ATIVIDADES PRÁTICAS SUPERVISIONADAS

Sistemas de Informação

6ª. Série

Programação Concorrente

A Atividade Prática Supervisionada (ATPS) é um procedimento metodológico de ensino-aprendizagem desenvolvido por meio de etapas, acompanhadas pelo professor, e que tem por objetivos:

- ✓ Favorecer a autoaprendizagem do aluno.
- ✓ Estimular a corresponsabilidade do aluno pelo seu aprendizado.
- ✓ Promover o estudo, a convivência e o trabalho em grupo.
- ✓ Auxiliar no desenvolvimento das competências requeridas para o exercício profissional.
- ✓ Promover a aplicação da teoria na solução de situações que simulam a realidade.
- ✓ Oferecer diferenciados ambientes de aprendizagem

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

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

AUTORIA:

Maurício Rodrigues de Morais

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.

- ✓ Iniciar, projetar, desenvolver, validar e gerenciar qualquer projeto de software.
- ✓ Identificar, analisar, documentar e solucionar problemas e necessidades passíveis de solução via computação.
- ✓ Assimilar criticamente conceitos que permitam a apreensão de práticas e teorias;
- ✓ Saber conciliar teoria e prática.

Produção Acadêmica

Relatórios parciais, com os resultados das pesquisas realizadas nas Etapas 1 até 4:

Participação

Para a elaboração dessa atividade, os alunos deverão previamente organizar-se em equipes com o número de participantes definido pelo professor e entregar seus nomes, RAs e *e-mails* ao professor da disciplina. Essas equipes serão mantidas durante todas as etapas.

DESAFIO

Um dos maiores desafios no mercado altamente competitivo de hoje é o que se denomina Desafio do Fornecimento. As empresas competem de maneira inovadora e agressiva para que seus produtos e serviços sejam entregues aos clientes no menor tempo e com o menor custo possível. Nesse cenário, o uso de soluções e ferramentas computacionais tem se mostrado um diferencial estratégico significativo. Mais e mais empresas estão investindo na modernização de seus *softwares* de apoio para que se destaquem em relação aos seus rivais de mercado.

Uma das empresas diretamente envolvidas nessa disputa é a **Tanngrísnir Logística**. Essa empresa de transportes é responsável pela entrega dos produtos de uma série de grandes fornecedores e, historicamente, é muito conceituada em seu segmento de mercado. Porém, com o advento das vendas pela *Internet*, a empresa tem percebido que sua capacidade de processamento de pedidos está, gradativamente, se mostrando aquém do necessário. Se continuar nesse ritmo de queda, em pouco tempo a Tanngrísnir deixará de ser a empresa de destaque que tem sido no último século.

Ao perceber a necessidade de mudança, a empresa decidiu que era hora de rever o processo de atendimento a pedidos. Atualmente, é feita a recepção por telefone e/ou por seu site dos pedidos para entrega de produtos. Apesar de permitir a entrada de vários pedidos ao mesmo tempo (são 50 atendentes por telefone e uma capacidade de 300 solicitações simultâneas pela *Internet*) o processamento interno dos pedidos é feito por um *software* que os analisa um de cada vez. Foi essa a restrição no fluxo identificada pela equipe interna de Tecnologia da Informação (TI).

Como todos os recursos humanos de TI da empresa estão alocados em outros projetos e dispõem de pouco ou nenhum tempo para a realização dessa modernização, o diretor de TI da Tanngrísnir decidiu abrir uma concorrência no mercado para que diversas empresas

apresentem uma solução para melhorar a capacidade de processamento de pedidos. A ideia geral é que seja implementada uma solução concorrente assíncrona para processamento dos pedidos, seguindo o clássico problema *Produtor-Consumidor*.

Para essa concorrência, não será necessária a implementação de um *software* completo, mas apenas um protótipo que torne possível evidenciar os resultados de uma futura implementação formal. Os pedidos serão enviados em um formato de dados que consiste de um identificador numérico com 20 dígitos e um pacote de dados em formato de texto de 1000 caracteres. Os clientes farão uso da nova ferramenta para alimentar um *buffer* interno com capacidade para 5000 pedidos. Um processo interno assíncrono (ou seja, os clientes não esperarão *online* pela confirmação, mas receberão uma resposta posterior) consumirá os pedidos e os processará individualmente.

Este desafio deverá ser realizado em quatro etapas por um grupo alunos. Em cada uma das etapas uma parte do protótipo será desenvolvida e/ou melhorada até que, ao final, seja possível reconhecer a validade da proposta representada pela nova tecnologia. Com este desafio você poderá aplicar vários conceitos importantes em desenvolvimento de aplicativos concorrentes e perceber o potencial de mercado desse tipo de solução.

Para este desafio será utilizado o seguinte livro-texto: TOSCANI, Simão Sirineo; OLIVEIRA, Rômulo; CARISSIMI, Alexandre da Silva. **Sistemas Operacionais e Programação Concorrente**. 1ª ed. Porto Alegre: Sagra Luzzatto, 2003.

Objetivo do Desafio

Elaborar um conjunto de relatórios com resultados de uma implementação de um protótipo de software que torne possível evidenciar os resultados de uma futura implementação formal.

Livro Texto da Disciplina

A produção desta ATPS é fundamentada no livro-texto da disciplina, que deverá ser utilizado para solução do desafio:

OLIVEIRA, Rômulo Silva de. ***Sistemas Operacionais***. 4ª ed. Porto Alegre: Bookman, 2010.

ETAPA 1 (tempo para realização: 5 horas)

- ✓ **Aula-tema: Revisão sobre Sistemas Operacionais. Conceitos básicos de sistemas operacionais e multiprogramação. Processos concorrentes.**

Esta atividade é importante para que você utilize os conceitos iniciais vistos em sala de aula para definir qual a melhor linguagem de programação a utilizar e criar os primeiros componentes do protótipo. Serão utilizados conceitos de multiprogramação, *Threads* e Programação concorrente.

Para realizá-la, é importante seguir os passos descritos.

PASSOS

Passo 1 (Aluno)

Ler os conceitos introdutórios sobre Programação Concorrente, Multiprogramação e *Threads*, existentes no livro-texto da disciplina. Além disso, releia a revisão sobre Sistemas Operacionais, apresentada pelo seu professor.

Passo 2 (Equipe)

Auxiliar a equipe interna de TI da Tanngrísnir a escolher qual deverá ser a linguagem de programação que será utilizada, bem como estimar alguns parâmetros iniciais de concorrência, é o objetivo dessa primeira etapa. Para isso, deverão ser criados agentes baseados em *thread* nas linguagens C e Java que consumam, de um *buffer* de 5000 posições previamente preenchido (ou seja, será implementado apenas o consumidor), cada um dos pedidos (um *struct* na implementação em C e um objeto na implementação em Java). O consumidor eliminará continuamente os pedidos do *buffer* e seu tempo de processamento será simulado por uma pausa de 10000 milissegundos. Ao final de cada processamento, deverá ser preenchido um *log* com a identificação da *thread*, a identificação do pedido, o horário de início e o horário de término do processamento. Quando o *buffer* estiver esgotado, as *threads* serão bloqueadas. Por ora, não há a necessidade de se preocupar com a exclusão mútua necessária entre os diversos consumidores.

Passo 3 (Equipe)

Fazer um experimento no qual sejam iniciadas quantidades distintas de *threads* simultâneas (1, 10, 50, 100, 500, 1000). Para cada uma dessas quantidades, faça 10 execuções – até que o *buffer* esteja vazio - e armazenem o tempo de execução para uma delas. Criar uma tabela na qual conste o tempo de cada um dos testes para as duas linguagens e o tempo médio para cada quantidade. Fazer ainda um quadro de resumo comparativo no qual constem os tempos médios de execução para cada quantidade de *threads* e o tempo de desenvolvimento por linguagem. Por fim, traçar um gráfico dos tempos médios e as quantidades de *threads*.

Passo 4 (Equipe)

Entregar ao professor da disciplina:

1. Uma mídia contendo o código fonte comentado do programa, ou seja, todos os arquivos do projeto desenvolvido no Passo 2.
2. Um documento impresso, denominado Relatório da Etapa 1, constituído de capa, o código fonte comentado e o relatório de testes desenvolvido no Passo 3.

Instrução: verificar com o professor da disciplina o tipo de mídia (CD ou *e-mail*, por exemplo) a ser entregue.

ETAPA 2 (tempo para realização: 5 horas)

- ✓ **Aula-tema: Processos concorrentes. Sincronização. *Deadlocks*.**

Esta atividade é importante para que você aprimore seus conhecimentos sobre os principais problemas da programação concorrente e entenda os cuidados necessários ao se trabalhar com recursos compartilhados.

Para realizá-la, é importante seguir os passos descritos.

PASSOS

Passo 1 (Aluno)

Ler os conceitos estudados sobre Exclusão Mútua e Objetos Compartilhados no livro-texto da disciplina. É importante que você esteja familiarizado com os diversos algoritmos apresentados e os conceitos subjacentes para que possa avançar no desenvolvimento do seu protótipo.

Passo 2 (Equipe)

Alterar o projeto criado na etapa anterior, de modo que agora o *buffer* seja alimentado por uma *thread* Produtora. O *buffer* de pedidos é o objeto compartilhado e deve- se garantir que a ação de produtores e consumidores seja feita de maneira exclusiva, ou seja, deverá haver exclusão mútua entre os processos para evitar colisões. Os produtores deverão criar as estruturas de dados referentes aos pedidos e seu tempo de processamento deverá ser simulado por uma pausa de 5000 milissegundos. Cada produtor alimentará o *buffer* continuamente, enquanto houver espaço no mesmo, e deverá ser bloqueado quando estiver cheio. Quando houver um novo espaço no *buffer,* o Produtor deverá ser reativado. Deve, ainda, alimentar um *log* com a identificação da *thread*, o horário de início e o horário de término do processamento. A partir desse ponto, o gerente do projeto (seu professor) escolherá qual linguagem deverá ser utilizada nesta e nas próximas etapas.

Passo 3 (Equipe)

Fazer um experimento no qual sejam iniciadas quantidades distintas de *threads* simultâneas (1, 10, 50, 100, 500, 1000), tanto consumidoras quanto produtoras. Para cada uma dessas quantidades, fazer 10 execuções com tempo fixo de 3 minutos e contabilizar a quantidade de pedidos que foram processadas. Criar uma tabela na qual conste a quantidade de pedidos para cada um dos testes, a média de pedidos processados e a quantidade de pedidos por segundo. Fazer ainda um gráfico das quantidades médias de pedidos e os números de *threads*.

Passo 4 (Equipe)

Entregar ao professor da disciplina:

a) Uma mídia contendo o código fonte comentado do programa, ou seja, todos os arquivos do projeto desenvolvido no Passo 2.
b) Um documento impresso, denominado Relatório da Etapa 2, constituído de capa, o código fonte comentado e o relatório de testes desenvolvido no Passo 3.

Instrução: verificar com o professor da disciplina o tipo de mídia (CD ou *e-mail*, por exemplo) a ser entregue.

ETAPA 3 (tempo para realização: 5 horas)

- ✓ **Aula-tema: Tratamento de sinais. Mecanismos de IPC (*Inter Process Communication*). Semáforos.**

Esta atividade é importante para que você aplique os novos conhecimentos vistos em sala de aula sobre exclusão mútua com semáforos para tornar seu protótipo mais robusto e compreensível.

Para realizá-la, é importante seguir os passos descritos.

PASSOS

Passo 1 (Aluno)

Ler os conceitos estudados sobre Semáforos no livro-texto da disciplina. É importante que você esteja familiarizado com o uso dessa técnica para que possa aprimorar a exclusão mútua do seu protótipo.

Passo 2 (Equipe)

Alterar o protótipo atual de modo que todo o processo de exclusão mútua seja feito por semáforos. Certificar-se de que o conceito de semáforo esteja sendo empregado corretamente e não haja falhas conceituais.

Passo 3 (Equipe)

Fazer novamente o experimento feito no Passo 3 da etapa anterior, agora com a nova versão baseada em semáforos. Comparar os resultados obtidos agora com os obtidos anteriormente. Documentar o novo teste e escrever uma análise comparativa do impacto (se houver) apresentado pela utilização dessa nova técnica, não só em termos de desempenho, mas também em termos de simplificação do código e facilidade de manutenção futura.

Passo 4 (Equipe)

Entregar ao professor da disciplina:

1. Uma mídia contendo o código fonte comentado do programa, ou seja, todos os arquivos do projeto desenvolvido no Passo 2.
2. Um documento impresso, denominado Relatório da Etapa 3, constituído de capa, o código fonte comentado e o relatório de testes desenvolvido no Passo 3.

Instrução: verificar com o professor da disciplina o tipo de mídia (CD ou *e-mail*, por exemplo) a ser entregue.

ETAPA 4 (tempo para realização: 5 horas)

- ✓ **Aula-tema: Tratamento de sinais. Mecanismos de IPC (*Inter Process Communication*). Semáforos.**

Esta atividade é importante para que você compreenda e aplique mais um dos conceitos fundamentais de programação concorrente, a sincronização, e conheça os desafios de sua implementação.

Para realizá-la, é importante seguir os passos descritos.

PASSOS

Passo 1 (Aluno)

Ler os conceitos estudados sobre criação de sincronização no livro-texto da disciplina. É importante que você esteja familiarizado com o conceito para que possa criar aplicativos concorrentes com maior controle da execução.

Passo 2 (Equipe)

Atender a um novo requisito que surgiu para a apresentação do seu protótipo. Agora, a Tanngrísnir deseja garantir que cada um dos pedidos será processado na ordem em que foi solicitado, sem que haja "furos na fila". Vocês devem aprimorar sua implementação da *thread* consumidora de modo a garantir essa especificação. Porém, pode ser que tal requisito tenha impacto significativo na quantidade de pedidos atendidos por unidade de tempo. Por isso, mantenham o código da etapa anterior intacto (fazer uma cópia e guardem-na) para que possa comparar o resultado das implementações depois.

Passo 3 (Equipe)

Fazer novamente o experimento feito no passo 3 da etapa anterior, agora com a nova versão. Comparem os resultados obtidos agora com os obtidos anteriormente. Documentar o novo teste e escrevam uma análise comparativa do impacto (se houver) apresentado pela inclusão desse novo requisito. Apresentar, por fim, um parecer sobre os custos de se implementar tal alteração.

Passo 4 (Equipe)

Entregar ao professor da disciplina:

1. Uma mídia contendo o código fonte comentado do programa, ou seja, todos os arquivos do projeto desenvolvido no Passo 2.
2. Um documento impresso, denominado Relatório da Etapa 4, constituído de capa, o código fonte comentado e o relatório de testes desenvolvido no Passo 3.

Instrução: verificar com o professor da disciplina o tipo de mídia (CD ou *e-mail*, por exemplo) a ser entregue.

Padronização

O material escrito solicitado nesta atividade deve ser produzido de acordo com as normas da ABNT, com o seguinte padrão (exceto para produções finais não textuais):

- 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.

Para consulta completa das normas ABNT, acesse a Normalização de Trabalhos Acadêmicos Anhanguera. Disponível em: <http://issuu.com/normalizacao/docs/normaliza____o_de_trabalhos_acad__m>. Acesso em: 13 maio 2014.