

Relatório Trabalho Prático II

Sistemas Distribuidos - 2017/2018 - 2ºSemestre

Realizado por :

Pedro Silva - 72645 - pedro.mfsilva@ua.pt

Francisco Teixeira - 67438 - franciscoteixeira@ua.pt

Turma P3

Grupo 2

Introdução

No âmbito da unidade curricular de Sistemas Distribuídos foi nos proposta a realização do assignment 2 “An Afternoon at the races”.

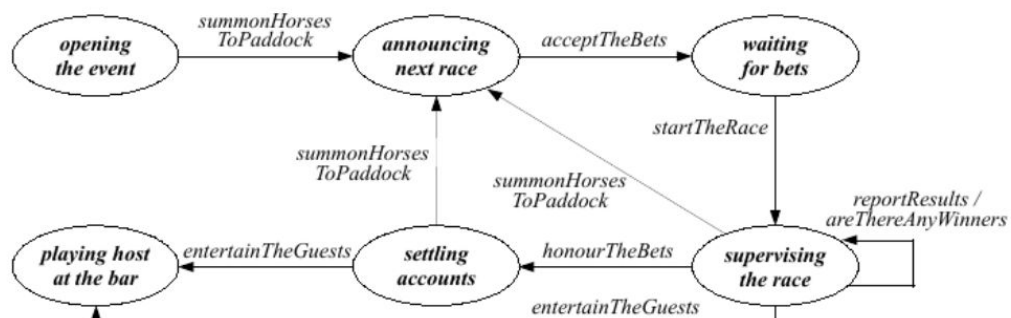
Este trabalho consiste em passar a solução multithreading de uma corrida de cavalos obtida no assignment 1 numa solução distribuída para correr em várias máquinas, onde ocorre comunicação entre diversas instâncias de threads diferentes e monitores de modo a descrever o desenvolvimento das corridas.

Este trabalho foi realizado usando a linguagem de programa de programação Java juntamente com o IDE NetBeans 8.2. Foi também criado um ficheiro “bash_script.sh” para fazer o deployment dos diversos ficheiros .jar para as máquinas correctas.

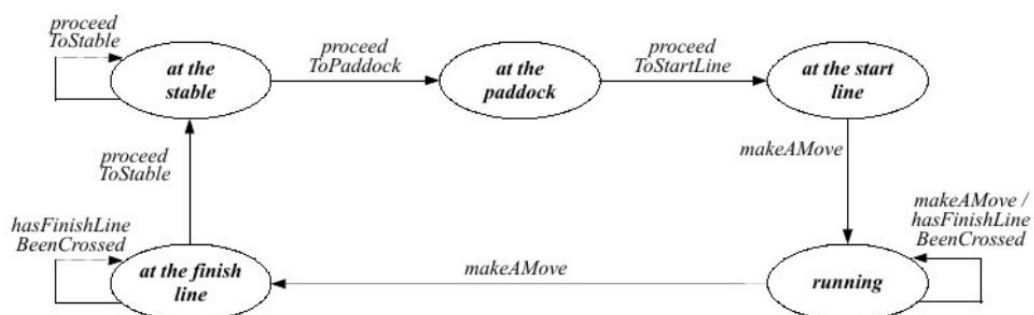
Diagramas de Estados

A figura abaixo consiste em evidenciar os estados possíveis das threads e os vários mecanismos que resultam numa passagem de um estado para o outro.

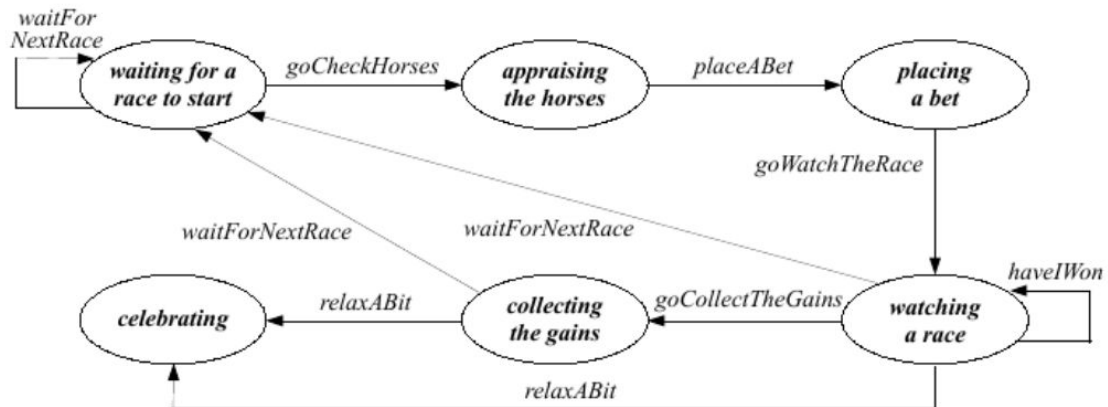
Broker life cycle



Horse / jockey life cycle

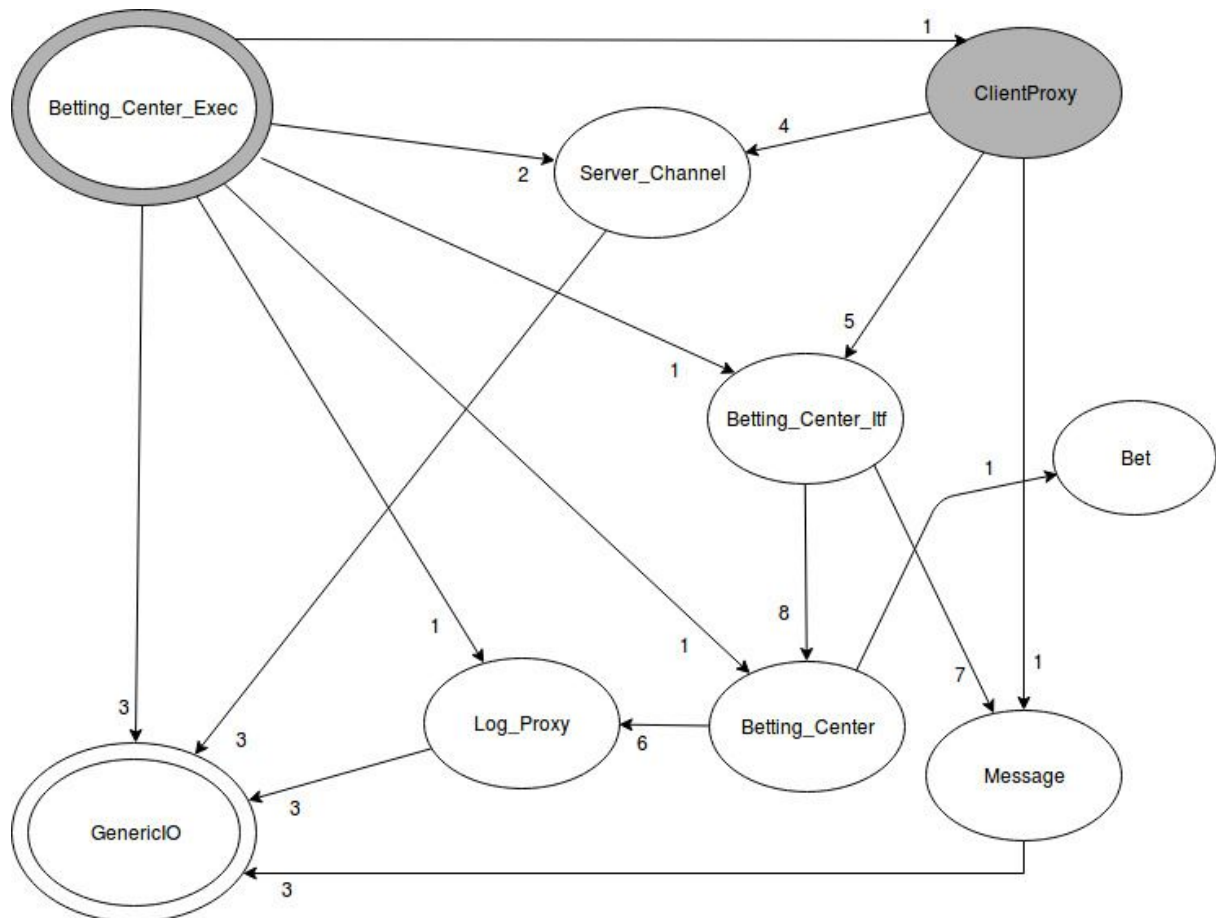


Spectator life cycle



Diagramas de Interação - Lado do Servidor.

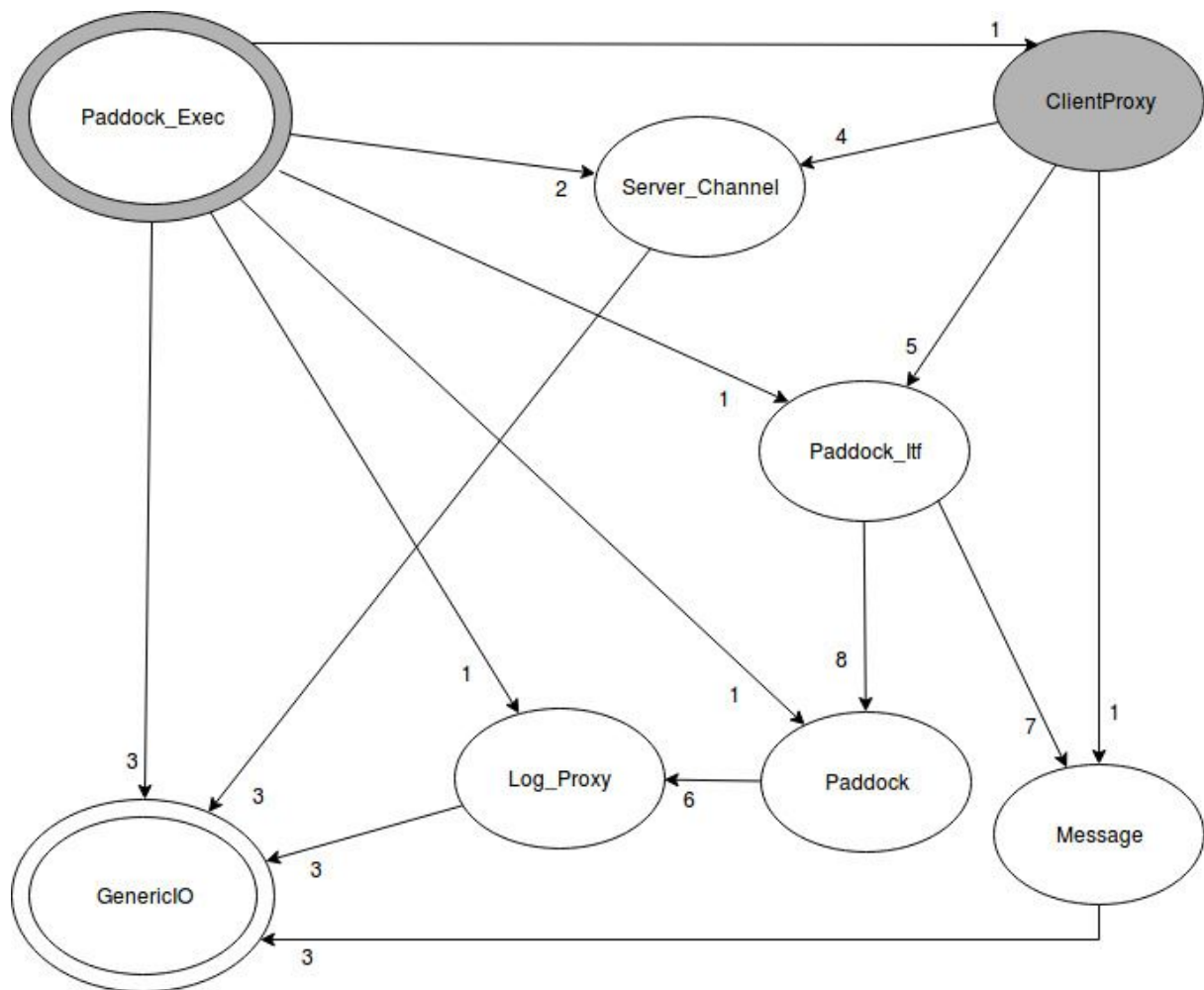
Diagrama do lado do Betting_Center



Legenda:

- 1 - Instantiate.
- 2 - Instantiate, start and commChannel.
- 3 - writelnString.
- 4 - read, write, close.
- 5 - processAndReply, serviceEnded..
- 6 - log.setBrokerState, log.setSpectatorState, log.setSpectatorHorseSel, log.setSpectatorMoneyToBet, log.setSpectatorMoney, log.changeLog.
- 7 - Instantiate, getType, getInfo, getId, getHorseID, getAddedInfo, getAddedInfo2.
- 8 - waitForBets, goWaitForHorses, settlingAccounts, horsesToPaddock, entertain, areThereAnyWinners, placeABet, goWatchRace, collectGains, goRelax, getRelaxCount, waitForNextRace.

Diagrama do lado do Paddock

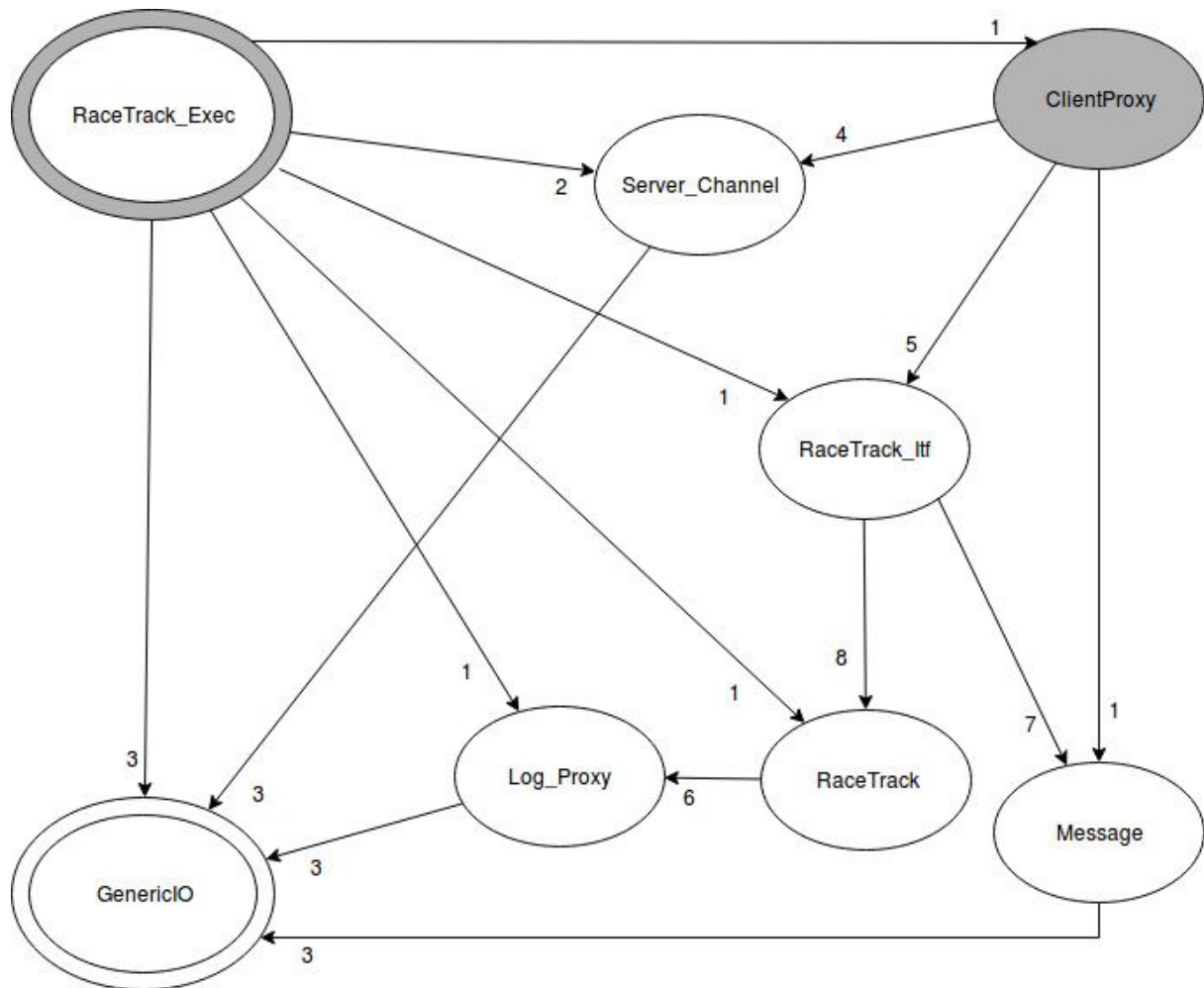


Legenda:

- 1 - Instantiate.
- 2 - Instantiate, start and commChannel.
- 3 - writelnString.
- 4 - read, write, close.
- 5 - processAndReply, serviceEnded..

- 6 - log.setBrokerState, log.setHorseState, log.setSpectatorState, log.setHorseOdds, log.changeLog.
- 7 - Instantiate, getType, getAddedInfo.
- 8 - announceNextRace, checkSpecCount, acceptTheBets, proceedToStartLine, proceedToPaddock, checkForSpectators, waitForNextRace, appraiseHorses, goPlaceABet.

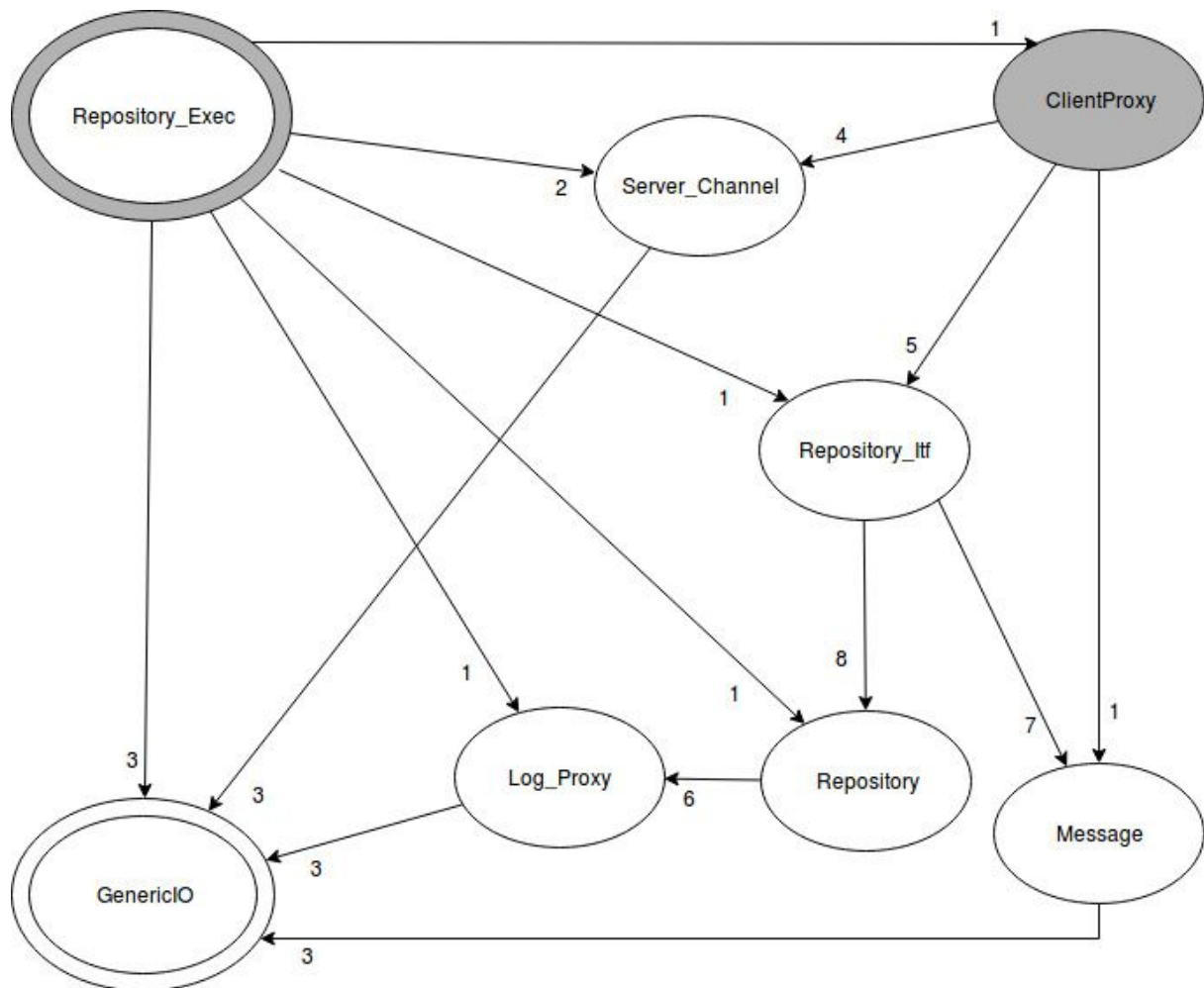
Diagrama do lado do RaceTrack



Legenda:

- 1 - Instantiate.
- 2 - Instantiate, start and commChannel.
- 3 - writelnString.
- 4 - read, write, close.
- 5 - processAndReply, serviceEnded..
- 6 - log.setBrokerState, log.setHorseState, log.setHorseIT, log.setHorseLength, log.setHorsePos, log.setHorsesAtTheEnd, log.changeLog.
- 7 - Instantiate, getType, getAddedInfo.
- 8 - waitForHorses, startRace, superviseTheRace, reportResults, checkOnHorses, atTheStartLine, makeAMove, hasFinishLineBeenCrossed, haveAllHorsesFinished, proceedToStable.

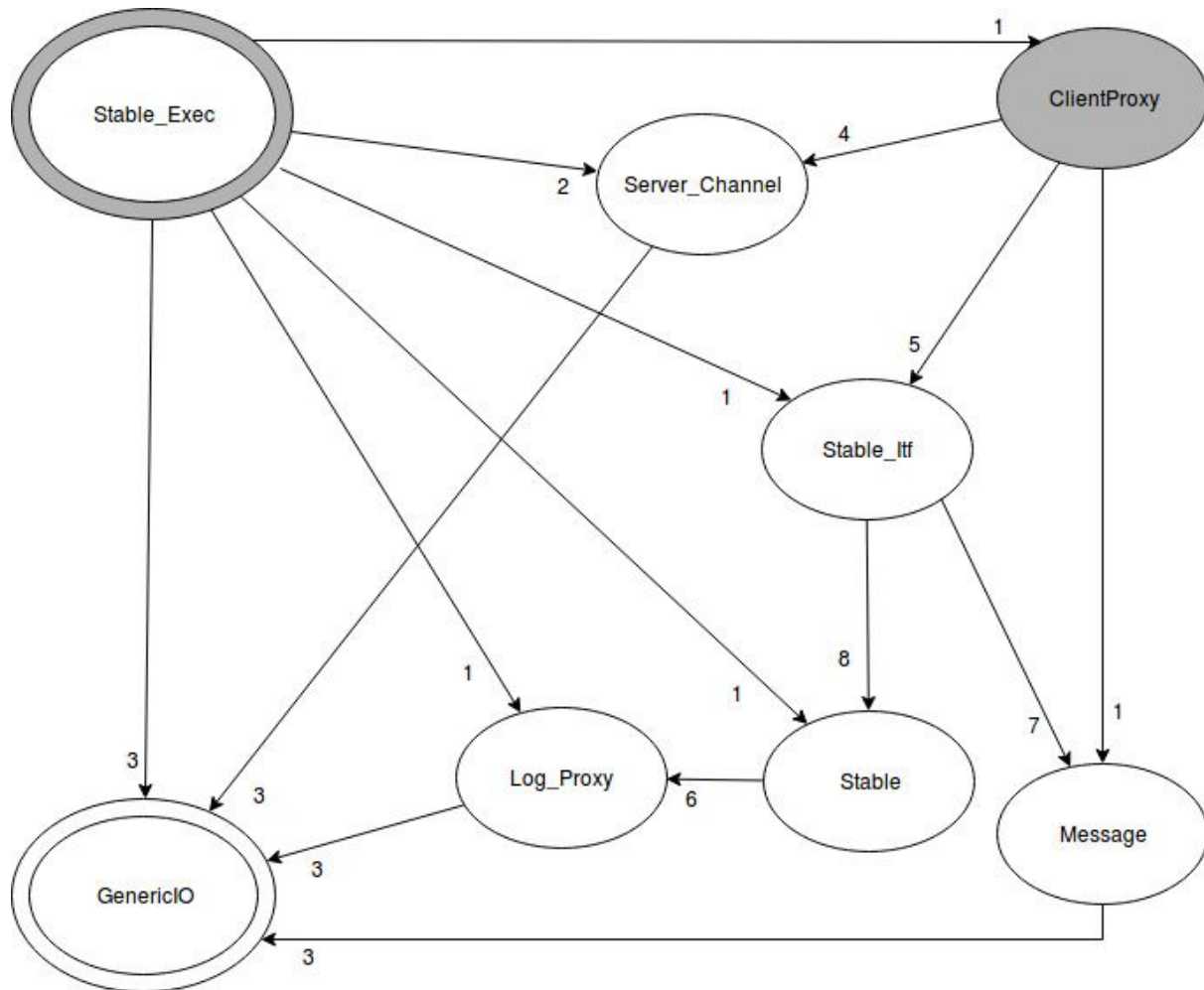
Diagrama do lado do Repository



Legenda:

- 1 - Instantiate.
- 2 - Instantiate, start and commChannel.
- 3 - writelnString.
- 4 - read, write, close.
- 5 - processAndReply, serviceEnded..
- 6 - log.setSpectatorState, log.changeLog.
- 7 - Instantiate, getType, getInfo, getAddedInfo.
- 8 - reportResults, alertWinners, checkWin, goCollectGains, checkFinalRace.

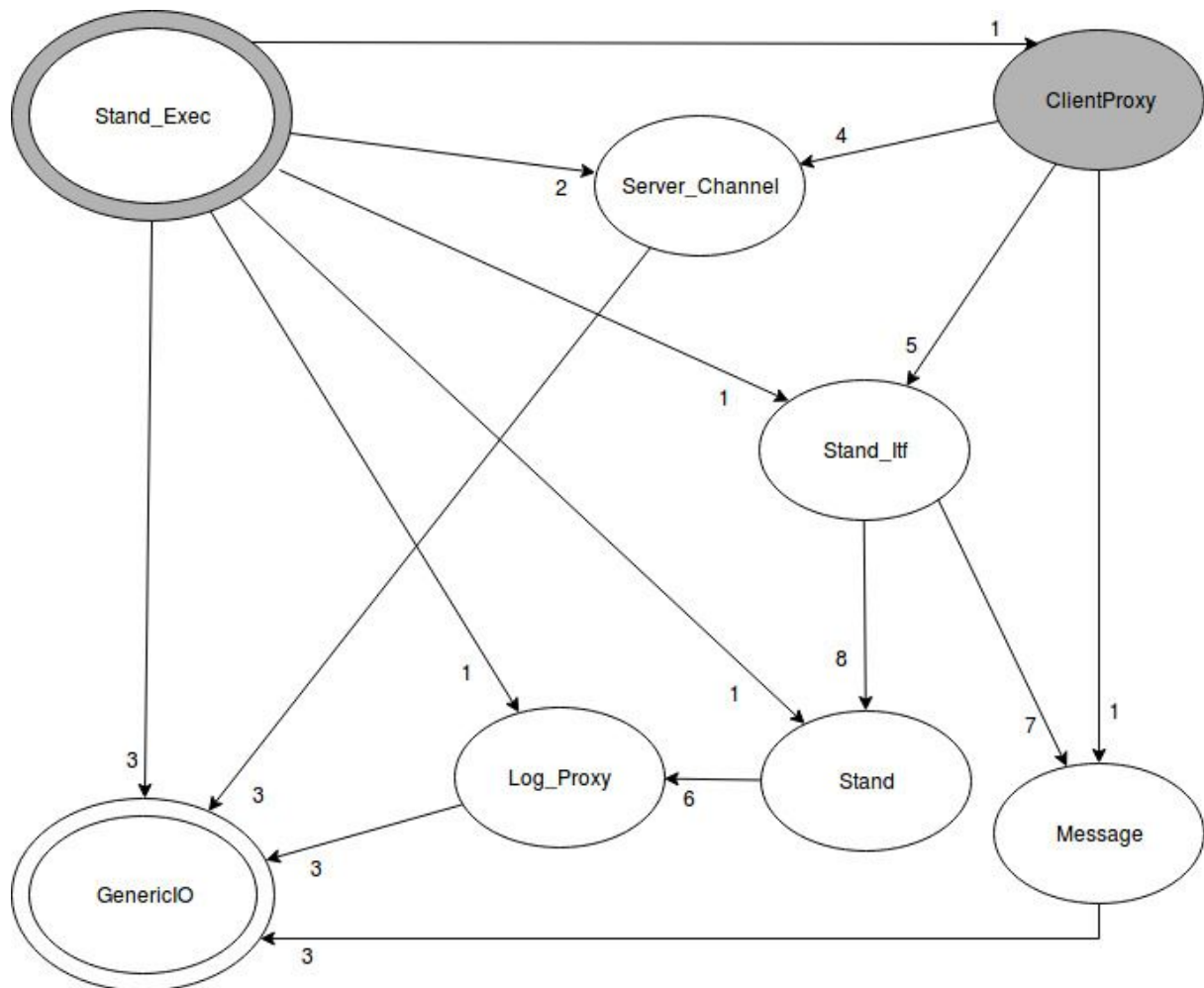
Diagrama do lado do Stable



Legenda:

- 1 - Instantiate.
- 2 - Instantiate, start and commChannel.
- 3 - writelnString.
- 4 - read, write, close.
- 5 - processAndReply, serviceEnded..
- 6 - log.printFirst, log.setBrokerState, log.setHorseState, log.changeLog.
- 7 - Instantiate, getType.
- 8 - openEvent, summonHorsesToPaddock, checkHorses, proceedToPaddock, returnToStable.

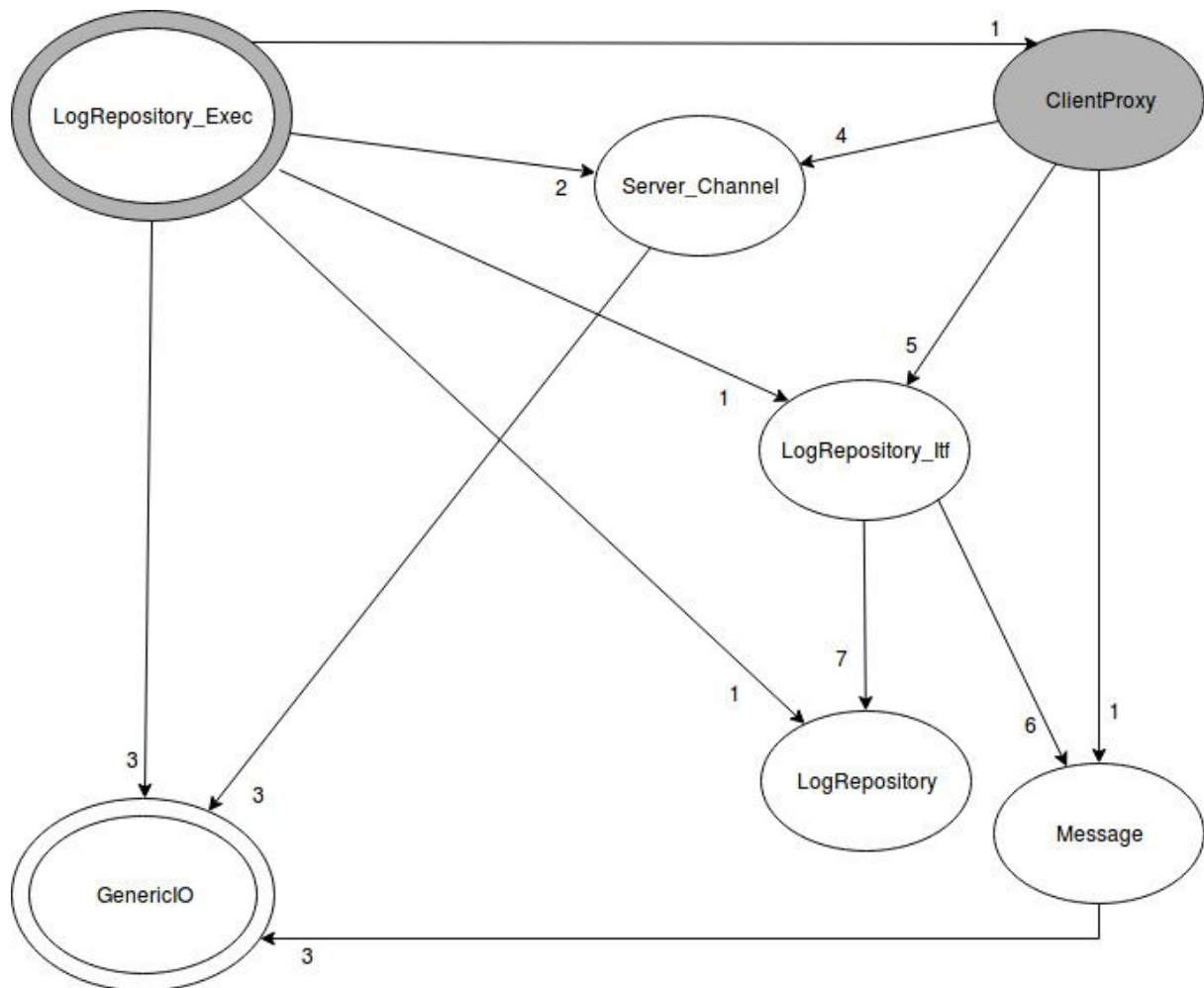
Diagrama do lado do Stand



Legenda:

- 1 - Instantiate.
- 2 - Instantiate, start and commChannel.
- 3 - writelnString.
- 4 - read, write, close.
- 5 - processAndReply, serviceEnded..
- 6 - log.setBrokerState, log.setSpectatorState, log.changeLog, log.finishLog.
- 7 - Instantiate, getType, getAddedInfo.
- 8 - goReportResults, entertain, honourBets, horsesToPaddock, checkSpectators, waitForNextRace, goCheckHorses, watchingTheRace, goCheckWin, celebrate, checkCelSpectators, requestFinishLog, theEnd.

Diagrama do lado do LogRepository

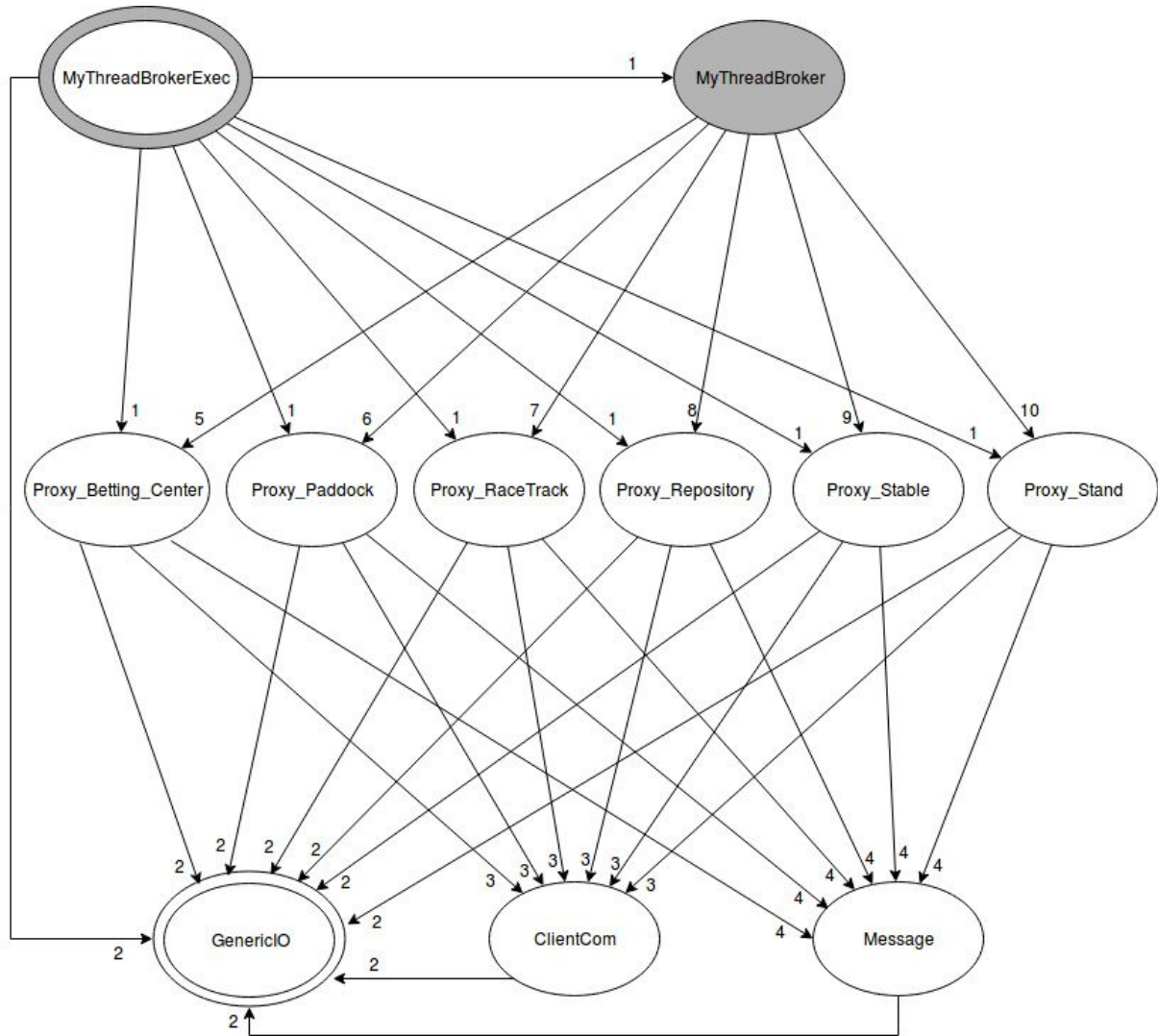


Legenda:

- 1 - Instantiate.
- 2 - Instantiate, start and commChannel.
- 3 - writelnString.
- 4 - read, write, close.
- 5 - processAndReply, serviceEnded.
- 6 - Instantiate, getType, getBroker_state, getId, getSpec_state, getHorse_state, getAddedInfo, getHorseID, getInfo.
- 7 - printFirst, changeLog, getRaceNumber, getTravelDistance, setBrokerState, setSpectatorState, setSpectatorMoney, setSpectatorMoneyToBet, setSpectatorHorseSel, setHorseState, setHorseLength, setHorseOdds, setHorseIT, setHorsePos, setHorsesAtTheEnd, setRaceNumber, finishLog.

Diagramas de Interação - Lado do Cliente.

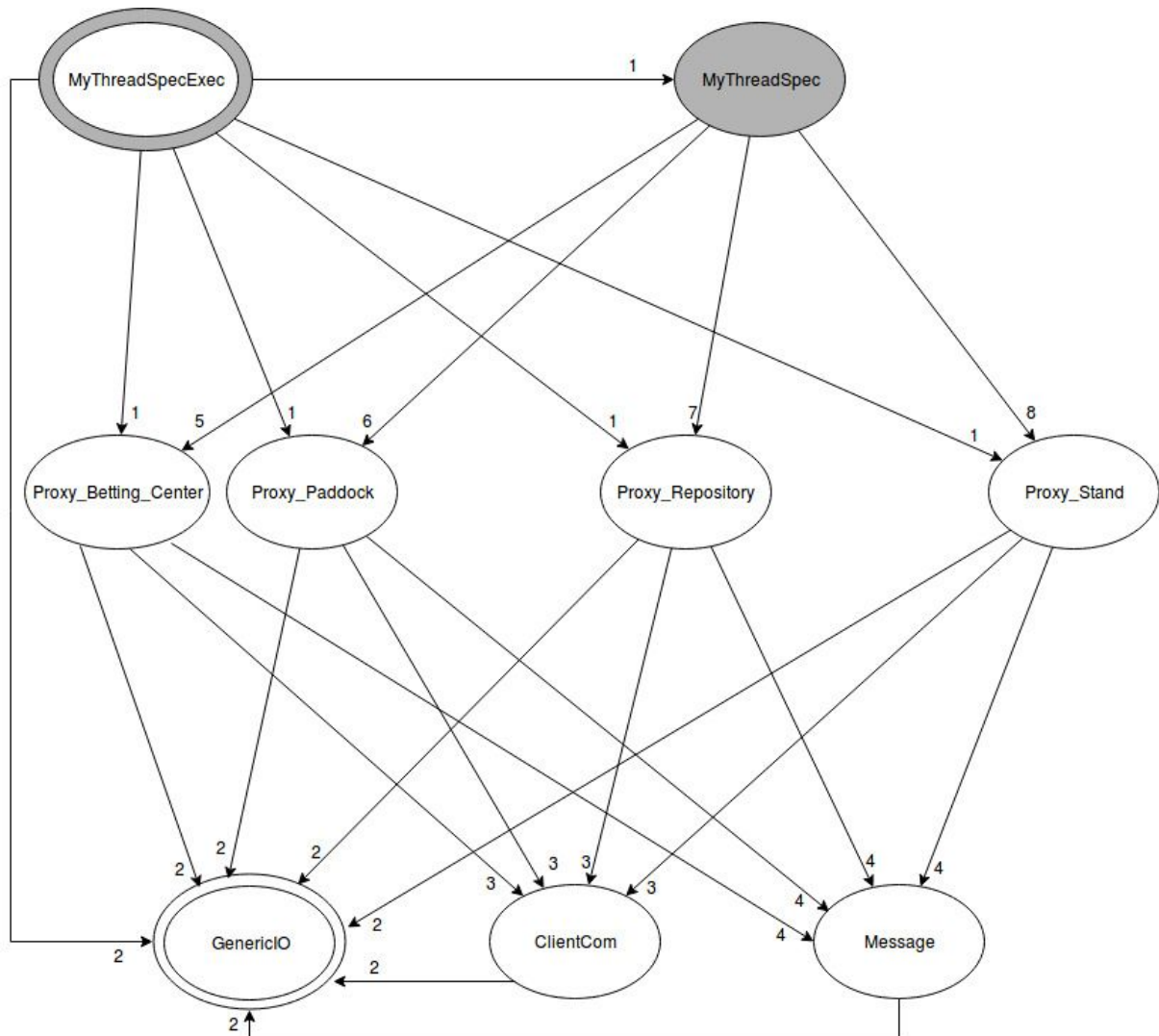
Diagrama do lado do Broker



Legenda:

- 1 - Instantiate.
- 2 - writelnString.
- 3 - Instantiate, open, write, read, close.
- 4 - Instantiate, getType, toString.
- 5 - waitForBets, goWaitForHorses, areThereAnyWinners, horsesToPaddock, settlingAccounts, entertain.
- 6 - checkSpecCount, acceptTheBets
- 7 - waitForHorses, startRace, superviseTheRace, reportResults.
- 8 - reportResults, checkFinalRace.
- 9 - openEvent, checkHorses, summonHorsesToPaddock.
- 10 - checkSpectators, goReportResults, honourBets, entertain, checkSpectators, checkCelSpectators, getRelaxCount, requestFinishLog, theEnd.

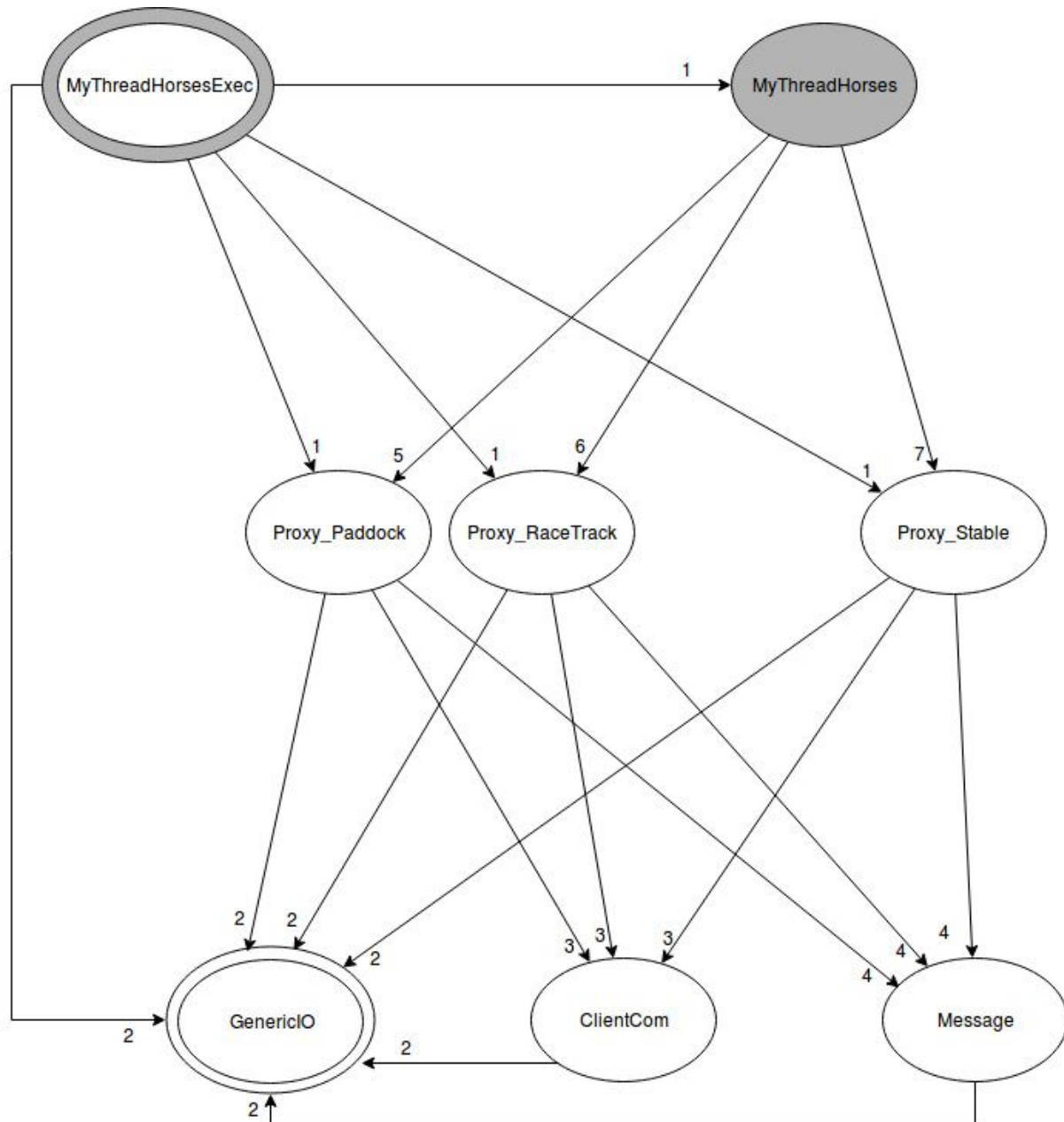
Diagrama do lado do Spectators



Legenda:

- 1 - Instantiate.
- 2 - writeInString.
- 3 - Instantiate, open, write, read, close.
- 4 - Instantiate, getType, toString.
- 5 - placeABet, goWatchRace, collectGains, goRelax, waitForNextRace.
- 6 - waitForNextRace, appraiseHorses, goPlaceABet.
- 7 - checkWin, checkFinalRace.
- 8 - goCheckHorses, watchingTheRace, goCheckWin, celebratewaitForNextRace.

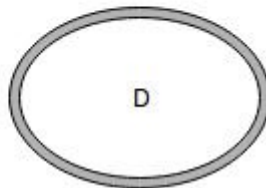
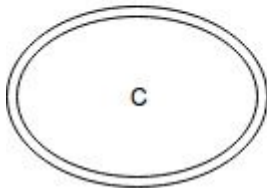
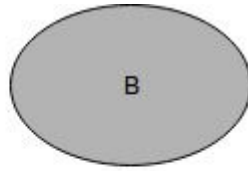
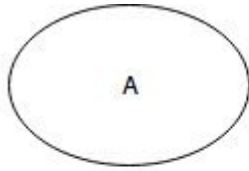
Diagrama do lado do Horses



Legenda:

- 1 - Instantiate.
- 2 - writeInString.
- 3 - Instantiate, open, write, read, close.
- 4 - Instantiate, getType, toString.
- 5 - checkForSpectators, proceedToPaddock, proceedToStartLine.
- 6 - atTheStartLine, hasFinishLineBeenCrossed, makeAMove, proceedToStable.
- 7 - returnToStable, proceedToPaddock.

Legenda Geral:



- **A** - Class Java normal.
- **B** - Thread.
- **C** - Biblioteca utilizada (não pertecente à oracle).
- **D** - Executável.