Anhanguera

ATIVIDADES PRÁTICAS SUPERVISIONADAS

TIAGO
ESTRUTURADA
2
ANTERIOR

Ciência da Computação

4ª. Série
Estrutura de Dados

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:
Thiago Salhab Alves
Faculdade Anhanguera de Santa Bárbara

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.

- ✓ Projetar, conduzir experimentos e interpretar resultados.
- ✓ Conceber, projetar e analisar sistemas, produtos e processos.
- ✓ Desenvolver e/ ou utilizar novas ferramentas e técnicas.
- ✓ Avaliar criticamente a operação e a manutenção de sistemas.

Produção Acadêmica

- Relatório 1 - Estrutura de Dados.
- Relatório 2 - Listas Ligadas.
- Relatório 3 - Filas e Pilhas.
- Relatório 4 - Grafos.

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 2 a 5 participantes;
- entregar seus nomes, RAs e *e-mails* ao professor da disciplina e
- 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 ABNT[1], 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

Estrutura de Dados é o nome dado a organização de dados e algoritmos de forma coerente e racional de modo a otimizar o seu uso. Por meio da maneira como os dados são organizados e como as operações são efetuadas sobre estes dados, se podem solucionar de forma simples problemas extremamente complexos. Existem diversos modelos de estruturas de dados, e novos modelos são criados constantemente, pois acompanham também a evolução dos algoritmos e das linguagens de programação. Escolher uma estrutura de dados ideal pode tornar-se um problema difícil para uma determinada solução. As pesquisas e estudos das estruturas de dados estão em constante desenvolvimento, apesar disso, existem estruturas que têm se mostrado padrão, ou seja, são clássicas. Podemos citar as listas, pilhas, filas, árvores e grafos como estruturas de dados clássicas.

Neste desafio, deverá ser apresentada uma solução computacional baseada em Estruturas de Dados para uma empresa aérea chamada VOEBEM que deseja automatizar alguns processos adotados pela companhia. A empresa necessita de um sistema que permita controlar a lista de voos diários e de passageiros por vôos, realize a reserva de assentos e liberação para taxiamento das aeronaves, realize o controle de armazenamento das bagagens no compartimento de carga e realize levantamento das rotas de voos.

Objetivo do Desafio

Utilizar as estruturas de dados listas, pilhas, filas e grafos para resolver problemas computacionais de uma empresa aérea através da construção de algoritmos para controle de voos e passageiros, reserva de assentos e liberação para taxiamento de aeronaves, controle de armazenamento das bagagens e levantamento das rotas de voos.

ETAPA 1 (tempo para realização: 5 horas)

✓ **Aula-tema: Introdução à Estrutura de Dados. Alocação Estática de Memória.**

Esta atividade é importante para que você conheça os fundamentos de Estruturas de Dados e à Alocação Estática de Memória.

Para realizá-la, devem ser seguidos os passos descritos.

PASSOS

Passo 1 (Aluno)

1. Fazer a leitura do capítulo 1: **Introdução às Estruturas de Dados** do livro texto da disciplina de Estrutura de Dados (TENEMBAUM, A.; LANGSAM, Y.; AUGESTEIN, M. **Estrutura de Dados usando C**. 1ª ed. São Paulo: *Pearson*, 2005. p. 1 a 83) focando a leitura em Vetores em C e implementação de Estruturas em C.
2. Fazer a leitura do material de aula: **Vetores e Matrizes.** Disponível em: <https://docs.google.com/open?id=0B_uLQd5hdqlWcnBnQVk0YVpTUXFzUmMyc1NWaVpvQQ>. Acesso em: 26 mar. 2012. Buscar compreender o conceito de Alocação Estática de Memória que faz a utilização de Vetores e Matrizes.

Passo 2 (Equipe)

Fazer a discussão do conteúdo de texto e da aula lida, que será utilizado para produzir relatório e implementação de uma estrutura, com destaque para:

1. alocação de Memória;
2. ponteiros em C;
3. estruturas de dados e C;
4. vetores em C;
5. implementando vetores unidimensionais;
6. operações com *strings* de caracteres;
7. estruturas em C;
8. implementando estruturas.

Passo 3 (Equipe)

Fazer as seguintes atividades:

1. Descrever e exemplificar o que é alocação estática de memória.
2. Fazer um programa em Linguagem C que implemente uma estrutura *avião (struct aviao)*, permitindo o cadastro e a exibição, com os seguintes campos:
   - modelo;
   - fabricante;
   - passageiros;
   - comprimento;
   - altura;
   - velocidade;
   - altitude;
   - motor.
3. Considerar como exemplo a criação da estrutura aluno:

```
struct aluno
{
  int ra;
  char nome[30];
  char endereco[50];
  char curso[60];
  char email[60];
};
```

Passo 4 (Equipe)

Elaborar e entregar ao professor um relatório com o nome de **Relatório 1 - Estrutura de Dados** contendo as atividades desenvolvidas nos passos anteriores.

ETAPA 2 (tempo para realização: 5 horas)

✓ **Aula-tema: Listas, Filas e Pilhas. Introdução à Alocação Dinâmica de Memória. Alocação Dinâmica de Memória e Listas Simplesmente Encadeadas.**

Esta atividade é importante para que você aplique os conceitos de Alocação Dinâmica de Memória e das Listas Simplesmente Encadeadas para desenvolver programas em Linguagem C para controle da lista de voos e passagens aéreas.

Para realizá-la, devem ser seguidos os passos descritos.

PASSOS

Passo 1 (Aluno)

1. Fazer a leitura do material de aula: **Alocação Dinâmica de Memória**. Disponível em: <https://docs.google.com/open?id=0B_uLQd5hdqlWTVFzdzZRYTZSSDJRRm9hLVg1RjRxZw>. Acesso em: 26 mar. 2012. Focar a leitura no funcionamento da alocação dinâmica de memória, função *malloc()* e função *free()*.
2. Fazer a leitura do capítulo 4: **Filas e Listas** do livro texto da disciplina de Estrutura de Dados (TENEMBAUM, À.; LANGSAM, Y.; AUGESTEIN, M. **Estrutura de Dados usando C**. 1ª ed. São Paulo: *Pearson*, 2005. p. 223 a 265) focando a leitura na implementação de Listas Ligadas.
3. Fazer a leitura do material de aula: **Lista Simplesmente Encadeada.** Disponível em: <https://docs.google.com/open?id=0B_uLQd5hdqlWQXdYT19jQUJUd2Vob1BYQ3dWZEkydw>. Acesso em: 26 mar. 2012. Focar a leitura na implementação de listas simplesmente encadeada.

Passo 2 (Equipe)

1. Fazer um programa em Linguagem C que criem uma estrutura de dados *voo (struct voo)* para a empresa VOEBEM, que servirá para implementar funções para controle da lista dos voos. A estrutura *voo* deve ser implementada de forma dinâmica e deve conter os seguintes dados:
   - número do voo;
   - data do voo;
   - horário do voo;
   - aeroporto de Saída;
   - aeroporto de Chegada;
   - rota;
   - tempo estimado de voo;
   - passageiros a bordo;
2. Implementar a função ***cadastrarVoo()*** que deve permitir o cadastro de um novo voo;
3. Implementar a função ***consultaVoo()*** que deve permitir obter as informações do voo com base na digitação do número do voo;
4. Implementar a função ***removeVoo()*** que permita a exclusão de um determinado voo.

Passo 3 (Equipe)

1. Fazer um programa em Linguagem C que implemente a estrutura *passagem (struct passagem)* para a empresa VOEBEM que servirá para implementar funções para controle da lista de passagens aéreas vendidas. A estrutura *passagem* deve ser implementada de forma dinâmica e deve conter os seguintes dados:
   - número da Passagem;
   - número do Voo;
   - data Embarque;
   - horário de Embarque;
   - portão de Embarque.
2. Implementar a função *cadastrarPassagem()* que deve permitir o cadastro de uma nova passagem;
3. Implementar a função *consultaPassagem()* que deve permitir obter informações sobre a passagem com base na digitação do número da passagem.

Passo 4 (Equipe)

Elaborar e entregar ao professor um relatório com o nome **Relatório 2 - Listas Ligadas**, contendo as atividades desenvolvidas nos passos anteriores desta etapa. Para os programas, apresentar no relatório o código fonte e telas do uso do sistema.

ETAPA 3 (tempo para realização: 5 horas)

✓ **Aula-tema: Listas, Filas e Pilhas. Filas e Pilhas com Alocação Dinâmica.**

Esta atividade é importante para que você manipule Filas e Pilhas utilizando a alocação dinâmica.

Para realizá-la, devem ser seguidos os passos descritos.

PASSOS

Passo 1 (Aluno)

1. Fazer a leitura do capítulo 4: **Filas e Listas** do livro texto da disciplina de Estrutura de Dados (TENEMBAUM, A.; LANGSAM, Y.; AUGESTEIN, M. **Estrutura de Dados usando C**. 1ª ed. São Paulo: *Pearson*, 2005. p. 207 a 218) focando a leitura na implementação de Filas em C.
2. Fazer a leitura do capítulo 2: **A Pilha** do livro texto da disciplina de Estrutura de Dados (TENEMBAUM, A.; LANGSAM, Y.; AUGESTEIN, M. **Estrutura de Dados usando C**. 1ª ed. São Paulo: *Pearson*, 2005. p. 86 a 129) focando a leitura na implementação de Pilhas em C.
3. Fazer a leitura do material de aula: **Filas.** Disponível em: <https://docs.google.com/open?id=0B_uLQd5hdqlWUVV6N0FxbzdRZm1KT0d2Y282bU1Zdw>. Acesso em: 26 mar. 2012. Focar a leitura na implementação de filas.
4. Fazer a leitura do material de aula: **Pilhas.** Disponível em: <https://docs.google.com/open?id=0B_uLQd5hdqlWUVV6N0FxbzdRZm1KT0d2Y282bU1Zdw>. Acesso em: 26 mar. 2012. Focar a leitura na implementação de pilhas.

Passo 2 (Equipe)

1. Fazer um programa em Linguagem C que implemente a estrutura *taxiamento (struct taxiamento)* para controlar a Liberação para Taxiamento das Aeronaves para decolagem na pista de voos. O taxiamento e as decolagens devem obedecer uma fila para não haver choques entre as aeronaves a decolar e que estão pousando. A estrutura *taxiamento* deve ser implementada de forma dinâmica e deve conter as seguintes informações:
   - número do voo;
   - modelo da Aeronave;
   - empresa Aérea;
   - horário de Saída;
2. Implementar a função *cadastrarTaxiamento()* que deve cadastras as informações do taxiamento e decolagem com as informações acima. O cadastro deve obedecer à disciplina de inserção dos dados FIFO (*First In First Out*).
3. Implementar a função *autorizaTaxiamento()* que deve apresentar qual a sequencia de taxiamentos e decolagens das aeronaves.

Passo 3 (Equipe)

1. Fazer um programa em Linguagem C que implemente a estrutura *bagagem (struct bagagem)* para o controle de armazenamento das bagagens no compartimento de cargas. O programa deve controlar o empilhamento e desempilhamento das bagagens e ser implementado de forma dinâmica, por meio das seguintes informações:
   - código da Bagagem;
   - número da Passagem;
   - número do Voo;
   - data Embarque;
   - horário de Embarque;
   - portão de Embarque.
2. Implementar a função *cadastraBagagem()* que deve permitir o cadastro de bagagens dos passageiros. O cadastro deve obedecer à disciplina de inserção dos dados LIFO (*Last In First Out*).
3. Implementar a função *recuperaBagagem()* que deve resgatar as bagagens que foram empilhadas pela função *cadastraBagagem()*.

Passo 4 (Equipe)

Elaborar e entregar ao professor um relatório com o nome **Relatório 3 - Filas e Pilhas**, contendo as atividades desenvolvidas nos passos anteriores desta etapa. Para os programas, apresentar no relatório o código fonte e telas do uso do sistema.

ETAPA 4 (tempo para realização: 5 horas)

✓ **Aula-tema: Grafos.**

Esta atividade é importante para que você conheça e aplique os conceitos e aplicação de Grafos em Linguagem C.

Para realizá-la, devem ser seguidos os passos descritos.

PASSOS

Passo 1 (Aluno)

1. Fazer a leitura do capítulo 8: **Grafos e Suas Aplicações** do livro texto da disciplina de Estrutura de Dados (TENEMBAUM, A.; LANGSAM, Y.; AUGESTEIN, M. **Estrutura de Dados usando C**. 1ª ed. São Paulo: *Pearson*, 2005. p. 408 a 425) focando a leitura na aplicação e representações de Grafos em C.
2. Fazer a leitura do material de aula: **Grafos e Suas Aplicações.** Disponível em: <https://docs.google.com/open?id=0B_uLQd5hdqlWdU9tQi1yS1pTQXUzNEQ1d3BsbkZBUQ>. Acesso em: 26 mar. 2012. Foque a leitura na implementação de filas.

Passo 2 (Equipe)

Fazer a discussão da leitura do capítulo do livro texto e do material de aula, que será utilizado como base para a implementação de rotas de voos por meio da estrutura grafo, com destaque para:

1. Representação de Grafos em C.
2. Algoritmo de Menor Caminho.
3. Representação por meio de Matriz de Adjacência.
4. Caminhamento em Amplitude.
5. Caminhamento em Profundidade.

Passo 3 (Equipe)

1. Fazer um programa em Linguagem C que implemente um levantamento de rotas entre uma Cidade A e uma Cidade B por meio de um Grafo, utilizando Matriz de Adjacência. Considerar um total de dez cidades e façam a ligação entre elas considerando as distâncias e tempo voo. Para percorrer da Cidade A para a Cidade B, considerar como melhor opção a distância e o tempo de voo. A Figura 1 apresentada a seguir, representa um mapa de rotas de voos de cidades brasileiras. Considerar, por exemplo, o deslocamento entre a cidade de Belo Horizonte à Fortaleza. Podem-se considerar as seguintes rotas de voos:
    1. Belo Horizonte → Fortaleza.
    2. Belo Horizonte → São Luís → Fortaleza.
    3. Belo Horizonte → Recife → Fortaleza.
    4. Belo Horizonte → Salvador → Recife → Fortaleza.

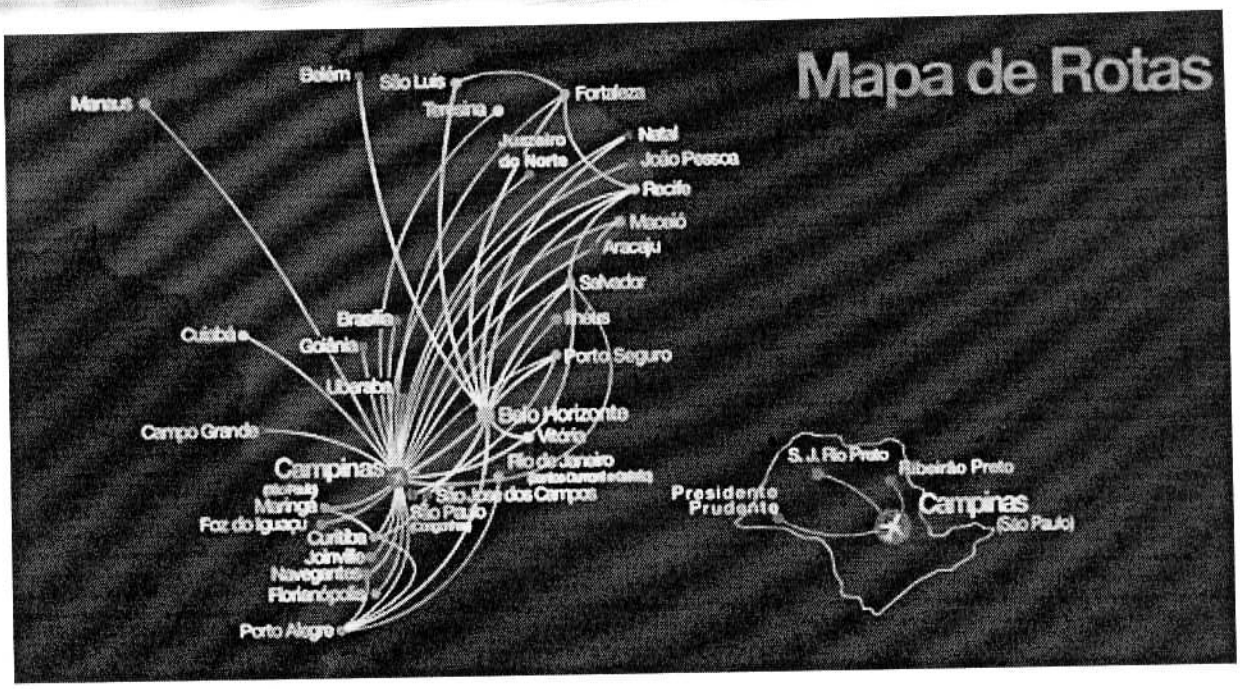

Figura 1 – Mapa de Rotas
Fonte: <http://donome.com.br/rotas-de-voos-da-azul-linhas-aereas-nacionais-internacionais-e-aeroportos/>. Acesso em: 28 mar. 2012.

2. Implementar a função *montarGrafo()* que permita cadastrar, aleatoriamente, dez cidades e determinar se há ligação entre elas e qual a distância de voo entre as cidades.
3. Implementar a função *caminhaGrafo()* que permita determinar qual o melhor caminho entre uma cidade A e uma cidade B, considerando a distância e a ligação entre as cidades.

Passo 4 (Equipe)

Elaborar e entregar ao professor um relatório com o nome **Relatório 4 - Grafos**, contendo as atividades desenvolvidas nos passos anteriores desta etapa. Para os programas, apresentar no relatório o código fonte e telas do uso do sistema.

Livro Texto da Disciplina:

TENEMBAUM, A.; LANGSAM, Y.; AUGESTEIN, M. *Estrutura de Dados usando C*. 1ª ed. São Paulo: *Pearson*, 2005.