3º Trabalho Prático de Avaliação

Objectivo: Desenvolvimento de sistemas distribuídos usando arquiteturas orientadas aos serviços (SOA) com Windows Communication Foundation (WCF)

Nota: O trabalho deve ser realizado até 21 de Junho de 2015, podendo o relatório que descreva o trabalho com as opções tomadas ser entregue até 30 de Junho de 2015. (Enviar, os projetos em Zip file e relatório, para lass@isel.ipl.pt). Note que, como foi referido na apresentação da disciplina, a qualidade do relatório terá peso na avaliação do trabalho realizado. O relatório deverá permitir ao leitor entender, qual o objetivo, os requisitos funcionais e não funcionais (note que os requisitos não são a transcrição deste enunciado), a arquitetura do sistema, as interações entre as partes, bem como os aspetos relevantes da implementação dessa arquitetura realçando os pontos fortes e fracos da solução que proponha. Evite descrever código a menos que se justifique nalguma situação, por exemplo, os contratos dos serviços envolvidos.

Considere um cenário de um jogo distribuído com os seguintes requisitos (Figura 1):

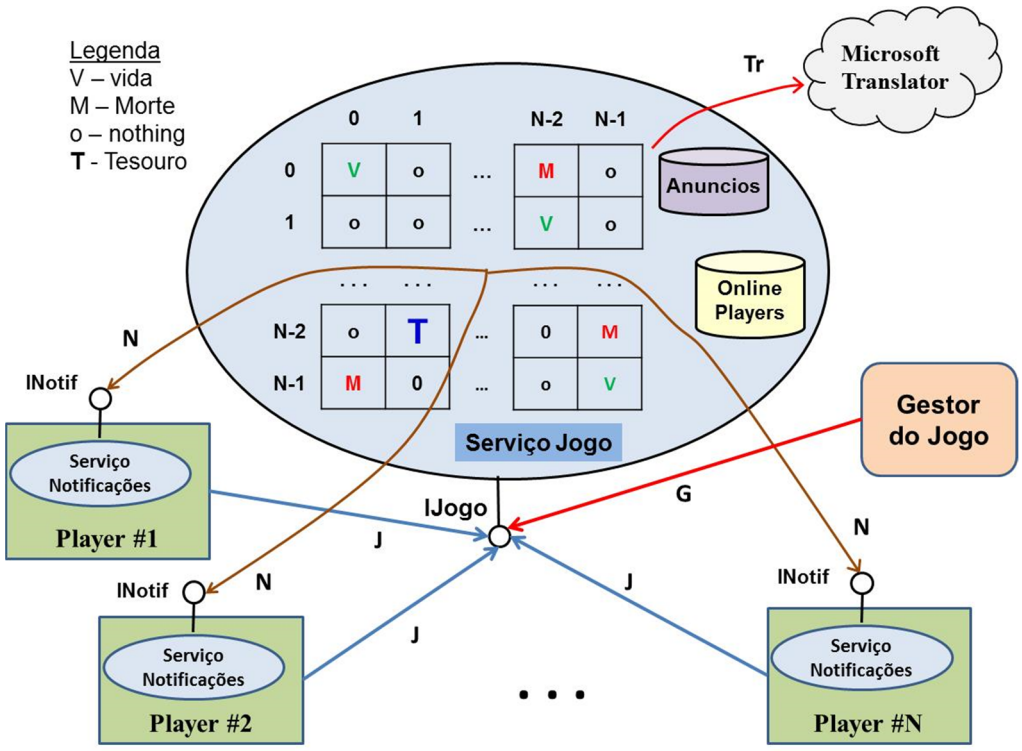

Figura 1 – Jogo distribuído com arquitetura baseada em serviços

- Existe um servidor central que aloja um serviço *Jogo* que suporta um jogo de procura de um tesouro, a ser jogado por múltiplos jogadores (*Players*) e representado numa matriz NxN preenchida aleatoriamente pelos símbolos indicados na legenda da Figura 1. O jogo tem as seguintes regras:
  - ✓ O Gestor do jogo inicia um jogo;
  - ✓ Os jogadores não conhecem o conteúdo da matriz;

- ✓ Após notificação que o jogo começou ou recomeçou, cada jogador joga indicando uma coordenada (x,y) da matriz;
- ✓ O jogo termina quando um jogador indicar a posição da matriz (x,y) que contém o tesouro (T), sendo todos os jogadores notificados de qual o jogador que ganhou;
- ✓ Um jogador morre se escolher uma posição com (M), mas se tiver anteriormente acumulado vidas (V) decrementa o número de vidas e pode continuar a jogar;
- ✓ A escolha de uma posição com (o) não tem qualquer significado, permitindo que o jogador continue a jogar;
- ✓ Um jogador que tente jogar sem o jogo estar iniciado deve receber uma exceção indicando que o jogo não está em curso;

- Existe uma aplicação *Player* que, para além de ser usada pelos jogadores, aloja um serviço de notificações para que o serviço jogo possa notificar os jogadores de eventos ocorridos durante o jogo, nomeadamente a indicação que o jogo terminou após a vitória de um qualquer jogador;
- Existe uma aplicação usada pelo gestor do jogo (*Gestor*) que permite iniciar o jogo e inserir anúncios publicitários (frases simples numa língua qualquer) que serão difundidos para os jogadores em jogo através do serviço de notificações;
- O serviço Jogo acede ao serviço *Microsoft Translator* sempre que tiver que traduzir anúncios de acordo com o perfil dos diversos jogadores;
- Embora faça parte do trabalho definirem as interações/ações (J,N,G,Tr) Assim existem as seguintes interações e ações:

  J: Registo de um jogador no serviço Jogo, indicando um nome que o permita identificar, a língua (Port, Ing,…) da sua preferência, bem outros detalhes que ache pertinentes; Jogar numa posição (x,y); Desistir do jogo;

  N: Notificação de início e fim de jogo; Notificação de anúncios; Notificação de número de vidas ou morte; Notificação que o jogo foi suspenso pelo Gestor;

  G: Iniciar, terminar ou suspender um jogo; Publicar anúncios indicando a língua respetiva;

  Tr: Quando necessário traduz um anúncio de uma língua para outra;

Sugestões:

1. Qualquer questão ou dúvida sobre requisitos, deve ser discutida com o professor;
2. Antes de começar a escrever código desenhe a arquitetura do sistema, os contratos dos serviços bem como os diagramas de interação mais importantes;
3. Para acesso ao serviço Microsoft Translator reveja os exercícios realizados nas aulas;
4. Deve utilizar ficheiros de configuração, simplificando assim a construção de um protótipo de demonstração com pelo menos 3 utilizadores da aplicação *Player*;
5. Tenha em atenção o tratamento e propagação de exceções utilizando *Fault Contracts*.
6. No relatório discuta e justifique as opções tomadas, por exemplo, o modo de instanciação, o controlo de concorrência, as opções escolhidas para os contratos, *Endpoints*, nomeadamente os *Bindings* etc. Por exemplo o serviço *Jogo* deve disponibilizar dois *Endpoints* diferentes, um para os jogadores e outro para o *Gestor*;
7. Quando tiver questões sobre os requisitos, verifique no site Moodle de suporte a SD se existem "*Frequently Asked Questions*" com esclarecimentos sobre o trabalho.

Luís Assunção