

# Jogo Multi-Utilizador Tolerante a Faltas: Pong

João Lopes  
26738

Ivan Passos  
30113

Rui Faial  
23053

## Abstract

*Este artigo descreve um jogo, o Pong, jogado por um máximo de 4 jogadores implementado numa arquitectura distribuída onde a desistência de um jogador ou a falha de um servidor não significa o fim do jogo.*

## 1 Introdução

Actualmente o uso de sistemas distribuídos dos está tão difundido que quase passa despercebido, mas o seu uso trouxe-nos grandes vantagens e alguns problemas para resolver. Citando Leslie Lamport "Um sistema distribuído é aquele que te impede de trabalhar devido a uma falha numa máquina da qual nunca ouviste falar". Falhas essas que têm de ser resolvidas. No âmbito da cadeira Tolerância a Faltas Distribuídas iremos desenvolver sistemas distribuídos tolerantes a faltas.

O Pong não é excepção. Para a resolver as falhas que possam ocorrer, propomos a implementação de uma arquitectura de servidores replicados onde é elegido um coordenador. O coordenador distingue-se dos outros servidores pois é o único que realiza a computação necessária para o bom funcionamento do jogo dos clientes. Cada interveniente guarda o último estado do jogo.

## 2 Jogo

O Pong é um jogo composto por um conjunto entre 2 a 4 jogadores, uma bola e 2 a 4 balizas. Cada jogador controla uma pequena barra e é responsável por uma das paredes do quadrado (baliza).

O objectivo do jogo é evitar que a bola toque na baliza, utilizando para isso a barra, o que fará com que a bola se desloque em direcção à baliza de um outro jogador. É atribuído um ponto ao jogador que deixou embater a bola na sua baliza. O jogo finda quando um dos jogadores atingir uma pontuação de 20 pontos, ou quando só resta um jogador. No final, ganha o jogador que tiver menor pontuação ou o último jogador em jogo.

## 3 Arquitectura

Com o objectivo de potencialmente eliminar as possíveis faltas no sistema, iremos desenvolver uma arquitectura baseada em servidores replicados onde apenas um deles é que toma decisões e os outros guardam o último estado do coordenador, formando assim o grupo de servidores que irão comunicar com o grupo dos clientes.

### 3.1 Comunicação

A comunicação entre o grupo dos servidores será feita usando o Protocolo de Difusão Fiável Uniforme. Este mecanismo evita a integração de um Servidor auxiliar para integração ou exclusão de servidores no grupo abolindo assim mais um potencial ponto de falha. No entanto cada servidor deverá conter informação acerca dos endereços (IP) dos restantes membros do grupo.

A mesma estratégia será utilizada para o grupo dos clientes que posteriormente irá comunicar com o grupo dos servidores. Para tal faremos uso das vantagens do APPIA.

### 3.2 Noção de estado

Definimos estado como sendo num determinado instante: a pontuação e a posição da barra de cada jogador; a posição, direcção e velocidade da bola.

### 3.3 Coerência de estado

A coerência de estado dentro de um sistema de servidores replicados e multi-utilizador, implica que num dado momento todos os seus elementos possuam o mesmo estado. Esta propriedade será conseguida através dos protocolos de ordenação, da gestão de grupos e da difusão uniforme de mensagens oferecidos pelo APPIA.

Através da gestão de grupos do APPIA, obtém-se uma vista para o grupo dos servidores e uma vista para

o grupo dos clientes. Entende-se por vista uma lista dos elementos activos de um grupo. As vistas são actualizadas consoante a saída ou entrada de novos elementos. Quando um elemento entra ou sai de um grupo, gerada uma nova vista do grupo. Para o grupo de clientes, a vista terá um máximo de 4 clientes, sendo os restantes descartados. Aquando da entrada de um novo elemento, é enviada uma mensagem ao coordenador com a nova vista, este compara com a antiga e actualiza o seu estado enviando o novo estado para todos os intervinientes. Para o grupo de servidores a saída de um elemento da vista, gera uma actualização da vista e caso esse elemento seja o coordenador, é realizada uma eleição para um novo coordenador.

### 3.4 Eleição do coordenador

Para a eleição do coordenador iremos implementar um algoritmo com as seguintes regras:

- O processo mais antigo é o coordenador.
- O primeiro processo a entrar na vista do grupo dos servidores, será o coordenador.
- No caso do coordenador falhar, o processo imediatamente a seguir na vista passa a ser o coordenador.
- A entrada de um novo processo leva á sua colocação no fim da fila dos servidores.

## 4 Fila protocolar

Iremos implementar a pilha protocolar usando a interface xml que o APPIA possui. A nossa pilha protocolar irá ser constituída pelos seguintes elementos:

ApplicationLayer
TotalAbstractLayer
VSynclayer
LeaveLayer
StableLayer
HealLayer
InterLayer
IntraLayer
SuspectLayer
MergeOutLayer
GossipOutLayer
GroupBottomLayer
TCPCompleteLayer

## 5 Referências

Paulo Veríssimo e Luis Rodrigues,  
 "Distributed Systems dos System Architects",  
 Klumer Academic Publishers ISBN 0-7923-7266-2

URL: <http://appia.di.fc.ul.pt>

Pinto, "Appia Group Communication Manual", 2001

Fernando Felício, Susana Guedes e Valter Conceição,  
 Relatório de TFD 2003

José Mocito, Liliana Rosa e Nuno Almeida  
 Relatório de TFD 2003