

## Laboratório 3

### Objetivos:

- Desenvolver serviços distribuídos com tecnologia *Google Remote Procedure Call* (gRPC)
- Distinguir entre chamadas unárias e chamadas com *stream* de servidor

Pretende-se desenvolver um serviço para cálculo de números primos, semelhante ao realizado no laboratório 2, mas usando gRPC. Um cliente do serviço pode enviar pedidos de cálculo indicando o intervalo para encontrar números primos. O serviço retorna sequência de número de forma assíncrona.

- 1) Junto com o enunciado são fornecidos 3 projetos (contrato, servidor e cliente), com a estrutura apresentada na Figura 1a). Os projetos `PrimesServiceImpl` e `PrimesClientApp` têm apenas a estrutura de directorias Maven e o respetivo `pom.xml`. O projeto `PrimesServiceContract`, para além do `pom.xml`, tem o contrato definido no ficheiro `PrimesService.proto` que, por convenção do compilador de protobuf, está localizado na directoria `\src\main\proto`.

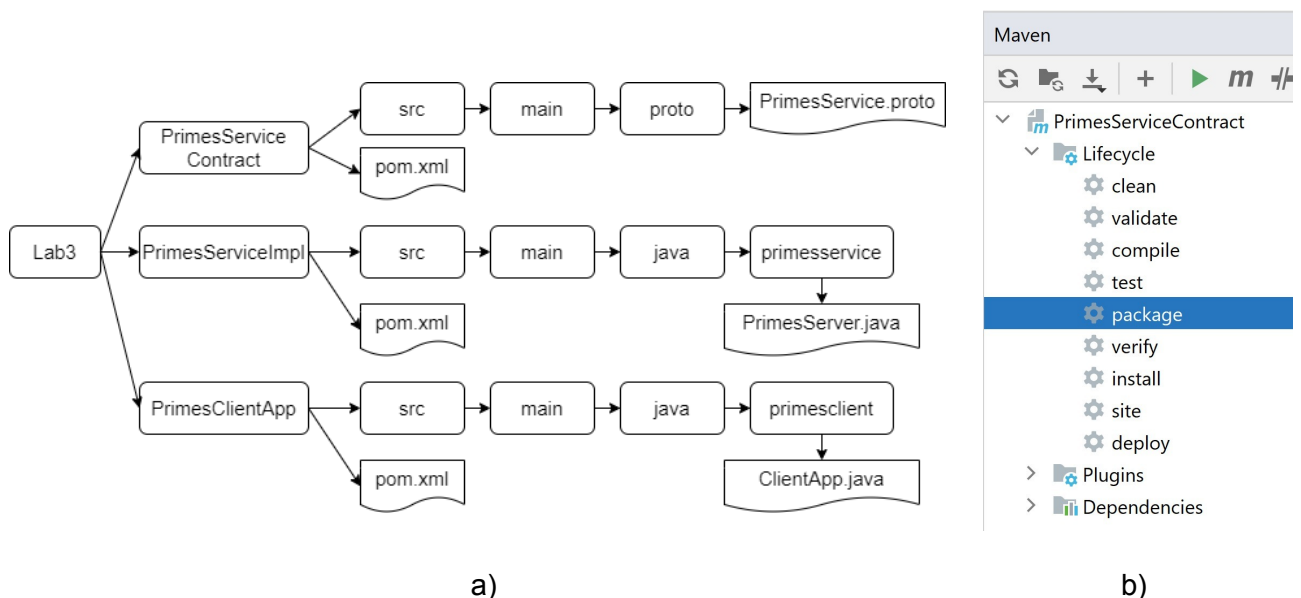


Fig. 1: a) Estrutura de directorias e ficheiros dos projetos fornecidos; b) Janela Maven no IntelliJ

No projeto `PrimesServiceContract`, usando o menu Maven no ambiente IntelliJ, gere o JAR com o contrato (ação **package**) e instale no repositório local do Maven (ação **install**).

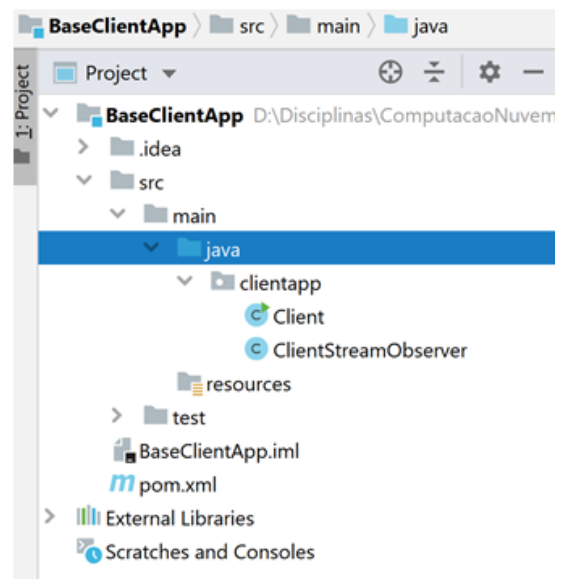
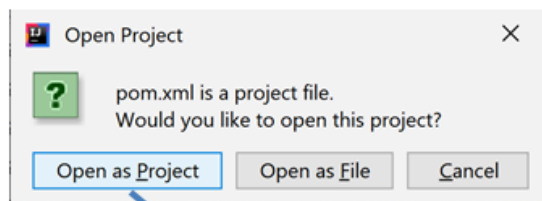
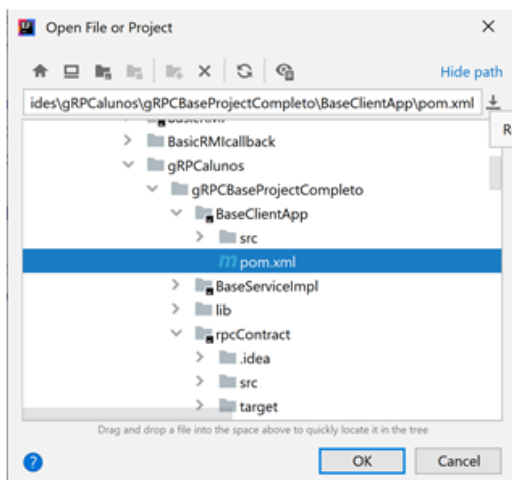
- 2) No projeto do servidor verifique que já existe no ficheiro `pom.xml` uma dependência para o artefacto contrato gerado na alínea 1. Implemente no ficheiro `PrimesServer.java` a inicialização do servidor e as operações do contrato.
- 3) No projeto do cliente verifique que já existe no ficheiro `pom.xml` uma dependência para o artefacto da contrato gerado na alínea 1. Implemente no ficheiro `ClientApp.java` a inicialização do cliente para acesso ao servidor, chamando as operações do contrato:

- a) Usando um stub bloqueante chame a operação `isAlive` para verificar a correta conectividade ao serviço
  - b) Usando um stub não bloqueante realize 5 chamadas à operação `findPrimes` para calcular os números primos entre 1 e 500, em intervalos de 100 números.
- 4) Após testes na máquina local, execute a aplicação servidor numa VM GCP, mantendo a aplicação cliente na sua máquina local. Verifique que ao contrário do problema encontrado com a solução RMI, o servidor consegue enviar os números primos à aplicação cliente, isto é, tudo funciona corretamente, independentemente da localização do cliente e do servidor. Garanta que a *firewall* da VM tem o porto usado pelo servidor aberto.

## ANEXO

Como abrir projetos *IntelliJ* vindos de outra fonte.

### Abrir o projeto selecionando na estrutura de diretorias o ficheiro `pom.xml`



### Se usar a opção *Import project* na janela inicial do IntelliJ, selecione na mesma o ficheiro `pom.xml`

