



Objetivo

Enquanto primeira aula prática, um objetivo é o de consolidação dos grupos e estabelecimento do ambiente de validação de entre quadro de desenvolvimento e ferramentas a serem progressivamente introduzidos. Outro objetivo, como motivação inicial é a discussão do caso simples de acesso distribuído concorrente a um serviço que mantém um vetor de inteiros com N elementos iniciados com valores arbitrários. A discussão considera desde já a adoção parcial do modelo ISoS na sua estruturação dos artefactos tecnológicos enquanto Service, CES ou ISystem. Na validação deverá ser evidenciada a ocorrência de violação do invariante $\sum_{i=0}^{N-1} vetor[i] = CONST$ quando múltiplas entidades computacionais autónomas Service, enquanto clientes, acedem às operações de leitura e escrita sobre elementos do vetor, no entendimento de que apenas são admitidas transferências entre elementos. O acesso concorrente às operações disponibilizadas pelo serviço Vetor deverá ser simulado na base dos elementos Service **IsyIESD/CesVector/SerVector** e **IsyIESD/CesVector/SerVectorCli** disponibilizados. Proposta de estratégia de coordenação que garanta o invariante.

Plano de Trabalhos

A aula começa com uma apresentação do ambiente de validação e em particular aspetos sobre a abordagem à primeira prática. Iniciar-se-á com a consolidação do quadro de grupos, após o que deverão passar imediatamente à avaliação dos dois serviços oferecidos como base de trabalho. Para a avaliação dos serviços, os alunos podem obter o projeto Git do Moodle em Prática/Práticas/Prática.01, local onde se encontra este enunciado. Entretanto, durante a prática será dado acesso a todos os alunos ao repositório Git na plataforma Github: <https://github.com/ISEL-ADEETC-MEIC-IESD/iesd2021sv> de onde passarão a poder obter a versão mais atualizada dos exemplos oferecidos.

Sugere-se como ferramenta de trabalho o ambiente integrado de desenvolvimento Eclipse ([link](#)) sendo que os projetos exemplo estão estruturados na base da ferramenta Apache Maven ([link](#)), seja na representação do projeto, *Project Object Model* (POM) seja na gestão de dependências a partir de um repositório central ([link](#)) em coordenação com o repositório local, por omissão em <user.dir>\.m2 (cache local). A validação ocorrerá maioritariamente sobre o ecossistema Java.

Questões para discussão

De entre outras que poderão ser acrescentadas pelo grupo, consideram como questões de partida:

- O modelo de interação entre o Service cliente e servidor (implementa a interface vetor);
- Situação para um único cliente a aceder ao vetor;
- Múltiplos clientes sem qualquer mecanismo de coordenação;
- Discussão do ou dos mecanismos de coordenação a considerar para que seja garantido o invariante $\sum_{i=0}^{N-1} vetor[i] = CONST$;
- Questionar a hipótese de um número muito elevado de elementos Service (**SerVectorCli**) clientes do serviço vetor (**SerVector**), e.g., na ordem dos milhões, dezenas ou centenas;
- Se se tratar de um serviço crítico, i.e., não pode falhar, melhor, a probabilidade de falha deverá ser muito reduzida, podendo ser enquadrado enquanto serviço tolerante a falhas;
- Importância para o caso CMSP dos problemas, estratégias/técnicas de solução ou conceitos discutidos.

Tópicos relevantes: ordenação de eventos, exclusão mútua, atomicidade, controlo da concorrência, operações que podem gerar conflito (*conflicting operations*), bloqueios (*deadlocks*), algoritmos de eleição, procura de acordos (*agreements*), modelo de transações X/Open, protocolo *two-phase-commit* (2PC).

Resultado

Discussão do grupo com o docente, centrada nas questões desafiadas para o caso de validação, Vector e CMSP.