

ATIVIDADES PRÁTICAS SUPERVISIONADAS

Sistemas de Informação

6ª Série

Classificação e Pesquisa

A atividade prática supervisionada (ATPS) é um procedimento metodológico de ensino-aprendizagem desenvolvido por meio de um conjunto de etapas programadas e supervisionadas e que tem por objetivos:

- ✓ Favorecer a aprendizagem.
- ✓ Estimular a corresponsabilidade do aluno pelo aprendizado eficiente e eficaz.
- ✓ Promover o estudo, a convivência e o trabalho em grupo.
- ✓ Desenvolver os estudos independentes, sistemáticos e o autoaprendizado.
- ✓ Oferecer diferentes ambientes de aprendizagem.
- ✓ Auxiliar no desenvolvimento das competências requeridas pelas Diretrizes Curriculares Nacionais dos Cursos de Graduação.
- ✓ Promover a aplicação da teoria e conceitos para a solução de problemas práticos relativos à profissão.
- ✓ Direcionar o estudante para a busca do raciocínio crítico e a emancipação intelectual.

Para atingir estes objetivos a ATPS propõe um desafio e indica os passos a serem percorridos ao longo do semestre para a sua solução.

A sua participação nesta proposta é essencial para que adquira as competências e habilidades requeridas na sua atuação profissional.

Aproveite esta oportunidade de estudar e aprender com desafios da vida profissional.



AUTORIA:

Jaqueline Brigidori Pugliesi
Faculdade Anhanguera de Valinhos

Jeanne Dobgenski
Anhanguera Educacional Ltda

Marcelo Augusto Cicogna
Faculdade Anhanguera de Valinhos

COMPETÊNCIAS E HABILIDADES

Ao concluir as etapas propostas neste desafio, você terá desenvolvido as competências e habilidades que constam, nas Diretrizes Curriculares Nacionais, descritas a seguir.

- ✓ Utilizar princípios e ferramentas que otimizem o processo de desenvolvimento e implementação de um projeto.
- ✓ Aplicar de forma eficiente os princípios de gerenciamento, organização e busca de informações.
- ✓ Analisar, organizar, abstrair e relacionar dados e informações.
- ✓ Trabalhar em equipe.

Produção Acadêmica

Relatórios parciais, com os resultados das pesquisas e atividades realizadas.

Participação

Esta atividade será, em parte, desenvolvida individualmente pelo aluno e, em parte, pelo grupo. Para tanto, os alunos deverão:

- Organizar-se, previamente, em equipes de participantes conforme orientação do professor.
- Entregar seus nomes, RAs e *e-mails* ao professor da disciplina.
- Observar, no decorrer das etapas, as indicações: Aluno e Equipe.

Padronização

O material escrito solicitado nesta atividade deve ser produzido de acordo com as normas da ABNT¹, com o seguinte padrão:

- Em papel branco, formato A4.
- Com margens esquerda e superior de 3cm, direita e inferior de 2cm.
- Fonte *Times New Roman* tamanho 12, cor preta.
- Espaçamento de 1,5 entre linhas.
- Se houver citações com mais de três linhas, devem ser em fonte tamanho 10, com um recuo de 4cm da margem esquerda e espaçamento simples entre linhas.
- Com capa, contendo:
 - Nome de sua Unidade de Ensino, Curso e Disciplina.
 - Nome e RA de cada participante.
 - Título da atividade.
 - Nome do professor da disciplina.
 - Cidade e data da entrega, apresentação ou publicação.

¹ Consulte o Manual para Elaboração de Trabalhos Acadêmicos. Unianhanguera. Disponível em: http://www.unianhanguera.edu.br/anhanguera/bibliotecas/normas_bibliograficas/index.html.

DESAFIO

Vivemos num mundo obcecado por possuir e manter informação e, tendo em vista a necessidade de armazenar esses dados, é fundamental guardá-los de uma forma organizada para facilitar sua recuperação.

Em sistemas computacionais essa é uma tarefa que tem sido muito explorada e aprimorada, pois os algoritmos de ordenação e busca são fundamentais para a maioria dos sistemas de informação. Pode-se perguntar por que tanto interesse em desenvolver algoritmos melhores e mais eficientes se já há uma grande variedade disponível. Não seria mais fácil escolher um deles e usar na aplicação sendo desenvolvida?

É na resposta dessa pergunta que este desafio se baseia. Por exemplo considere o exemplo de uma empresa de *outsourcing* – que presta serviços de tecnologia da informação (TI) a terceiros, havia padronizado os algoritmos de ordenação e busca que eram aplicados para atender as demandas de seus clientes. Essa padronização facilitava e agilizava o trabalho dos desenvolvedores, pois sempre que essa necessidade ocorria usavam os algoritmos determinados previamente. Entretanto, começaram a receber tantas reclamações de seus clientes sobre o desempenho dos sistemas que o gerente de TI, responsável pelo setor, convocou uma equipe de analistas para estudarem essa ocorrência indicando a(s) falha(s) e as ações necessárias para resolver o problema.

O gerente de TI solicitou à equipe (que será identificada nesse desafio por *Leader team*) relatórios parciais que indicassem o progresso da análise, nos quais deveriam constar exatamente os passos realizados em cada teste e, sempre que possível, que resumissem o estudo em gráficos e dados estatísticos.

Para resolver esse desafio considere que você e seus colegas (até 4 indivíduos) integram a equipe convocada para estudar e solucionar o problema descrito. Para isso, entreguem ao professor da disciplina seus nomes e RAs, lembrando que deverão permanecer juntos na execução de todas as tarefas. Em caso de necessidade de alteração o professor deverá ser consultado.

É importante lembrar que o objetivo do desafio é estudar o comportamento de algoritmos clássicos de ordenação e busca, não sendo o foco a implementação deles. Por isso, a equipe poderá utilizar códigos disponíveis em livros da área ou disponibilizados pelo professor da disciplina. Em qualquer caso é importante citar, no relatório de cada etapa, a fonte dos códigos utilizados.

Leiam atentamente as etapas a seguir e boa sorte!

Objetivo do Desafio

Elaborar um estudo mostrando o progresso da análise, teste e gráficos e dados estatísticos sobre os algoritmos de ordenação e busca de uma empresa.

ETAPA 1 (tempo para realização: 5 horas)

- ✓ **Aula tema: Pesquisa de Dados: Sequencial. Pesquisa de Dados: Binária. Métodos de Ordenação: Seleção, Troca.**

Esta atividade é importante para que você verifique a importância da especificação adequada de algoritmos que recuperam informações em base de dados, tendo em vista a eficiência e o esforço computacional.

Para realizá-la é importante seguir os passos descritos.

PASSOS

Passo 1 (Equipe)

Considerar que a primeira decisão tomada por *Leader team* foi estabelecer uma bateria de testes com os algoritmos de busca e ordenação definidos como padrão para mostrar o comportamento em situações distintas. Portanto, testarão os seguintes algoritmos.

1. Busca linear.
2. Busca linear com sentinela.
3. Busca Binária.
4. Ordenação usando seleção.
5. Ordenação usando *bubblesort*.

Como cada integrante da equipe ficou responsável por realizar um teste, foi estabelecido o uso de uma função geradora de números aleatórios para que todos os testes sejam executados com a mesma base de dados. Desta forma, usarão os algoritmos 1 e 2, descritos a seguir, que geram números reais e inteiros, respectivamente, dentro de uma faixa determinada e com distribuição uniforme.

```

/*****
Algoritmo 1 - Gerador de números reais aleatórios

Gerador de distribuicao uniforme retorna um numero
double (real com longa precisão) na faixa low - high,
ou seja, [low,high].
*****/

double unif(long int *seed, double low, double high)
{
    double    unif_ret;
    long int  m,a,b,c, k;
    double    value_0_1;

    m = 2147483647;
    a = 16807;
    b = 127773;
    c = 2836;

    k = *seed/b;
    *seed = a * (*seed % b) - k*c;
    if (*seed < 0)
        *seed = *seed + m;
    value_0_1 = (double) *seed/m;
    unif_ret = low+value_0_1*(high - low);
    return (unif_ret);
}
/*****
Algoritmo 2 - Gerador de números inteiros aleatórios

Gerador de distribuicao uniforme retorna um numero inteiro
na faixa low - high, ou seja, [low,high].
*****/

int inteiros_unif(seed, low, high)
long int *seed; int low; int high;
{

```

```

int    unif_ret;
long int m,a,b,c, k;
double value_0_1;

m = 2147483647;
a = 16807;
b = 127773;
c = 2836;

k = *seed/b;
*seed = a * (*seed % b) - k * c;
if (*seed < 0)
    *seed = *seed + m;
value_0_1 = (double) *seed/m;
unif_ret = low+value_0_1*(high-low+1);
return (unif_ret);
}

```

Na Tabela 1, apresentam-se os dados que a equipe testará para realizar a análise dos algoritmos citados, sendo *N* o número de números gerados, *low* a faixa inferior de número a ser gerado, *high* é a faixa superior, *seed* é a semente a ser usada nos testes, e, por fim, os números a serem procurados na busca.

Tabela 1: Parâmetros para a realização dos testes computacionais.

<i>N</i>	<i>low</i>	<i>high</i>	<i>seed</i>	Número procurado	Segundo número procurado
100	0	100000	1234554321	87	100001
1000	0	100000	1234554321	87	100001
10000	0	100000	1234554321	87	100001
100000	0	100000	1234554321	87	100001

Os testes a serem realizados, devem, necessariamente, utilizar os parâmetros da Tabela 1 para as duas buscas lineares citadas. Além disso, é necessária a utilização das duas funções geradoras de números aleatórios, ou seja, *unif()* e *inteiros_unif()* para os mesmos parâmetros.

Os testes com os algoritmos de busca deverão apresentar para cada dimensão de *N*:

1. o tempo gasto na execução de cada algoritmo para encontrar ambos os valores indicados na Tabela 1;
2. o número de testes que cada algoritmo efetuou para buscar cada um dos números solicitados.

Passo 2 (Equipe)

Ler o texto e fazer as atividades apresentadas a seguir.

Para os mesmos conjuntos de dados usados nos testes do Passo 1 deverão ser aplicados os algoritmos de ordenação por seleção e *bubblesort*.

Os testes com os algoritmos de ordenação deverão apresentar para cada dimensão de *N*:

1. o tempo gasto na execução de cada algoritmo para os dados;
2. o número de testes que cada algoritmo efetuou para trocar os elementos e efetivar a ordenação.

Passo 3 (Equipe)

Ler o texto e fazer as atividades apresentadas a seguir.

O último teste a ser realizado nessa etapa é o uso da busca binária. Como esse método só funciona em bases de dados ordenadas, testem o algoritmo nas bases que foram dispostas em ordem no Passo 2. Para que as análises sejam adequadas, a equipe deverá refazer os testes de busca, agora com os três algoritmos indicados: linear, linear com sentinela e busca binária.

Os testes com os algoritmos de busca deverão apresentar para cada dimensão de N :

1. o tempo gasto em cada execução do algoritmo para encontrar ambos os valores indicados na Tabela 1;
2. o número de testes que cada algoritmo efetuou para buscar cada um dos números solicitados.

Passo 4 (Equipe)

Documentar essa etapa de estudos em um relatório com o nome de **Relatório 1 – Pesquisa de Dados** apresentando quais os resultados alcançados em cada Passo executado. Para isso, deverão apresentar:

1. qual o método de busca com melhor desempenho computacional – é importante lembrar que os testes relacionados a tempo serão dependentes da máquina utilizada;
2. indicar o desempenho da busca binária x busca linear x busca linear com sentinela: descrever se compensa ordenar uma base de dados para então recuperar a informação com a busca binária ou se outra busca foi mais eficaz nos testes realizados;
3. qual foi a conclusão da equipe nessa bateria de testes.

ETAPA 2 (tempo para realização: 5 horas)

- ✓ **Aula tema: Métodos de Ordenação: Seleção, Troca. Métodos de Ordenação: Inserção. Métodos de Ordenação: Intercalação.**

Esta atividade é importante para que você analise e compreenda o comportamento de algoritmos de ordenação clássicos, identifique a aplicação adequada e perceba o quão fundamental é realizar testes computacionais para estudar o desempenho de métodos de ordenação.

Para realizá-la é importante seguir os passos descritos.

PASSOS

Passo 1 (Equipe)

Ler o texto e fazer as atividades apresentadas a seguir.

Uma vez que *Leader team* realizou os testes para averiguar a *performance* dos algoritmos tidos como padrão para a empresa, chegou o momento de buscar novas alternativas, indicando o comportamento de outros algoritmos de ordenação.

Para isso, decidiram testar algoritmos de ordenação clássicos para comprovar sua eficácia e a sua indicação de aplicação. Os algoritmos eleitos foram:

- A. ordenação por troca: *quicksort*;
- B. ordenação por inserção: binária;
- C. ordenação por inserção: *shellsort*;
- D. ordenação por seleção: *heapsort*;
- E. ordenação por intercalação: *mergesort*.

Nesse estudo os testes serão efetuados em dados gerados com a função geradora de números inteiros aleatórios (Algoritmo 2). Será necessário gerar três (03) instâncias de cada dimensão de dados, apresentando qual foi a semente e o intervalo de valor (*low*, *high*) usado para gerar de cada instância.

Observem que esses intervalos deverão ser distintos e não iguais – para cada instância. Logo, para testar cada algoritmo de ordenação, deverão ser usados os parâmetros apresentados na Tabela 2. O preenchimento desta tabela deverá constar no relatório parcial com os valores usados para a realização dos testes.

Tabela 2: Parâmetros para a realização dos testes computacionais.

Instância	<i>low</i>	<i>high</i>	<i>seed</i>
1			
2			
3			

Os testes serão empíricos e deverão apresentar o número de trocas e o número de comparações efetuadas pelo algoritmo durante o processo de ordenação, assim como o tempo gasto. As bases de dados para testes devem ter 500, 5.000 e 50.000 elementos. Esses elementos devem estar classificados em ordem aleatória, ascendente e descendente.

Nesse primeiro passo, as equipes deverão implementar os algoritmos, definir os valores para a Tabela 2 e apresentar na Tabela 3 os resultados dos testes com os dados aleatórios. Lembrar-se de informar quais foram os parâmetros usados para iniciar o *quicksort* e *shellsort*, os quais devem ser indicados no relatório.

Tabela 3: Resultados para dados em ordem Aleatória – média entre as 3 instâncias.

TIPO	Dados Aleatórios								
Instância	Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3		
Dados	500			5.000			50.000		
Algoritmo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo
A									
B									
C									
D									
E									

Passo 2 (Equipe)

Realizar os mesmos testes efetuados no Passo 1 para os dados classificados em ordem ascendente. Ou seja, vão analisar o comportamento dos algoritmos de ordenação quando aplicados em base de dados que já estão ordenadas. Para isso, deverão usar os mesmos dados do Passo 1 (Tabela 2) e apresentar na Tabela 4 os resultados dos testes com os dados Ascendentes.

Tabela 4: Resultados para dados em ordem Ascendente – média entre as 3 instâncias.

TIPO	Dados em ordem Ascendente								
Instância	Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3		
Dados	500			5.000			50.000		
Algoritmo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo
A									
B									
C									
D									
E									

Passo 3 (Equipe)

Realizar os mesmos testes já efetuados para os dados classificados em ordem decrescente. Ou seja, vão analisar o comportamento dos algoritmos de ordenação quando aplicados em base de dados em ordem decrescente. Para isso, deverão usar os mesmos dados do Passo 1 e 2 (Tabela 2) e apresentar na Tabela 5 os resultados dos testes com os dados Descendentes.

Tabela 5: Resultados para dados em ordem Descendente – média entre as 3 instâncias.

TIPO	Dados em ordem Descendente								
Instância	Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3		
Dados	500			5.000			50.000		
Algoritmo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo
A									
B									
C									
D									
E									

Passo 4 (Equipe)

Ler o texto e fazer as atividades apresentadas a seguir.

A *Leader team* deverá documentar essa etapa de estudos apresentando quais os resultados alcançados em cada Passo executado. Para isso deverão apresentar:

1. o desempenho de cada algoritmo de ordenação deverá ser analisado descrevendo as principais características reveladas pelos testes;
2. deverá ser apresentada a análise de todos os algoritmos em conjunto, sendo descrito qual a indicação de cada um deles, ou seja, uma análise comparativa. Com base nesta análise, indicar quais os três (03) algoritmos que considera os mais importantes explicando o porquê desta indicação;
3. escolher um entre os três algoritmos indicados para apresentar em forma gráfica o comportamento do processo de ordenação para um teste de dimensão 500, com a classificação aleatória, ascendente e decrescente. Para isso, deverão apresentar graficamente os dados antes da ordenação, em dois estágios intermediários e no final do processo. Dica: vejam a página do professor Jason Harrison. Disponível em: <<http://people.cs.ubc.ca/~harrison/Java/sorting-demo.html>>. Acesso em: 18 set. 2011.
4. Organizar todas as informações trabalhadas nesse passo em um relatório com o nome de **Relatório 2 – Métodos de Ordenação** e entregar o material ao professor da disciplina.

ETAPA 3 (tempo para realização: 5 horas)

- ✓ **Aula tema: Árvores de Pesquisa, Árvores Binárias de Pesquisa. Árvores AVL. B-Trees.**

Esta atividade é importante para que você se familiarize com árvores de busca e assimile seu comportamento e operação.

Para realizá-la é importante seguir os passos descritos.

PASSOS

Passo 1 (Equipe)

Ler o texto e fazer as atividades a seguir.

Uma estrutura bastante usada para guardar informações são as árvores binárias. Dessa forma, outro teste a ser realizado pela *Leader team* é organizar a informação em árvores binárias. Para isso usarão os mesmos dados definidos na Etapa 2 e apresentarão o esforço computacional para organizá-los em árvore binária na Tabela 6.

Tabela 6: Resultados para dados em ordem Aleatória – média entre as 3 instâncias.

TIPO	Dados Aleatórios								
Instância	Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3		
Dados	500			5.000			50,000		
Algoritmo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo
Árvore Binária									

Passo 2 (Equipe)

Ler o texto e fazer as atividades a seguir.

A *Leader team* definiu a necessidade de averiguar o comportamento de árvores AVL. Para isso usarão os mesmos dados do Passo 1 e apresentarão o esforço computacional para organizá-los em árvore AVL na Tabela 7.

Tabela 7: Resultados para dados em ordem Aleatória – média entre as 3 instâncias.

TIPO	Dados Aleatórios								
Instância	Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3		
Dados	500			5.000			50,000		
Algoritmo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo
Árvore AVL									

Passo 3 (Equipe)

Considerar que outra estrutura em árvores a ser explorada é a B-Tree. Para isso devem usar os mesmos dados do Passo 1 e apresentar o esforço computacional para organizá-los em árvore B-Tree na Tabela 8.

Tabela 8: Resultados para dados em ordem Aleatória – média entre as 3 instâncias.

TIPO	Dados Aleatórios								
Instância	Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3			Média entre instâncias 1, 2 e 3		
Dados	500			5.000			50.000		
Algoritmo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo	No. de comparações	No. de trocas	tempo
<i>B-Tree</i>									

Passo 4 (Equipe)

Documentar essa etapa de estudos em um relatório com o nome de **Relatório 3 – Árvores – Parte 1** apresentando quais os resultados alcançados em cada Passo executado. Deverão apresentar:

1. a análise sobre cada algoritmo utilizado;
2. apontar as diferenças entre os algoritmos;
3. indicar sua aplicação.

ETAPA 4 (tempo para realização: 5 horas)

- ✓ **Aula tema: Árvores de Pesquisa, Árvores Binárias de Pesquisa. Árvores AVL.**

Esta atividade é importante para que você desenvolva sua capacidade de análise e síntese.

Para realizá-la é importante seguir os passos descritos.

PASSOS

Passo 1 (Equipe)

Ler o texto e fazer as atividades a seguir.

Como foi visto no decorrer desse desafio, existem na literatura dezenas de algoritmos de ordenação direcionados para as mais diversas aplicações. Alguns algoritmos de ordenação são tão sofisticados ao ponto de levarem o desenvolvedor a gastar dias para estudá-los e testá-los. Um bom profissional dessa área, no entanto, sabe onde procurar e pesquisar a ferramenta necessária para ajudar no desenvolvimento de seu trabalho.

Por esse motivo a *Leader Team* deverá apresentar explicações sobre as análises realizadas neste estudo.

1. A primeira questão a ser considerada é por que houve reclamações de clientes com relação a desempenho.
2. A segunda é se existe um algoritmo de ordenação que pode ser usado como padrão para qualquer base de dados. Se sim, expliquem e digam qual, se não explicarem, por quê.

Passo 2 (Equipe)

Apresentar os principais aspectos que devem ser levados em consideração no momento da escolha do método a ser usado para ordenar uma base de dados e mantê-la ordenada. Para isso deverão considerar principalmente:

1. a dimensão da base de dados;
2. o número de execuções de ordenação realizadas pelo sistema;
3. o número de buscas realizadas pelo sistema.

Com base nesses dados, as equipes deverão elaborar o relatório final a ser entregue ao gerente de TI que solicitou o estudo.

Passo 3 (Equipe)

Finalizar o trabalho em um relatório com o nome de **Relatório 4 – Árvores – Parte 2**. Ele deverá apresentar:

1. a contextualização do Passo 1 e 2 dessa Etapa final;
2. a posição da equipe sobre as melhores práticas em situações que envolvam pequenas bases de dados, grandes bases de dados e bases de dados de tamanho médio;
3. as considerações finais da equipe sobre o estudo realizado.

Livro Texto da Disciplina

ZIVIANI, Nivio. *Projeto de Algoritmos: com implementação em Pascal e C*. 2ª ed. São Paulo: Pioneira Thomson Learning, 2007.