



Objetivo

Obter competências ao nível conceptual sobre o quadro tecnológico OSGi, i.e., sobre conjunto de especificações que disponibiliza e ambiente de execução Apache Karaf [\[ref\]](#). As especificações OSGi são atualmente da responsabilidade da fundação Eclipse e encontram-se na versão 8.0 [\[ref\]](#). Trata-se de um quadro de modularidade dinâmico para o ecossistema Java. Tem especial relevância para a discussão em IESD a especificação 100 – *Remote Services* [\(link\)](#). Considera-se como facilitador do desenvolvimento a especificação 112 – *Declarative Services* [\(link\)](#) enquanto implementação do modelo (padrão) *Dependency Injection (DI)/Inversion of Control (IoC)*. Mantém-se como desafio, a estratégia de uma infraestrutura que facilite o desenvolvimento e a operação de entidades computacionais independentes, enquanto elementos Service que implementa zero ou mais interfaces. Considera-se nesta fase, o enquadramento do exemplo de referência simples CesVector, sobre a infraestrutura OSGi (implementação Apache Karaf).

Plano de Trabalhos

Validação do conjunto de exemplos CesOSGi:

- CesOSGiAPIM (não implementado); interfaces no projeto SerOSGiHello
- CesOSGiDOE, conjunto de projetos para instanciação sobre Apache Karaf, modularidade Feature
- SerJavaHello and SerJavaHelloMain – quadro exclusivamente Java (JVM); wrap;
- SerOSGiBa/SerOSGiBas and SerOSGiBaCli/SerOSGiBasCli – BundleActivator e serviços;
- SerOSGiDs and SerOSGiDsCli – a adoção da especificação Declarative Services
- SerOSGiRs and SerOSGiRsCli – elementos serviço distribuídos (Remote Services)

Notas para a validação

É distribuído com a prática o repositório iesd2324sv atualizado na sua estrutura e resolução de inconsistências na estrutura maven dos projetos. É acrescentado o conjunto de projetos para validação do quadro OSGi e em particular a sua perspetiva de distribuição.

Como sugerido na prática.02, porque o Eclipse IDE apresenta inconsistências na sua integração de projetos Maven, o ambiente integrado de desenvolvimento (IDE) aberto NetBeans [\[ref\]](#) é sugerido como uma alternativa. É também alternativa o IDE Theia, uma iniciativa da Fundação Eclipse que se enquadra na plataforma Theia [\[ref\]](#), por sua vez, na base do código aberto do IDE VisualCode. O IDE Theia enquadra-se ainda enquanto IDE que pode ser utilizado na infraestrutura Che que gere DevWorkSpaces numa rede Kubernetes [\[ref\]](#); eventualmente a explorar em práticas posteriores.

- Numa primeira abordagem poderão considerar todas as componentes/serviços numa instância Karaf e posteriormente instâncias em servidores distintos (e.g., WSL2 ou máquina virtual na infraestrutura VirtualBox);
- Todos os exemplos têm em CesOSGiDOE um subprojecto Maven com package kar, o qual gera a entidade *feature*, na realidade, uma composição de bundles para facilitar a instalação de um conjunto de bundles. Nos ficheiros README encontram-se as instruções para a instalação isolada de cada um dos bundles de uma entidade *Feature*;

Para a versão distribuída, foi acrescentado um módulo ISosIFR com diretorias com aprovisionamento de máquinas virtuais e respetiva configuração na base da ferramenta Vagrant

- Terá que existir uma instância zookeeper a funcionar como serviço de diretoria (registry). A instanciação, nesta fase, é concretizada sobre Karaf através do conjunto de entidades *Feature*, como indicado no ficheiro *readme.adoc* do projeto KfeatureOSGiRsDOE com os artefactos para instanciação do exemplo de serviços distribuídos (Remote Services).

Considere a migração da implementação de cada grupo a partir do exemplo referência ISyIESD/CesVector, para uma versão que execute sobre OSGi, e.g., projeto a designar por CesOSGiVector com SerOSGiVector e SerOSGiVectorCli.

Para facilitar a validação são fornecidos dois projetos Vagrant, ZkUnreliableVagrant e KarafHelloVagrant, respetivamente para criar uma máquina virtual na plataforma VirtualBox para instalação do sistema Apache Zookeeper e uma instalação com o ambiente de execução Apache Karaf. A ferramenta Vagrant é resumida em documento anexo a esta prática.

Questões para discussão

Esquematização e discussão de estratégia de validação do invariante quando existam múltiplos serviços vetor a ser acedidos por múltiplos clientes. Recorda-se que neste caso o invariante refere-se ao somatório dos somatórios dos serviços vetor envolvidos. Avaliação de vantagens/desvantagens na utilização do quadro OSGi, Remote Services, na validação da coordenação distribuída que garanta a atomicidade dos acessos Leitura/escrita nos serviços vector. Poderão adotar outros quadros tecnológicos desde que devidamente fundamentados e preserve a logica SOA.

Resultado

Discussão do grupo com o docente, centrada nas questões coordenação da concorrência entre elementos Service, distribuídos. Na prática de 4 de abril deverá ser disponibilizado um relatório de uma página em ficheiro associado gxx-iesd-prática03.pdf para discussão com o docente.

Luís Osório