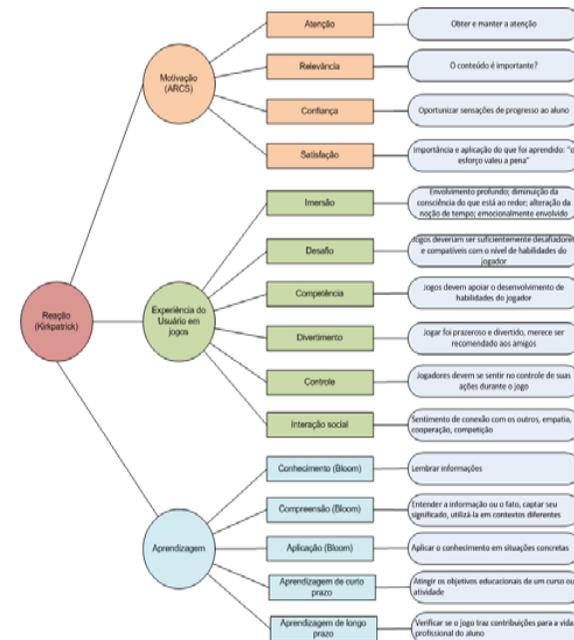
INCoD

Aplicação e avaliação

Objetivo da avaliação: Avaliar a motivação, experiência do usuário e a aprendizagem do ponto de vista do aluno.

Design de pesquisa: estudo de caso - *one-shot post-test* (X O)

Modelo MEEGA utilizado para avaliação



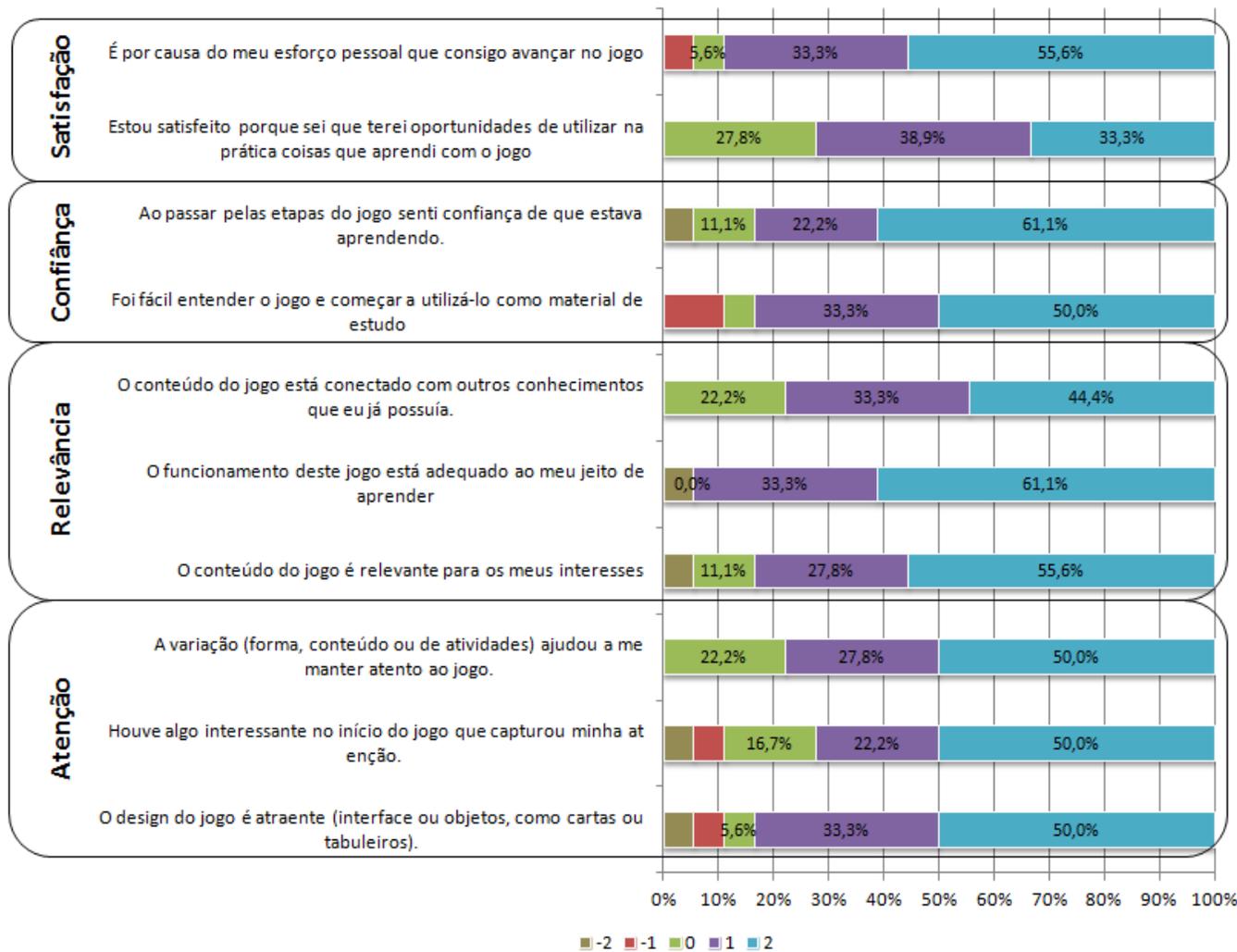
[SAVI, R., WANGENHEIM, C. G. von e BORGATTO, A. Um Modelo de Avaliação de Jogos Educacionais na Engenharia de Software. 25th Brazilian Symposium on Software Engineering (SBES)/São Paulo/Brazil, 2011]

Aplicação

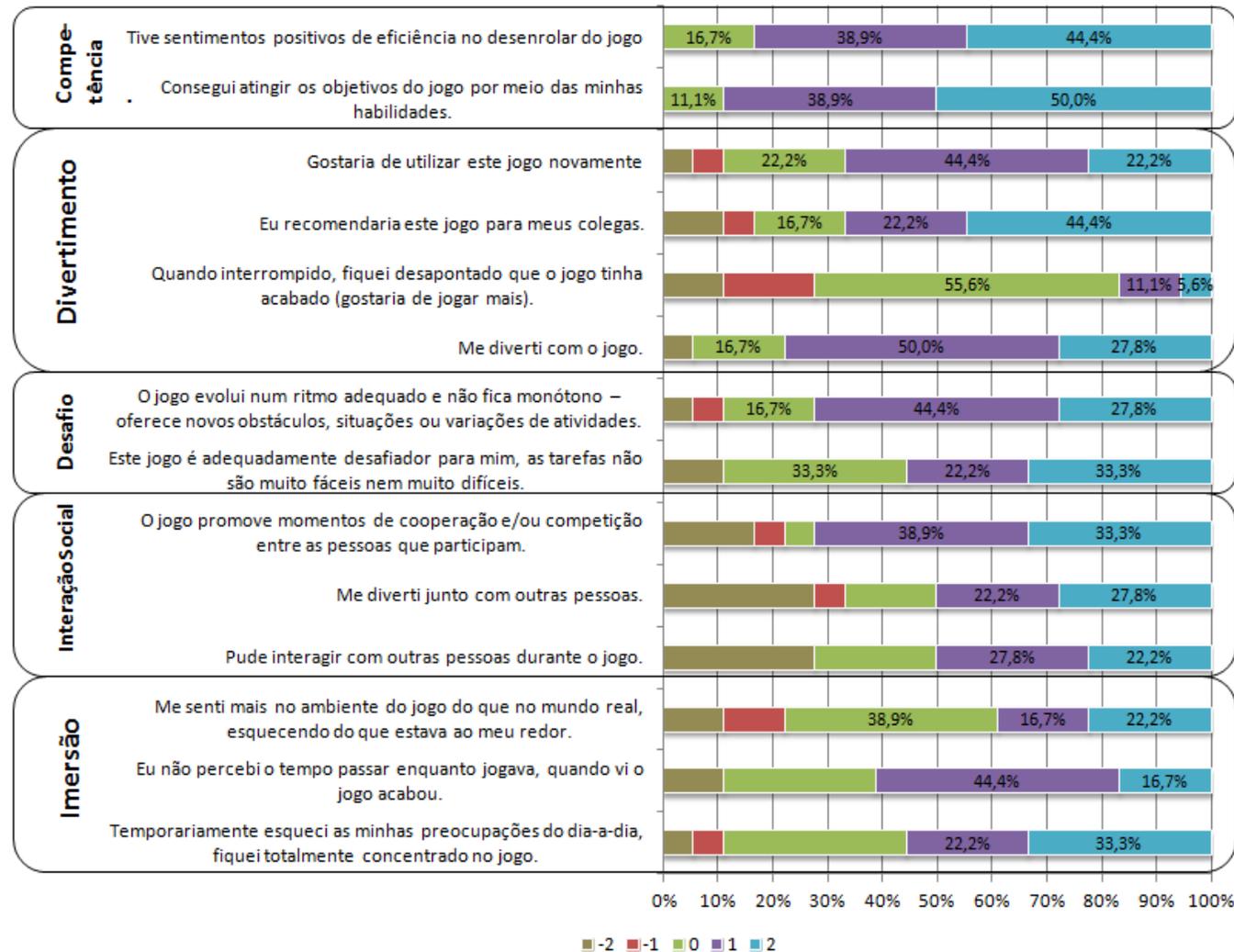


INE5408 –Estrutura de Dados
Bacharelado em Ciências da Computação
Ano/Semestre: 2012.1
Número de alunos: 18

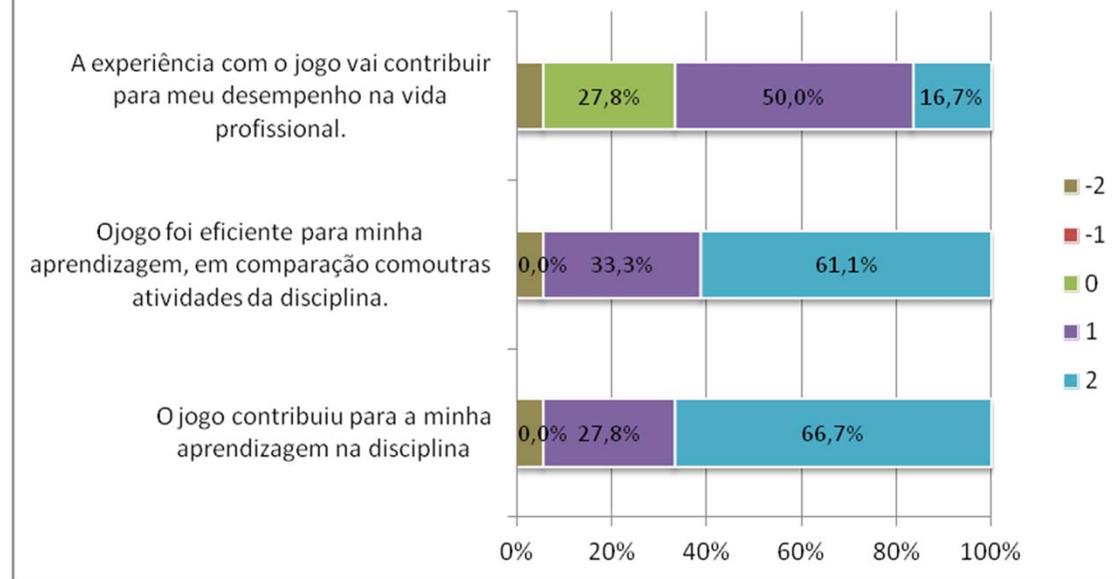
Motivação



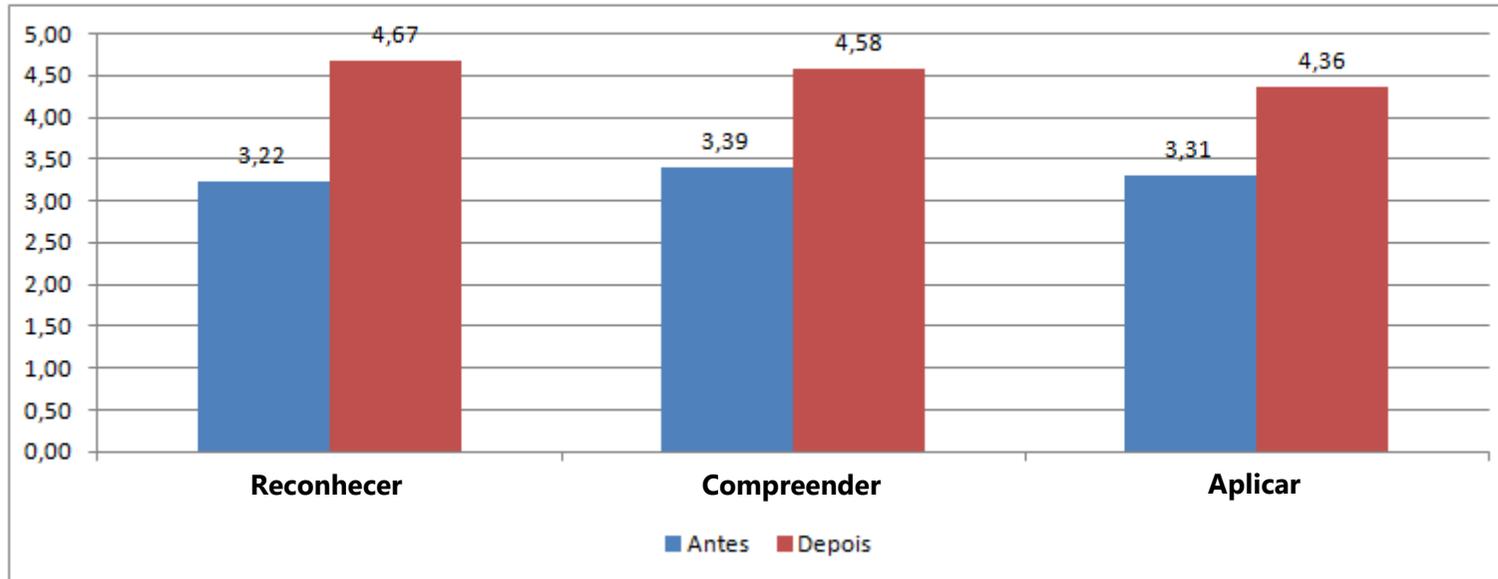
Experiência do Usuário



Aprendizagem



Objetivos de Aprendizagem



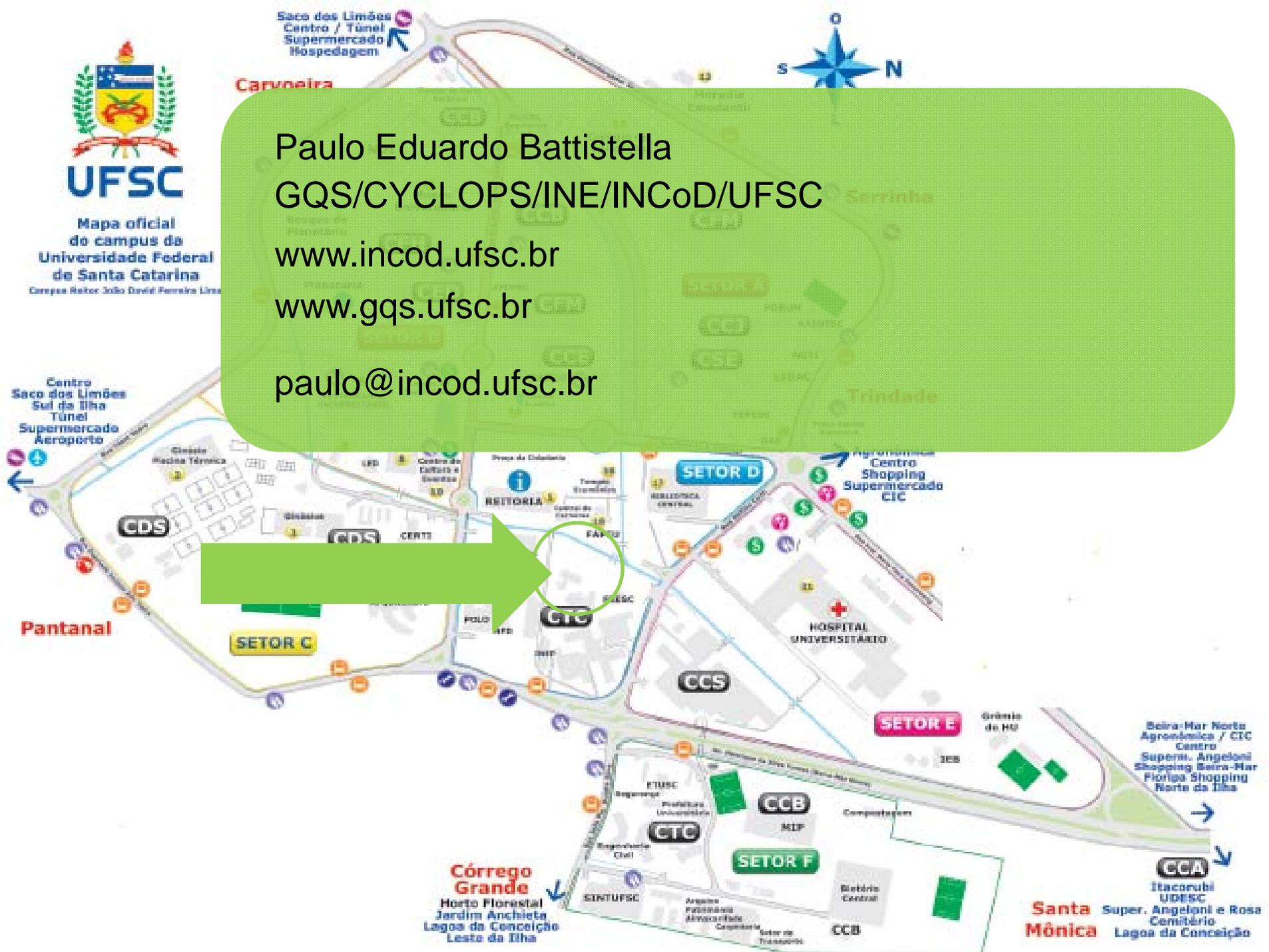
Considerações Finais

- Esta primeira pesquisa indica que o jogo contribuiu positivamente para a aprendizagem dos alunos.
- **Motivação**
 - Os alunos identificaram que o jogo é relevante para seu aprendizado
 - O jogo prendeu a atenção dos alunos em todo o período
- **Experiência do Usuário**
 - Mais de 75% dos alunos se divertiram ao jogar o jogo
- **Interação Social**
 - Ponto fraco do jogo, por tratar-se de um jogo *single-player*
- **Aprendizagem**
 - Mais de 90% dos alunos avaliaram o jogo como eficiente, em relação a outras estratégias instrucionais
- A partir deste estudo de caso, o jogo foi inserido no plano de ensino da disciplina de estruturas de dados no INE/UFSC



Mapa oficial
do campus da
Universidade Federal
de Santa Catarina
Carrega Reitor João David Ferreira Lima

Paulo Eduardo Battistella
GQS/CYCLOPS/INE/INCoD/UFSC
www.incod.ufsc.br
www.gqs.ufsc.br
paulo@incod.ufsc.br





Atribuição-Usu Não-Comercial-Compartilhamento pela Licença 2.5 Brasil

Você pode:

- copiar, distribuir, exibir e executar a obra
- criar obras derivadas

Sob as seguintes condições:

Atribuição — Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.

Usu Não-Comercial — Você não pode utilizar esta obra com finalidades comerciais.

Compartilhamento pela mesma Licença — Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.

Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/br/> ou mande uma carta para Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.