

---

# Nova proposta de atividades desplugadas no ensino de algoritmos de ordenação linear na educação superior

## RESUMO

**Fernando Tomaz da Silva**

[fernando.tomaz@dce.ufpb.br](mailto:fernando.tomaz@dce.ufpb.br)

Universidade Federal da Paraíba - campus IV

**Flávia Veloso Costa Souza**

[flavia@dce.ufpb.br](mailto:flavia@dce.ufpb.br)

Universidade Federal da Paraíba - campus IV

**Wagner Emanuel Costa**

[wagner@dce.ufpb.br](mailto:wagner@dce.ufpb.br)

Universidade Federal da Paraíba - campus IV

O objetivo deste trabalho é elaborar e aplicar atividades desplugadas no ensino superior com alunos da disciplina de Estrutura de Dados II do curso Bacharelado em Sistemas de Informação da Universidade Federal da Paraíba - UFPB Campus IV. Neste trabalho são descritas duas oficinas já realizadas para apoiar no ensino de algoritmos de ordenação linear. A abordagem metodológica foi a investigação quantitativa que investigou a aprendizagem dos alunos através de exames escritos e teste estatísticos sob os resultados. Os resultados indicam que o impacto da metodologia é significativo.

**PALAVRAS-CHAVE:** Novel atividade desplugada. Ensino de algoritmo. Estrutura de Dados.

---

Trabalho de Conclusão de Curso apresentado pelo discente Fernando Tomaz da Silva sob a orientação da professora Flávia Veloso Costa Sousa e o professor Wagner Emanuel Costa como parte dos requisitos para obtenção do grau de Licenciado em Ciência da Computação na UFPB Campus IV

---

## 1. INTRODUÇÃO

O presente artigo apresenta um relato de experiência da criação e uso de atividades de computação desplugada na educação superior. As atividades tratam do assunto de algoritmo de ordenação linear. O artigo apresenta como formato metodológico uma análise quantitativa que investigou a aprendizagem dos alunos através de exames escritos e teste estatísticos.

O uso da computação desplugada nos últimos anos tem se tornado um método que pode contribuir no processo de ensino de conceitos de Ciência da Computação, possibilitando uma nova experiência no processo de ensino e aprendizagem. De acordo Bell et al. (2011), o propósito é desenvolver nos alunos habilidades de comunicação, resolução de problemas e criatividade num contexto significativo. Esse método de ensino pode ser utilizado desde a educação básica até a educação superior (VIEIRA et al., 2013).

A computação desplugada é um conjunto de atividades lúdicas desenvolvidas com o objetivo de ensinar os conceitos da Ciência da Computação sem o uso de computadores. Ela foi inicialmente proposta por Tim Bell, Ian H. Witten e Mike Fellows (BELL et al., 2011). No livro dos autores citado tem-se por exemplo a seguinte prática “Colorindo com Números - Representação de Imagens” por pixel, cuja finalidade é mostra como uma imagem de um computador pode ser representada por números. Essa atividade necessita uma folha de atividade gradeado para representa uma matriz de pixels, disponibilizada no livro, que apresenta algumas imagens que podem ser representadas utilizando este método. A atividade trabalha conceitos matemáticos de números, pixels e contagens, responsáveis direto pela reprodução das telas dos computadores. Tendo como faixa etária um público-alvo a partir de 7 anos. A independência de recursos de hardware e software que observamos no exemplo é uma das vantagens de abordagens dessa natureza.

Como exemplo do uso da computação desplugada no ensino superior, descreve-se o trabalho de Costa et al. (2017). O artigo apresenta várias oficinas com o objetivo de apoiar o ensino dos algoritmos de ordenação por comparação e tabela hash com endereçamento fechado. As oficinas foram aplicadas para o ensino superior. A primeira oficina abordou algoritmos de ordenação, nesta oficina, a atividade do algoritmo selection sort foi feita com ajuda de 10 alunos segurando cartões numerados de maneira aleatória. Em uma lista com 10 posições produzidas na sala de aula, os alunos tinham que ordenar seguindo o passo do algoritmo. As atividades dos algoritmos de ordenação insertion sort, mergesort e quicksort são semelhantes. Logo após a aplicação de cada atividade desplugada, os alunos eram divididos em grupos, cada grupo recebeu uma cartolina e foram orientados a representar a lógica de funcionamento de cada algoritmo.

A segunda oficina de Costa et al. (2017), abordou tabela hash para o desenvolvimento da atividade foram selecionados 7 alunos simbolizando os dados a serem inseridos na tabela que era representada por cadeiras. Nesta atividade os alunos precisavam resolver uma função de hash e de acordo com o resultado descobrir em qual posição da cadeira seria inserido. Caso houvesse colisão, as pessoas na fila sentavam uma cadeira para trás e a que estava para entrar na tabela sentava na frente, simulando o encadeamento. A eficácia das técnicas foi examinada usando tanto análise de dados qualitativa quanto quantitativa. Os

---

resultados apontam que as atividades desplugadas propostas nas oficinas tiveram resultados satisfatórios na aprendizagem dos alunos.

Outro exemplo do uso da computação desplugada no ensino superior é o trabalho de Andriani et al. (2015). O artigo apresenta uma experiência de uma dinâmica de grupo intitulada fluxograma humano elaborado para a Semana de Integração de Engenharia de Computação (SIECOMP) da Universidade Estadual de Feira de Santana, para a introdução de aspectos da computação como forma de familiarizar os estudantes do primeiro semestre com o funcionamento dos algoritmos. A dinâmica trabalhou dois algoritmos, o primeiro para ensinar cálculo de módulo de um número e o segundo para gerar números da sequência Fibonacci. Na dinâmica, os alunos foram representantes dos elementos computacionais, ou seja, alguns deles foram variáveis, outros representaram diferentes instruções, usuário que informava entradas para o programa e recebia as saídas e o responsável por percorrer as instruções, seguindo o fluxo criado pelo programador e interpretando o funcionamento passo a passo. Por isso, o nome sugestivo de fluxograma humano, já que o fluxograma computacional é representado e executado por pessoas e suas ações.

Os relatos de atividades desplugadas na literatura estão principalmente relacionados com o uso de atividades proposta no livro de “Ciência da Computação Desplugada” de Bell et al. (2011). Como o trabalho de Bell et al. (2011) tem como alvo ensino básico e médio, percebe-se um cenário propício de elaboração de novas atividades desplugadas para educação superior.

Este artigo está organizado da seguinte forma: a Seção 2 descreve a metodologia adotada. A Seção 3 apresenta o planejamento das oficinas. A Seção 4 mostra a descrição das oficinas. A Seção 5 são relatados os resultados do teste estatístico pareado. Por fim, a Seção 6 apresenta as considerações finais e perspectivas de trabalhos futuros.

## **2. METODOLOGIA**

Esta pesquisa foi desenvolvida na Universidade Federal da Paraíba - UFPB Campus IV que é localizada na cidade de Rio Tinto - PB. Os sujeitos da pesquisa foram 25 alunos da disciplina de Estrutura de Dados II do curso Bacharelado em Sistemas de Informação do período 2017.1.

Inicialmente foram selecionados os conteúdos que seriam trabalhados com o apoio da computação desplugada. Após essa etapa, um conjunto de atividades desplugadas foi elaborado pelo pesquisador para apoiar no ensino de algoritmos de ordenação linear. As aplicações das atividades aconteceram em agosto de 2017.

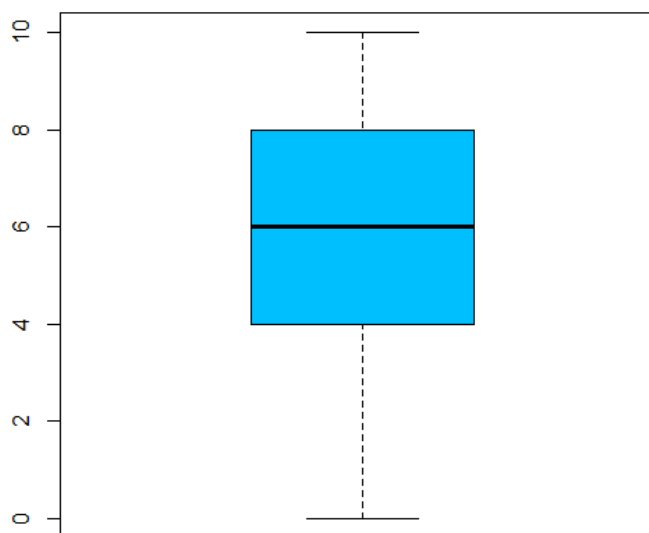
Como instrumentos de pesquisa foram utilizados pré-teste e pós-teste, box plot, teste estatístico pareado de Wilcoxon. O pré-teste teve o objetivo de diagnosticar o conhecimento prévios dos alunos a respeito dos conteúdos abordados e o pós-teste verificar se depois das atividades teve alguma influência na aprendizagem dos alunos. Ou seja, o pré-teste serve para isolar os efeitos das aulas anteriores daqueles que a atividades ira causar. Estes últimos são avaliados pelas diferenças das notas entre pré-teste e pós-teste de cada aluno.

Os resultados dos testes, pré e pós, são exibidos também via box plots. Segundo Morettin (2009), o box plot dá uma ideia da posição, dispersão,

assimétria, caudas e dados discrepantes. O box plot é um gráfico construído com base no resumo dos cinco números, constituído por: valor mínimo, primeiro quartil, mediana, terceiro quartil e valor máximo. O gráfico é formado por uma caixa construída paralelamente ao eixo da escala dos dados (pode ser horizontal ou vertical). Essa caixa vai desde o primeiro quartil até o terceiro quartil e nela traça-se uma linha na posição da mediana. Essa caixa, que descreve os 50% centrais da distribuição, é comum a todas as variantes do box plot. Na variante que usa efetivamente o resumo dos cinco números, traça-se uma linha paralela à escala que vai de cada extremidade da caixa ao correspondente valor extremo dos dados.

Na figura 1, tem-se um box plot como ilustração. Observa-se nele, a linha central, mais escura, o valor da mediana, a base e o topo da caixa o primeiro e terceiro quartil, enquanto que as extremidades inferiores e superiores apontam o valor mínimo e máximo. No exemplo, a extremidade inferior está no número 0, o primeiro quartil está no número 4, a mediana está no número 6, o terceiro quartil está no número 8 e a extremidade superior está no número 10.

Figura 1 – Exemplo de formato de um box plot.



(Fonte: Própria, 2017)

Para analisar as notas obtida no pré-teste e pós-teste, foi utilizado teste estatístico pareado de Wilcoxon. Dentre os testes estatísticos destaca-se dois testes para populações pareadas, o t de Student e o teste de Wilcoxon. A escolha correta depende da análise dos dados. O teste t de Student baseia-se na suposição de que as populações sejam normais. Uma violação dessa suposição não permite o uso confiável deste teste. Neste caso, o teste de Wilcoxon é a ferramenta mais confiável. Como é observado nas seções seguintes, o teste de Shapiro-Wilk indicou ausência de normalidade nos resultados logo, o teste de Wilcoxon foi o eleito para uso.

### 3. OFICINAS

A definição dos conteúdos que seriam utilizados nas oficinas foi organizado junto com o professor da disciplina de Estrutura de Dados. Os conteúdos

---

escolhidos foram: algoritmos de ordenação linear (counting sort, radix sort e bucket sort). A subseção 3.1 são descritos os algoritmos de ordenação linear.

### 3.1 Descrição dos algoritmos de ordenação linear

#### 3.1.1 Counting sort

O counting sort (ou ordenação por contagem) não é uma ordenação por comparação. A ordenação por contagem pressupõe que cada um dos  $n$  elementos de entrada é um inteiro no intervalo de 1 a  $k$ , para algum inteiro  $k$ .

A ideia básica da ordenação por contagem é determinar, para cada elemento de entrada  $x$ , o número de elementos menores que  $x$ . Essa informação pode ser usada para inserir o elemento  $x$  diretamente em sua posição no arranjo de saída. Por exemplo, se há 17 elementos menores que  $x$ , então  $x$  é colocado na posição de 18 na saída. Esses contadores são atualizados para lidar com a situação na qual vários elementos têm o mesmo valor, pois não pode inserir todos eles na mesma posição.

A ordenação por contagem é ilustrada na figura 2. A entrada  $A$  é um arranjo  $A[1 .. n]$  e, portanto, comprimento  $[A] = n$ . O arranjo  $B[1 .. n]$  contém a saída ordenada, e o arranjo  $C[0 .. k]$  fornece um espaço de armazenamento temporário. Nas linhas 2 a 4 do pseudocódigo, temos um laço de repetição que inicializar os contadores inicialmente com zero. Após a inicialização dos contadores, nas linhas 6 e 7 conta-se quantos elementos possui no arranjo  $A$ . Nas linhas 11 e 12, conta-se quantos números menores ou iguais a cada valor acontece no arranjo  $C[0 .. k]$ . Finalmente, no laço de repetição entre as linhas 14 e 16, cada elemento do arranjo  $A$  é inserido em sua posição correta no arranjo  $B$ .

Figura 2 – Pseudocódigo do algoritmo de ordenação counting sort.

---

Counting-Sort( $A, B, k$ )

---

**Entrada:** Vetor  $A$  de tamanho  $n$ . Todos os valores em  $A$  são menores que  $k$

**Saída:** O Vetor  $B$  é o Vetor  $A$  ordenado

```
1: %Inicialize os contadores
2: para  $i \leftarrow 0$  até  $k$  faça
3:    $C[i] \leftarrow 0$ 
4: fim para
5: %Conte os elementos
6: para  $j \leftarrow 1$  até  $n$  faça
7:    $C[A[j]] \leftarrow C[A[j]] + 1$ 
8: fim para
9: %Agora  $C[i]$  contém o número de elementos em  $A$  iguais a  $i$ 
10: %A próxima etapa consiste em usar essa informação para contar elementos menores ou iguais a  $i$ 
11: para  $i \leftarrow 1$  até  $k$  faça
12:    $C[i] \leftarrow C[i] + C[i - 1]$ 
13: fim para
14: para  $j \leftarrow n$  até 1 faça
15:    $B[C[A[j]]] \leftarrow A[j]$ 
16:    $C[A[j]] \leftarrow C[A[j]] - 1$ 
17: fim para
18: retorne  $B$ 
```

---

---

Na figura 2, observa-se que não há comparação entre elementos de entrada em qualquer lugar. Em vez disso, a ordenação por contagem utiliza os valores dos elementos para efetuar a indexação em um arranjo.

Uma propriedade importante da ordenação por contagem é o fato de ela ser estável: números com o mesmo valor aparecem no arranjo de saída na mesma ordem em que se encontram no arranjo de entrada. A estabilidade da ordenação por contagem é importante por outra razão: a ordenação por contagem é usada frequentemente como sub-rotina em radix sort. A estabilidade da ordenação por contagem é crucial para a correção de radix sort.

### 3.1.2 Radix sort

O radix sort (ou ordenação da raiz) é um algoritmo de ordenação estável que pode ser usado para ordenar itens que estão identificados por chaves únicas. Cada chave é uma cadeia de caracteres ou número, e o radix sort ordena estas chaves em qualquer ordem relacionada com a lexicografia.

A ideia básica do radix sort é ordenar uma coluna por vez. Ordena primeiro sobre o dígito menos significativo colocando em uma única pilha. Então, a pilha inteira é ordenada novamente sobre o segundo dígito menos significativo e recombinados de maneira semelhante. O processo continua até sobre todos os dígitos.

A figura 3 mostra o pseudocódigo do radix sort. O processo a seguir supõe que cada elemento no arranjo de  $n$  elemento  $A$  tem  $d$  dígitos, onde o dígito 1 é o dígito de mais baixa ordem e o dígito  $d$  é o dígito de mais alta ordem. O radix sort depende de um outro procedimento de ordenação estável, o counting sort.

Figura 3 – Pseudocódigo do algoritmo de ordenação radix sort.

---

**Algoritmo 2 Radix Sort( $A, d$ )**

---

- 1: **para**  $i \leftarrow 1$  até  $d$  **faça**
  - 2:   Usar uma ordenação estável para ordenar o arranjo  $A$  sobre o dígito  $i$
  - 3: **fim para**
- 

### 3.1.3 Bucket sort

O bucket sort (ou ordenação por balde) funciona em tempo linear. Como a ordenação por contagem, o bucket sort é rápido porque pressupõe algo sobre a entrada. Enquanto a ordenação por contagem presume que a entrada consiste em inteiros em um intervalo pequeno, bucket sort presume que a entrada é gerada por um processo aleatório que distribui elementos uniformemente sobre o intervalo  $[0,1)$ .

A ideia de bucket sort é dividir o intervalo  $[0,1)$  em  $n$  subintervalos de igual tamanho, ou baldes, e depois distribuir os  $n$  números de entrada sobre os baldes. Tendo em vista que as entradas são uniformemente distribuídas sobre  $[0,1)$ , não espera que muitos números caiam em cada balde. Para produzir a saída, simplesmente ordenamos os números em cada balde, e depois percorre os baldes em ordem, listando os elementos contidos em cada um.

---

O bucket sort pressupõe que a entrada é um arranjo de  $n$  elementos,  $A$ , onde os valores são dados ponto flutuante com distribuição uniforme entre zero e um. O código exige um arranjo auxiliar  $B[0 \dots n-1]$  de listas ligada (baldes) e assume que existe um mecanismo para manter tais lista.

A figura 4 mostra a operação do bucket sort, nas linhas 2 e 3, os elementos do arranjo  $A$  são inserido em baldes. Nas linhas 5 e 6, os elementos são ordenados em cada balde utilizando outro algoritmo de ordenação, como por exemplo o insertion sort e colocando na ordem adequada no arranjo  $B$ . Finalmente, a linha 8, tem-se a instrução de concatenar os baldes de forma a ordenar os elementos.

Figura 4 – Pseudocódigo do algoritmo de ordenação bucket sort.

---

**Algoritmo 3** Bucket Sort( $A$ )

---

```
1:  $n \leftarrow$  comprimento de  $A$ 
2: para  $i \leftarrow 1$  até  $n$  faça
3:   Inserir  $A[i]$  na lista  $B[\lfloor nA[i] \rfloor]$ 
4: fim para
5: para  $i \leftarrow 0$  até  $n - 1$  faça
6:   Ordenar a lista  $B[i]$  com ordenação por inserção
7: fim para
8: Concatenar as listas  $B[0], B[1], \dots, B[n - 1]$  juntas em ordem.
```

---

Na subseção 3.2 a seguir, estão os procedentes de planejamento das oficinas, contextualização do material utilizado nas atividades e as propostas de atividades desenvolvidas para os algoritmos de ordenação linear.

### 3.2 Planejamento da oficina de algoritmo de ordenação

Inicialmente foi feita uma reunião para planejar as atividades desplugadas dos algoritmos de ordenação, na discussão com o professor da disciplina de Estrutura de Dados, surgiu a ideia de associar dados mais elaborados do que simples números e ainda assim ter uma ferramenta que pudesse ser levada a turmas tanto do ensino superior quanto do ensino médio.

Segundo Hamming (2012), “O propósito da computação é insight, não números.”. Nessa perspectiva, durante as discussões foi sugerido utilizar cartas do jogo Pokémon Estampas Ilustradas. Diversas cartas desse jogo descrevem várias características dos personagens, Pokémon. Como essa variedade de características permite vários critérios de ordenação, e além disso, é um universo particularmente famoso no mundo presente, a sugestão foi acolhida. Após essa etapa, foram elaboradas as atividades desplugadas de cada oficina.

O planejamento das oficinas foi dividido em dois momentos. O primeiro para familiariza com os instrumentos envolvido na pratica de desplugada, e o primeiro dos algoritmos, counting sort. O segundo momento, com os alunos já familiarizados com os instrumentos, aborda-se radix sort e bucket sort. A definição do counting sort como primeiro é devido a dependência do radix sort para outro algoritmo de ordenação. Cada encontro tem duração de 1 hora 40 minutos.

### 3.2.1 Universo Pokémon

O Pokémon começou a ficar famoso através de um videogame lançado para a Nintendo chamado Game Boy em 1996. Logo após, foi lançada uma série animada Pokémon: Indigo League, inspirada no jogo, essa série virou uma “febre” no mundo todo. A série mostra as aventuras de um jovem e audacioso Treinador chamado Ash Ketchum e seu monstro de estimação, Pokémon, Pikachu em suas viagens pelo mundo para se torna um mestre Pokémon.

No universo Pokémon, os monstros de estimação, Pokémon, são criaturas de todas as formas e tamanhos que convivem com os humanos na natureza. Os Pokémon são inspirados em animais, plantas, mitos e objetos que possuem poderes especiais relacionados com os elementos que fazem parte de sua natureza. Os Pokémon crescem e ganham experiência, podendo até mesmo evoluir para Pokémon mais fortes. A maioria dos Pokémon apresenta três estágios de evolução. Isto significa que os Pokémon podem vir a passar por até duas evoluções.

Desde o lançamento, a série Pokémon popularizou no mundo todo. Hoje em dia, os produtos Pokémon incluem jogo de cartas Pokémon Estampas Ilustradas, novas séries de TV animada, filmes, jogos online, brinquedos e outros de forma que é algo reconhecido por crianças e adultos.

### 3.2.2 Jogo Pokémon Estampas Ilustradas

O jogo Pokémon Estampas Ilustradas é um jogo bastante conhecido, onde jogadores tem a possibilidade de competirem em diferentes faixas etárias contra outros jogadores em três eventos: Campeonato Regional, Campeonato Nacional e o último evento é o World Championships - Campeonato Mundial Pokémon. Para se jogar utiliza-se um deck de cartas colecionáveis. Cada carta representa entidades ou elementos do universo Pokémon, que podem ser Pokémon, itens ou entidades. Cada Pokémon tem um tipo. No total são 11 os tipos possíveis de um Pokémon, como ilustrado na Figura 5.

Figura 5 – Tipos de Pokémon.



(Fonte: [bulbapedia.bulbagarden.net/wiki/Type\\_\(TCG\)](http://bulbapedia.bulbagarden.net/wiki/Type_(TCG)), 2017)

Na figura 6 temos um exemplo de uma carta de Pokémon. Nela podemos observar várias informações. Na parte superior, as principais características de um Pokémon, como por exemplo o estágio de evolução, o nome do Pokémon, o PS (pontos de vida) e o tipo de Pokémon. No exemplo, a carta refere-se ao Pokémon Decidueye, ele é um Pokémon de estágio 2, tem 140 PS e é do tipo planta.



Figura 6 – Partes de uma carta de Pokémon.



(Fonte: Própria, 2017)

No contexto do jogo, um personagem Pokémon pode evoluir, ganhar novas habilidades e características, mediante outra carta Pokémon a ela relacionadas. As cartas de evolução são classificadas como sendo: Pokémon básico, Pokémon estágio 1 e Pokémon estágio 2. Um Pokémon estágio 1 entra em jogo quando é associado a carta do básico listada. Ao passo que um Pokémon estágio 2 deve ser associado a uma carta estágio 1, nesse listada. No exemplo da figura 7, abaixo, oferece um exemplo de cartas de estágio de evolução. A carta Cosmoem estágio 1 é uma evolução da carta Cosmog básico e a carta Solgaleo estágio 2 é uma evolução da carta Cosmoem estágio 1.

Figura 7 – Ilustração das cartas de Pokémon básico, estágio 1 e estágio 2.



(Fonte: Própria, 2017)

---

No site ([https://bulbapedia.bulbagarden.net/wiki/Category:Theme\\_Decks](https://bulbapedia.bulbagarden.net/wiki/Category:Theme_Decks)) Bulbapedia reunir uma grande variedade de cartas do jogo Pokémon Estampas Ilustradas. Essas cartas podem ser impressas e utilizadas em atividade de ensino caso de indisponibilidade no comércio. Uma outra opção é elaborar um conjunto específico de cartas adequada a audiência. No apêndice E há exemplos de como fazer.

### **3.2.3 Proposta de atividade desenvolvida para o counting sort**

Para o planejamento da atividade do counting sort, utilizou-se como critério de ordenação 3 tipos de cartas de Pokémon diferentes (Básico, Estágio 1 e Estágio 2) e cartas de Energia. As categorias das cartas “Básico” foi associada ao número 1, “Estágio 1” ao número 2, e as categorias das cartas “Estágio 2” ao número 3, por fim as cartas de “Energia” ao número 4. Como cada categoria representa um número, é possível ordenar as cartas segundo seu tipo.

Nesta atividade cada grupo recebe 10 cartas de Pokémon, cartolina, lápis e post-it. O objetivo da atividade é desenhar na cartolina o funcionamento de ordenação do algoritmo lecionado. Observando o comportamento do método de ordenação.

### **3.2.4 Proposta de atividade desenvolvida para o radix sort**

Para a segunda atividade, relativa ao radix sort, utiliza-se 2 critérios de ordenação: o primeiro utiliza-se os tipos de cartas Pokémon (Planta, Psíquico, Metal e Incolor) o tipo “Planta” foi associado ao número 1, “Psíquico” ao número 2, “Metal” ao número 3 e o tipo “Incolor” ao número 4. O segundo critério de ordenação, utiliza-se as cartas de estágio de evolução (Básico, Estágio 1 e Estágio 2) a categoria “Básico” foi assoado ao número 1, “Estágio 1” ao número 2 e o “Estágio 2” ao número 3.

Nesta atividade cada aluno recebe 15 cartas de Pokémon e são instruídos a ordenar as cartas pelo tipo e, obedecendo a estabilidade do algoritmo de ordenação, ordenar as cartas segundo o critério de estágio de evolução. Observando o comportamento do método de ordenação.

### **3.2.5 Proposta de atividade desenvolvida para o bucket sort**

A última atividade proposta, relativa ao bucket sort, utiliza-se os valores PS de cartas Pokémon, com PS entre 10 e 90. Uma seleção de cartas foi extraída do site Bulbapedia ([bulbapedia.bulbagarden.net/wiki/Category:Theme\\_Decks](https://bulbapedia.bulbagarden.net/wiki/Category:Theme_Decks)) para garantir que a distribuição de valores PS entregue aos alunos se aproxime de uma distribuição uniforme entre 10 e 90.

Nesta atividade cada grupo recebe uma seleção de 10 cartas e um tabuleiro numerado de 0 a 9 simulando o vetor balde (Figura 8). Em cada carta os alunos devem dividir valor do PS por 100, resultando em valor entre 0 e 1 e ordenar as cartas segundo o bucket sort. De acordo com o resultado, colocar as cartas na posição respectiva do tabuleiro. Depois os alunos têm que concatenar as cartas nos baldes de forma ordenada.

Figura 8 – Simulação do vetor balde.

0	1	2	3	4
5	6	7	8	9

(Fonte: Própria, 2017)

#### 4. DESCRIÇÃO DAS OFICINAS

A primeira oficina de algoritmo de ordenação teve início em 03 de agosto de 2017 e contemplou o algoritmo de ordenação counting sort. Iniciou-se com uma apresentação da proposta da oficina. Neste dia, haviam 18 alunos na aula. Antes de iniciar a atividade desplugada os alunos foram convidados para responde o pré-teste (apêndice B), com duração de 20 minutos para responde as questões relacionadas com o algoritmo counting sort, o teste possuía 3 questões. A primeira questão ordenação de números, a segunda escrever o algoritmo e a terceira ordenação de um conjunto de cartas conforme é apresentado no apêndice B.

Depois que os alunos terminaram o pré-teste, o pesquisador apresentou o conceito do algoritmo aos alunos através de exemplos do cotidiano, depois a turma foi dividida em quatro grupos. Logo após, cada grupo recebeu o material para realizar a atividade proposta e, foram orientados como seria a atividade.

No início, os alunos tiveram um pouco de dificuldade em associar o estágio de evolução das cartas com números, ou seja, as atividades provavelmente apresentaram novidades para os alunos. Por isso, foi necessário explicar novamente para cada grupo. Percebeu-se que os alunos não estavam acostumados com ordenação de categorias de cartas.

Observou-se que cada grupo utilizou diferentes estratégias na resolução da atividade (apêndice D). Com esta atividade esperava-se que os alunos praticassem o funcionamento de ordenação do algoritmo. Quando terminou a atividade os alunos responderam o pós-teste (apêndice B).

A segunda oficina teve início em 17 de agosto de 2017 e contemplou os algoritmos de ordenação radix sort e bucket sort. Seguiu o mesmo cronograma da primeira oficina. Iniciou-se com uma apresentação da proposta da oficina. Neste dia, haviam 13 alunos na aula. Aplicou o pré-teste (apêndice C), com duração de 15 minutos para responde as questões relacionadas com os algoritmos radix sort e bucket sort. O teste possuía 4 questões. A primeira questão ordenação de um conjunto de cartas, a segunda escrever o algoritmo, a terceira e a quarta são semelhantes tal qual é apresentado no apêndice C.

Depois desse momento, foi explicado o conceito dos algoritmos aos alunos e foram orientados como seriam as atividades. Depois a turma foi dividida em três grupos. Logo após, cada grupo receberam os materiais das atividades propostas.

---

Todos os grupos realizaram bem as atividades. Quando terminou as atividades os alunos responderam o pós-teste (apêndice C).

## 5. TESTE ESTATÍSTICO PAREADO

A análise estatística dos dados envolveu as notas do pré-teste e pós-teste, no qual foram organizados para totalizar 10 pontos. Para analisar os dados foi utilizado o teste pareado de Wilcoxon da linguagem de programação R (R Core Team, 2017).

As subseções seguintes apresentam os resultados do primeiro experimento com o algoritmo counting sort e o segundo experimento com os algoritmos radix sort e bucket sort.

### 5.1 Teste estatístico pareado experimento com o counting sort

A Tabela 1 é uma apresentação dos resultados obtidos no pré-teste e pós-teste relativo o algoritmo counting sort. Nesta tabela estão listados a nota mínima, as notas que demarcam o primeiro quartil, a mediana, a média, terceiro quartil e a nota máxima obtidas pelos alunos.

Tabela 1 – Sumário das notas obtidas nos testes de algoritmo de ordenação counting sort

	Min.	1° Quartil	Mediana	Média	3° Quartil	Max.
Sumário Pré-Teste	0.00	2.00	3.00	3.088	4.00	7.00
Sumário Pós-Teste	1.50	3.00	5.00	5.118	7.00	10.00

Em síntese, a Tabela 1 as seguintes análises foram feitas no pré-teste e pós-teste. Analisando cada resposta obtida no pré-teste, as questões que tiveram maior declínio de erros, foram as questões: escrever o algoritmo e ordenar um conjunto de cartas. Notou-se que a maioria dos alunos apresentaram dificuldades em entender como associar números a categoria para ordenar as cartas solicitadas na questão, onde são expostas várias categorias de cartas simulando números desordenados em um arranjo. Em relação as questões que solicitava escrever o algoritmo, a maioria dos alunos deixaram as questões em branco ou escreveram que não se lembrava. As questões que apresentaram mais acertos, foram as questões de ordenação de números. Entretanto teve alunos que erraram ou deixaram em branco.

Analisando cada resposta obtida no pós-teste, observou-se uma evolução na quantidade de acertos, viu-se que a maioria dos alunos obteve sucesso no pós-teste, com notas bem altas, como 10. As questões com mais acertos foram: ordenação de um conjunto de cartas e ordenação de números.

Ao comparar as notas da turma, no pré-teste e pós-teste, as notas do pós-teste houve um aumento significativo em relação as notas do pré-teste.

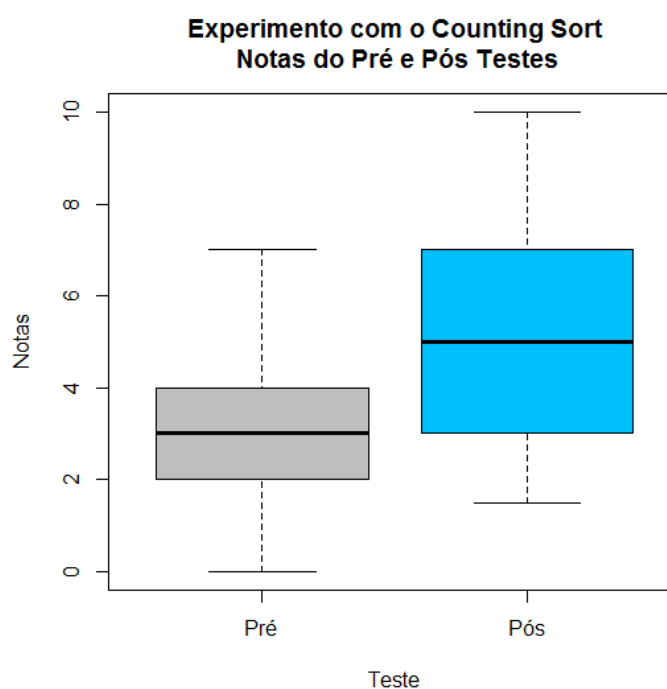
O teste de Shapiro-Wilk foi utilizado para analisar a normalidade das notas obtidas no pré-teste e pós-teste. O resultado do teste apresenta um p-valor = 0,01625, abaixo de 5% (0,05) de significância, sugerindo que os dados não podem ser associados a uma distribuição normal. Apontado para o teste pareado de Wilcoxon como a melhor ferramenta para analisar dados nessas circunstâncias.

Utiliza-se o teste de Wilcoxon com p-valor de 0,025, pois utiliza-se o teste duas vezes. A primeira vez para responder se há indícios de diferenças significativas entre os resultados do pré-teste e pós-teste. A segunda busca indícios de qual conjunto de notas é superior, as do pré-teste ou as do pós-testes. A segunda pergunta somente é realizada se o resultado da primeira pergunta indicar diferenças.

Aplicando o teste pareado de Wilcoxon para analisar se as notas do pré-teste e pós-teste são diferentes, o teste apresenta um p-valor = 0,001027, indicando que as diferentes notas são significativas. Analisando se as notas do pré-teste são inferiores à do pós-teste, temos evidências de que as notas do pré-teste são inferiores as notas do pós-teste pois, o teste apresenta um p-valor = 0,0005135.

No gráfico da Figura 9 temos visualmente os indícios que respaldam as diferenças expressas nos testes. Os box plots exibem os resultados do pré-teste e pós-teste. Analisando o box plot das notas obtida no pós-teste, em azul, e no pré-teste, em cinza, é possível observar que as 25% melhores notas do pós-teste estão acima das melhores notas do pré-teste. Observa-se que a mediana das notas do pré-teste coincide com o primeiro quartil do pós-teste. Além disso, o terceiro quartil das notas do pré-teste estão abaixo da mediana das notas do pós-teste. Esta discrepância entre as notas do pré-teste e pós-teste, são evidência que as atividades desplugadas impactaram na aprendizagem dos alunos.

Figura 9 – Comparação entre as notas do pré-teste e pós-teste do experimento com o algoritmo de ordenação counting sort.



---

## 5.2 Teste estatístico pareado experimento com o radix sort e bucket sort

A Tabela 2 é uma apresentação dos resultados obtidos no pré-teste e pós-teste relativos aos algoritmos radix sort e bucket sort. Nesta tabela estão listados a nota mínima, as notas que demarcam o primeiro quartil, a mediana, a média, terceiro quartil e a nota máxima obtidas pelos alunos.

Tabela 2 – Sumário das notas obtidas nos testes dos algoritmos de ordenação radix sort e bucket sort

	Min.	1° Quartil	Mediana	Média	3° Quartil	Max.
Sumário Pré-Teste	1.50	2.00	3.00	3.625	4.375	7.00
Sumário Pós-Teste	4.50	5.878	8.25	7.583	9.00	10.00

Em síntese, a Tabela 2 as seguintes análises foram feitas no pré-teste e pós-teste. Analisando cada resposta apresentada do pré-teste, as questões que tiveram maior declínio de erros, foram as questões: escrever o algoritmo e ordenação de um conjunto de cartas relacionados aos algoritmos radix sort e bucket sort.

Analisando cada resposta obtida no pós-teste, foi possível perceber um resultado satisfatórios nas respostas. Observou-se uma evolução na quantidade de acertos no pós-teste, apresentando notas bem altas, como 10. As questões que solicitava ordenar um conjunto de cartas por categoria, a maioria dos alunos acertaram. As questões que solicitava escrever o algoritmo, os alunos apresentaram diferentes respostas como por exemplo escreveram o pseudocódigo e explicaram através de passo a passo da sequência do algoritmo. Contudo, observou-se que depois das atividades desplugadas realizadas durante a oficina, os alunos tiveram um desempenho melhor em relação ao pós-teste. Junto a isso, poucos foram os alunos que errou ou deixaram alguma questão em branco.

Ao comparar as notas da turma, no pré-teste e pós-teste, as notas do pós-teste está bem acima das notas do pré-teste.

O teste de Shapiro-Wilk foi utilizado para analisar a normalidade das notas obtidas no pré-teste e pós-teste. O resultado do teste apresenta um p-valor = 0,04174, abaixo de 5% (0,05) de significância, sugerindo que os dados não podem ser associados a uma distribuição normal. Apontado para o teste pareado de Wilcoxon como a melhor ferramenta para analisar dados nessas circunstâncias.

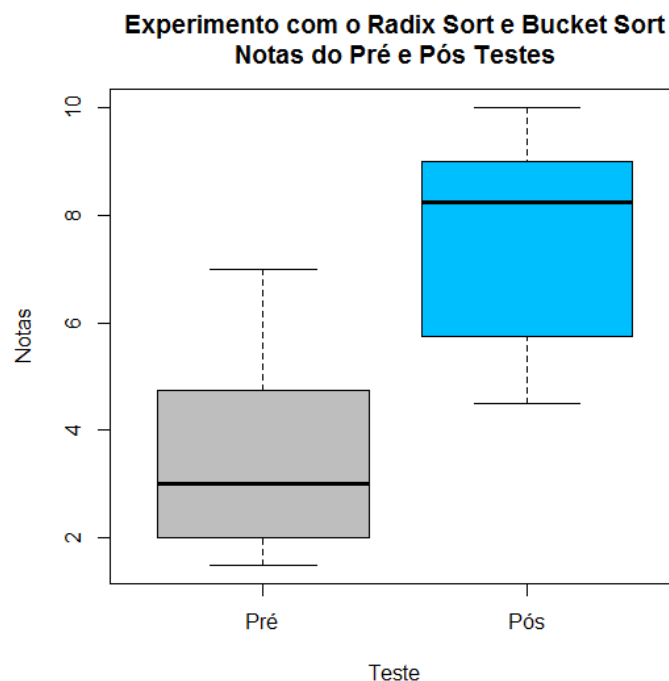
Utiliza-se o teste de Wicoxon com p-valor de 0,025, pois utiliza-se o teste duas vezes. A primeira vez para responder se há indícios de diferenças significativas entre os resultados do pré-teste e pós-teste. A segunda busca indícios de qual conjunto de notas é superior, as do pré-teste ou as do pós-testes. A segunda pergunta somente é realizada se o resultado da primeira pergunta indicar diferenças.

Aplicando o teste pareado de Wilcoxon para analisar se as notas do pré-teste e pós-teste são diferentes, o teste apresenta um p-valor = 0,002468, indicando que

as notas são diferentes. Analisando se as notas do pré-teste são inferiores à do pós-teste, temos evidências de que as notas do pré-teste são inferiores as notas do pós-teste pois, o teste apresenta p-valor = 0,001234.

No gráfico da Figura 10 temos visualmente os indícios que respaldam as diferenças expressas nos testes. Os box plots exibem os resultados do pré-teste e pós-teste. Analisando o box plot das notas obtida no pós-teste, em azul, e no pré-teste, em cinza, é possível observar que a menor nota do pós-teste é melhor que a mediana das notas do pré-teste, observa-se que o primeiro quartil das notas do pós-teste está acima do terceiro quartil das notas do pré-teste e a mediana das notas do pós-teste está bem acima da melhor nota do pré-teste. Esta discrepância entre as notas do pré-teste e pós-teste, são evidência que as atividades desplugadas influenciaram positivamente na aprendizagem dos alunos.

Figura 10 – Comparação entre as notas do pré-teste e pós-teste do experimento com os algoritmos de ordenação radix sort e bucket sort.



(Fonte: Própria, 2017)

## 6. CONSIDERAÇÕES FINAIS

Este artigo teve como objetivo relatar o planejamento e a aplicação de atividades de computação desplugada no ensino superior para apoiar o ensino de algoritmos de ordenação linear da disciplina de Estrutura de Dados II.

Percebeu-se que grande parte dos alunos estiveram bastante participativos durante as atividades realizadas nas oficinas. A atividade do counting sort, foi abordado como ordenar um conjunto de cartas por meio de categorias no qual cada categoria é associada a um número. A atividade do radix sort, consistia em ordenar as cartas sob dois critérios: o tipo de carta e o estágio de evolução das cartas obedecendo a estabilidade do algoritmo de ordenação. Por fim, a última atividade do bucket sort, utilizou os valores PS das cartas para ordenar com o

---

auxílio de um tabuleiro. Nessas atividades, os alunos tiveram a oportunidade de praticar os conteúdos estudados anteriormente em sala de aula.

Foi feita uma investigação usando a metodologia quantitativa de testes pareados para verificar a aprendizagem dos alunos, aplicando teste antes e depois de cada atividade desplugada. Os resultados da investigação quantitativa indicam que o uso das atividades propostas no ensino dos algoritmos de ordenação linear, influenciou significativamente na aprendizagem dos alunos.

Como trabalho futuros pretende-se aplicar as atividades em turma do ensino médio. Além disso, pretende-se continuar desenvolvendo novas atividades desplugadas para ensinar os algoritmos de tabela hash.



---

**APÊNDICE A: PLANEJAMENTO DE OFICINA - ALGORITMO DE ORDENAÇÃO**

UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE CIÊNCIAS APLICADAS E EDUCAÇÃO  
DEPARTAMENTO DE CIÊNCIAS EXATAS

**Nova proposta de atividades desplugadas no ensino de algoritmos de ordenação linear na educação superior**

**Planejamento de Oficina - Algoritmos de Ordenação**

<b>OFICINA 1</b>		
<b>Conteúdo:</b> Counting Sort		
<b>Carga Horária:</b> 1:40h	<b>Horário:</b> 8:00h às 9:40h	<b>Data:</b> 03/08/2017

<b>ROTEIRO DA OFICINA</b>	
<b>Passo a Passo:</b>	<b>Duração:</b>
Apresentação da proposta da oficina	5 minutos
Aplicar pré-teste	20 minutos
Explicação do algoritmo Counting Sort e da atividade desplugada	10 minutos
Atividade - Algoritmo de ordenação Counting Sort	45 minutos
Aplicar pós-teste	20 minutos
<b>TOTAL</b>	<b>1 hora 40 minutos</b>

---

## OFICINA 2

**Conteúdo:** Radix Sort e Bucket Sort

**Carga Horária:** 1:40h

**Horário:** 8:00h às 9:40h

**Data:** 17/08/2017

### ROTEIRO DA OFICINA

<b>Passo a Passo:</b>	<b>Duração:</b>
Apresentação da proposta da oficina	5 minutos
Aplicar pré-teste	15 minutos
Explicação do algoritmo Radix Sort e da atividade desplugada	10 minutos
Atividade 1 - Algoritmo de ordenação Radix Sort	20 minutos
Explicação do algoritmo Bucket Sort e da atividade desplugada	10 minutos
Atividade 2 - Algoritmo de ordenação Bucket Sort	20 minutos
Aplicar pós-teste	20 minutos
<b>TOTAL</b>	<b>1 hora 40 minutos</b>

## APÊNDICE B: PRÉ-TESTE / PÓS-TESTE - COUNTING SORT

### TESTE – COUNTING SORT

Aluno: .....

1. Considerando o algoritmo Counting Sort e um vetor A de inteiros.

$A = \{4, 5, 1, 2, 5, 6, 3, 2, 6, 3, 1, 1\}$

- a) Informe quantas vezes cada número acontece;
  - b) Informe quantos números menores ou iguais a cada valor acontece;
  - c) Descreva como codificar essa etapa;
  - d) Escreva o vetor ordenado segundo o Counting Sort.
2. Escreva o algoritmo Counting Sort.
  3. Considerando as 10 cartas de Pokémon embaralhadas, categoria do Pokémon Básico, Estágio 1, Estágio 2 e Energia.





- Desenhar os contadores de cada categoria inicialmente com zero;
- Desenhar os contadores de quantas cartas de cada categoria;
- Somar o contador ao da categoria anterior;
- Desenhar o vetor ordenado segundo o Counting Sort.




APÊNDICE C: PRÉ-TESTE / PÓS-TESTE – RADIX SORT E BUCKET SORT

TESTE – RADIX SORT E BUCKET SORT

Aluno: .....

1. Considere o algoritmo Radix Sort e as 8 cartas de Pokémon embaralhadas.



- Desenhar a ordenação das cartas por Tipo do Pokémon  seguindo a ordem apresentada na pergunta;
- Desenhar a ordenação das cartas por Estágio de Evolução Básico, Estágio 1 e Estágio 2.

2. Descreva o algoritmo Radix Sort.



3. Considere o algoritmo Bucket Sort e as 10 cartas de Pokémon embaralhadas.



- Desenhar os baldes e colocar as cartas nos baldes respectivos;
- Desenhar os baldes ordenados.

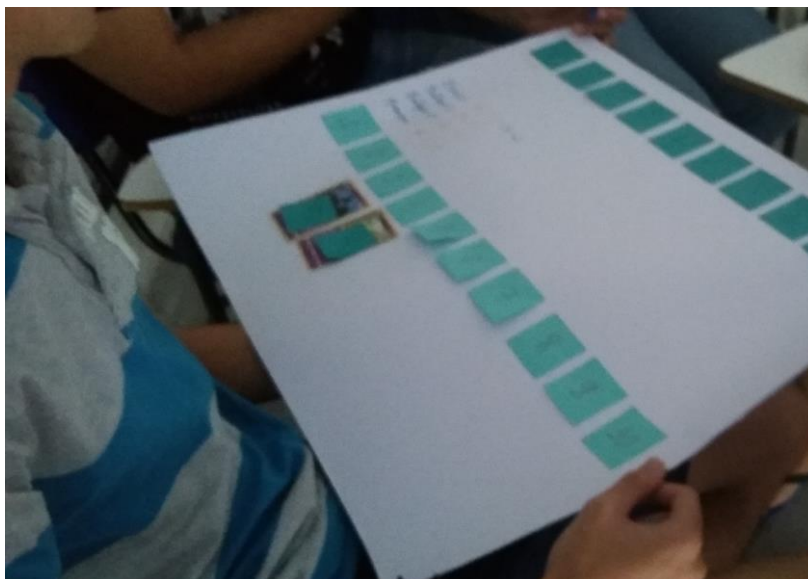
4. Descreva o algoritmo Bucket Sort.

---

## APÊNDICE D: ARTEFATOS DAS ATIVIDADES DESPLUGADAS

A estratégia utilizada pelo grupo A na figura D1 foram as seguintes: os alunos utilizaram as folhas de post-it para simular dois arranjos com 10 posições cada, o primeiro, a entrada contém as cartas desordenadas e o segundo contém a saída ordenada das cartas. Os alunos escreveram na cartolina o passo do algoritmo com diferentes cores para diferenciar os elementos de cada instrução do algoritmo, facilitando assim a ordenação das categorias das cartas.

Figura D1 – Atividade desplugada counting sort - grupo A.

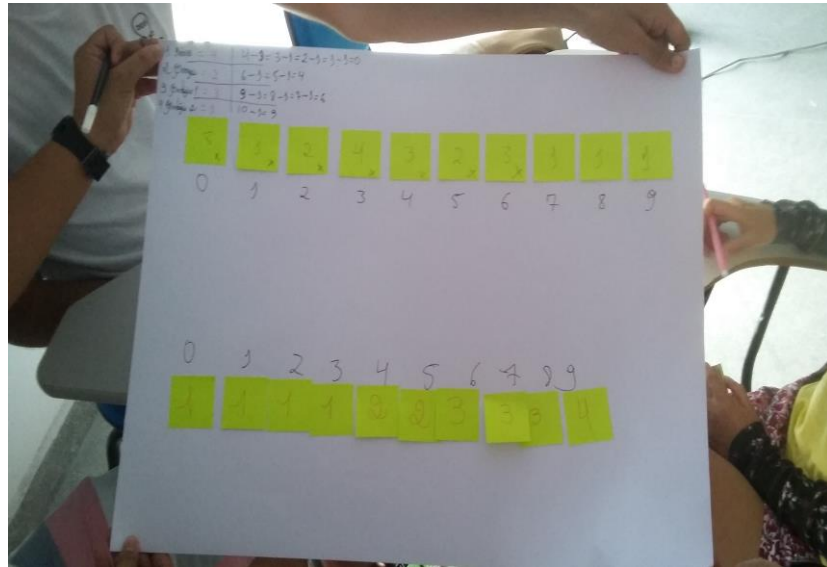


(Fonte: Própria, 2017)

A estratégia utilizada pelo grupo B na figura D2 foi bastante diferente, os alunos utilizaram apenas números ao invés de utilizar as categorias das cartas. A estratégia foram como segue: primeiramente os alunos relacionaram números para cada categoria. Após essa etapa, os alunos utilizaram as folhas de post-it para simular 2 arranjos com 10 posições cada. O primeiro arranjo contém números desordenados e o segundo arranjo contém a saída com os números ordenados. Para ordenar os números do primeiro arranjo, os alunos escreveram o passo do algoritmo para auxiliar na ordenação.



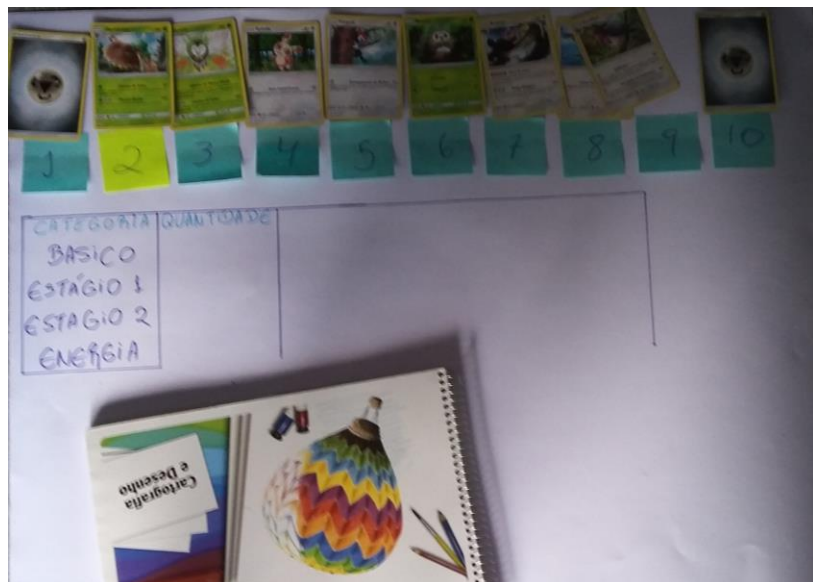
Figura D2 – Atividade desplugada counting sort - grupo B.



(Fonte: Própria, 2017)

A estratégia utilizada pelo grupo C na figura D3 foram as seguintes: os alunos utilizaram as folhas de post-it para simular dois arranjos com 10 posições cada, o primeiro, a entrada contém as cartas desordenadas e o segundo contém a saída ordenada das cartas. Além disso, utilizaram uma tabela bem detalhada como referência para ordenar o primeiro arranjo. Essa tabela foi uma maneira encontrada pelo grupo no qual facilitou a ordenação das cartas com facilidade.

Figura D3 – Atividade desplugada counting sort - grupo C.



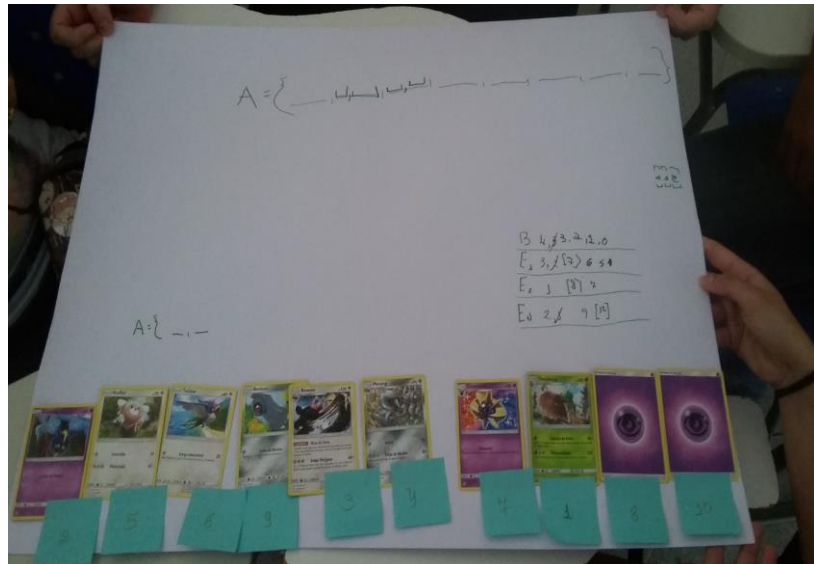
(Fonte: Própria, 2017)

Por fim, a estratégia utilizada pelo grupo D na figura D4 foram as seguintes: o conceito do pseudocódigo do algoritmo counting sort foi utilizado para ordenar as cartas. Os alunos desenharam na cartolina a simulação de um arranjo A com 10 posições. Para ordenar o arranjo A, utilizaram contadores conforme o



pseudocódigo, através dos índices de cada posição do arranjo A, obedecendo o comportamento do método de ordenação do counting sort.

Figura D4 – Atividade desplugada counting sort - grupo D.



(Fonte: Própria, 2017)

## APÊNDICE E: CARTA PERSONALIZÁVEL DO JOGO OVERWATCH

Overwatch é um jogo eletrônico multiplayer de tiro que se passa no futuro do planeta Terra. O jogo enfatiza a jogabilidade cooperativa usando um elenco de vários "heróis", que possuem suas próprias habilidade e poderes únicos que se une numa força-tarefa internacional, para restaurar a paz num mundo devastado pela guerra.

Diante deste contexto, um conjunto de cartas foi elaborado inspirados nos personagens do jogo de Overwatch disponível em ([http://overwatch.wikia.com/wiki/Overwatch\\_Wiki](http://overwatch.wikia.com/wiki/Overwatch_Wiki)) do Wikia Overwatch. Estas cartas descrevem várias características dos personagens que permite vários critérios de ordenação. A seguir observaremos soluções e dicas de cartas personalizadas feitas no aplicativo PowerPoint.

**Descrição:** Bastian é um robô que se transforma e explora o mundo; fascinado pela natureza, porém cauteloso com a humanidade.



Bastian

### Bastian

Tipo: Defensivo 

Dificuldade: 

Vida:	Armaduras:	Escudo:	Total:
200	150	0	350

Ataque: submetralhadora



20 de munição  
8 disparos por segundo  
20 de dano

**Mercy:** Diga-me seus pensamentos, meu amigo.

**Bastian:** [beeping amigável]

**Descrição:** D.Va é uma ex-gamer profissional transformada em piloto de mecha de última geração na defesa de sua cidade natal.



D.Va

**D.Va**

Tipo: Tanque 

Dificuldade: 

Vida: 400   Armaduras: 200   Escudo: 0   Total: 600

Ataque: Canhões de Fusão



∞ de munição  
7 disparos por segundo  
3 de dano

**Soldado: 76:** a guerra não é um jogo.

**D.Va:** Você está certo de que a vida não é um jogo, Soldado: 76?

**Descrição:** Soldado: 76 é um vigilante que vai lutar até o fim para entregar os inimigos da Overwatch nas mãos da justiça.



Soldado: 76

**Soldado: 76**

Tipo: Ofensivo 

Dificuldade: 

Vida: 200   Armaduras: 0   Escudo: 0   Total: 200

Ataque: Rifle de Pulso Pesado



25 de munição  
10 disparos por segundo  
17 de dano


**Soldado: 76:** Você sabe, eu ainda te devo por salvar minha vida uma vez.


**Mercy:** Uma vez? Você quer dizer no Egito? Ou na Rússia? Ou no Brasil?

**Descrição:** Mercy é um anjo da guarda, uma curandeira sem igual, cientista brilhante e defensora convicta da paz.




Mercy

Tipo: Suporte 

Dificuldade: 

Vida: 200 Armaduras: 0 Escudo: 0 Total: 200

Ataque: Bastão de Caduceu



30 dano  
60 disparos por segundo

**Mercy:** Então, o que estamos fazendo aqui?

**Soldado: 76:** eu preciso fazer uma visita a um amigo.

**Descrição:** Genji é um ciborgue ninja e guerreiro mortal que fez as pazes com seu corpo mecânico.



Genji

Tipo: Ofensivo 

Dificuldade: 

Vida: 200 Armaduras: 0 Escudo: 0 Total: 200

Ataque: Dragonblade



120 dano por ação alcance de 5m  
8 m / s velocidade de movimento  
Dura 6 segundos

**D. Va:** Você pode ser rápido Genji, mas você não é mais rápido do que uma bala.

**Genji:** por que não descobrimos?

Além disso, outras coisas podem ser feitas. Como exemplo: animais, carros e outros. A seguir temos um exemplo de cartas com animais.

Leão



**Tamanho:** 1,5 metro de comprimento

**Grupo:** Mamíferos

**Tipo de alimentação:** Carnívoros

Tucano



**Tamanho:** 56 centímetros de comprimento

**Grupo:** Aves

**Tipo de alimentação:** Onívoros

Peixe



**Tamanho:** 30 centímetros de comprimento

**Grupo:** Peixes

**Tipo de alimentação:** Herbívoros

---

# New proposal of unplugged activities in the teaching of linear ordering algorithms in higher education

## ABSTRACT

The objective of this work is to elaborate and apply unplugged activities in higher education with students of the discipline of Data Structure II of the Bachelor degree in Information Systems of the Federal University of Paraíba - UFPB Campus IV. In this work two workshops have been described to support the teaching of linear ordering algorithms. The methodological approach was quantitative research. The effectiveness of the proposal was measured through written tests and statistical testing under the results. The results indicate that the impact of the methodology is significant.

**KEYWORDS:** Unplugged activity novel. Algorithm teaching. Data structure.

---

## REFERÊNCIAS

ANDRIANI, Fabricio Caixeta; SENA, Claudia Pinto Pereira; CARDOSO, Naan Silva. FLUXOGRAMA HUMANO: DINÂMICA PARA O ENSINO DE ALGORITMOS BASEADA NA COMPUTAÇÃO DESPLUGADA PARA CURSOS DE ENGENHARIA E TI.

BELL, T., WITTEN, I, H. e FELLOWS, M. (2011). **Computer Science Unplugged: Ensinando Ciência da Computação sem o uso do computador**. Tradução coordenada por Luciano Porto Barreto.

CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., STEIN, C. (2002). Algoritmos: teoria e prática. 2nd Edição. Rio de Janeiro. Editora Campus.

da COSTA, T. L. S., SOUZA, F. V. C., COSTA, W. E. (2017). **O uso da computação desplugada para apoiar a aprendizagem de algoritmos de ordenação e tabela hash**. Trabalho de conclusão de curso. Departamento de Ciências Exatas. Universidade Federal da Paraíba.

HAMMING, R. W. (2012) Methods of Mathematics Applied to Calculus, Probability, and Statistics. 2nd Edition.

MORETTIN, Pedro Alberto. Estatística básica, 8ª edição, 8th edição. Saraiva, 06/2009.

O box plot. Disponível em:  
<<http://www.uff.br/cdme/conheceboxplot/conheceboxplot-html/boxplot.pdf>>.  
Acesso em: 28 nov. 2017.

R CORE TEAM (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

VIEIRA, A. and ODETTE PASSOS, R. B. (2013). Um relato de experiência do uso da técnica computação desplugada. In: **XXXIII Congresso da Sociedade Brasileira da Computação**, p. 671–680.