

**Vídeos base:**

1) Semáforos em Java: <https://youtu.be/JgwKfdn2kDU>

2) Github:

**Tutorial Basico GitHub com Eclipse e EGit Usando Chave SSH:**

<http://www.youtube.com/watch?v=fBBSazTSGZw>

**OU**

**Usando Github com Github Desktop em Projetos**

Eclipse: <http://www.youtube.com/watch?v=EgHjIYyS4U>

**Exercícios:**

Os códigos devem ser carregados para o Github e o formulário respondido com os links dos repositórios

- 1) Um servidor com multiprocessamento recebe requisições que envolve realizar cálculos e fazer transações com bancos de dados. Por ter uma quantidade grande de núcleos de processamentos e threads, além de um bom algoritmo de escalonamento de threads, enquanto as threads fazem cálculos, estes podem ocorrer simultaneamente, mas quando se faz a transação no banco de dados, esta deve ser apenas uma thread por vez. Considere um conjunto de threads com IDs definidas na própria aplicação com números iniciando em 1 e incrementando de um em um. As threads tem comportamento como segue:
  - a) Threads com ID dividido por 3 resultando em resto igual a um fazem as transações:
    - Cálculos de 0,2 a 1,0 segundos
    - Transação de BD por 1 segundo
    - Cálculos de 0,2 a 1,0 segundos
    - Transação de BD por 1 segundo
  - b) Threads com ID dividido por 3 resultando em resto igual a dois fazem as transações:
    - Cálculos de 0,5 a 1,5 segundos
    - Transação de BD por 1,5 segundo
    - Cálculos de 0,5 a 1,5 segundos
    - Transação de BD por 1,5 segundo
    - Cálculos de 0,5 a 1,5 segundos
    - Transação de BD por 1,5 segundo
  - c) Threads com ID dividido por 3 resultando em resto igual a zero fazem as transações:
    - Cálculos de 1 a 2 segundos
    - Transação de BD por 1,5 segundo
    - Cálculos de 1 a 2 segundos
    - Transação de BD por 1,5 segundo
    - Cálculos de 1 a 2 segundos
    - Transação de BD por 1,5 segundo

Faça uma aplicação em Java que simule a situação de 21 Threads simultâneas, com exibição em console de cada passo que a Thread está realizando.

- 2) Existem diversos jogos de simulação. Um deles simula a participação de cozinheiros em uma cozinha profissional realizando pratos. Numa das fases, o cozinheiro precisa realizar o cozimento de 5 pratos simultâneos, onde cada cozimento não depende da interação do jogador. Pratos de ID ímpar, são chamados de Sopa de Cebola e levam de 0,5 a 0,8 segundos para ficar prontos. Pratos de ID par, são chamados de Lasanha a Bolonhesa e levam de 0,6 a 1,2 segundos para ficar prontos. Quando um prato inicia, é necessário comunicar, em console, que se iniciou e, a cada 0,1 segundos, deve-se exibir o percentual de cozimento (O percentual é definido pelo tempo total dividido por 0,1 segundos). Quando um prato fica pronto, é necessário comunicar em console o final e fazer a entrega, que leva 0,5 segundos. O jogador só pode entregar um prato por vez e deve comunicar a entrega. Simular a situação em Java.

3) Numa prova de triatlo moderno, o circuito se dá da seguinte maneira:

- 3Km de corrida onde os atletas correm entre 20 e 25 m / 30 ms
- 3 tiros ao alvo com pontuação de 0 a 10 (Cada tiro)
- 5 km de ciclismo onde os atletas pedalam entre 30 e 40 m/ 40 ms

25 atletas participam da prova e largam juntos, no entanto, apenas 5 armas de tiro estão à disposição. Cada atleta leva de 0,5 a 3s por tiro. Conforme os atletas finalizam o circuito de corrida, em ordem de chegada, pegam a arma para fazer os disparos. Uma vez encerrados os disparos, a arma é liberada para o próximo, e o atleta segue para pegar a bicicleta e continuar o circuito.

Para determinar o ranking final dos atletas, considera-se a seguinte regra:

- O primeiro que chegar recebe 250 pontos, o segundo recebe 240, o terceiro recebe 230, ... , o último recebe 10.

- Soma-se à pontuação de cada atleta, o total de pontos obtidos nos 3 tiros (somados)

Ordenar a pontuação e exibir o resultado final do maior pontuador para o menor.