

Relatório Trabalho Prático III

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 3 “An Afternoon at the races”.

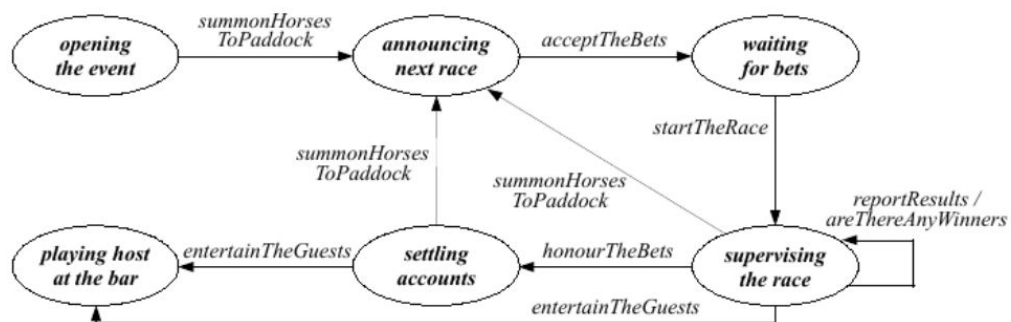
Este trabalho consiste em passar a solução multithreading de uma corrida de cavalos obtida no assignment 1 numa solução distribuída para correr em várias máquinas, onde ocorre comunicação entre diversas instâncias de threads diferentes e monitores de modo a descrever o desenvolvimento das corridas, desta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

Este trabalho foi realizado usando a linguagem de programa de programação Java juntamente com o IDE NetBeans 8.2. Foi também criado um ficheiro .sh para correr os diversos ficheiros nas máquinas correctas.

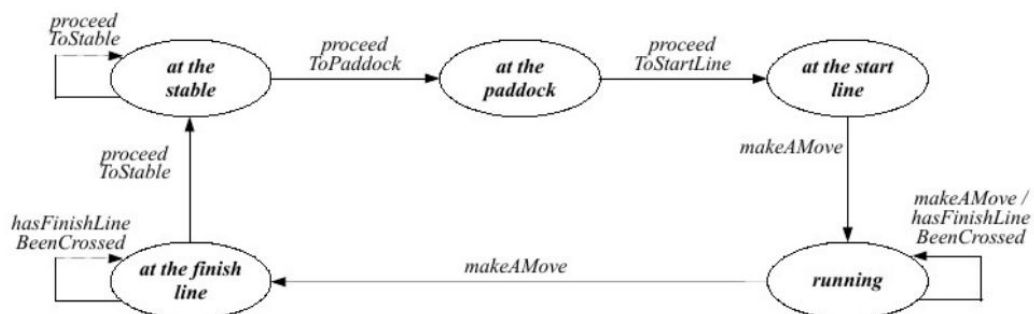
Diagramas de Estados

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.

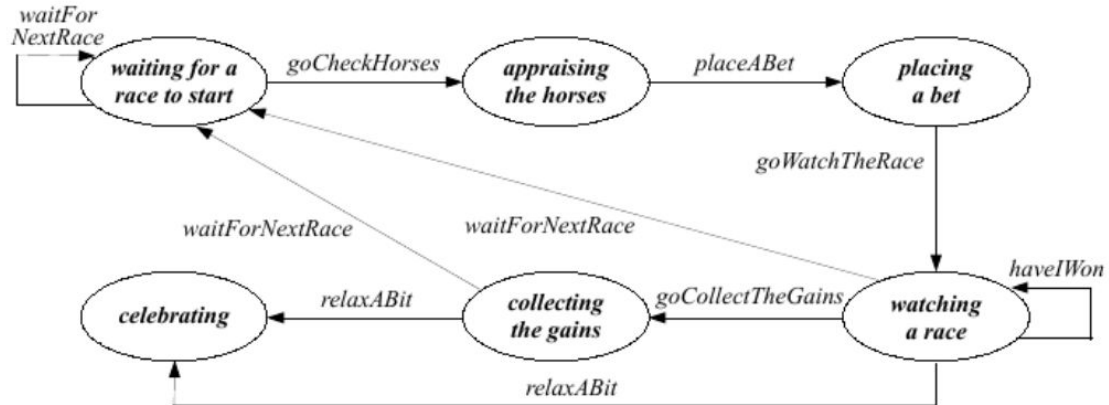
Broker life cycle



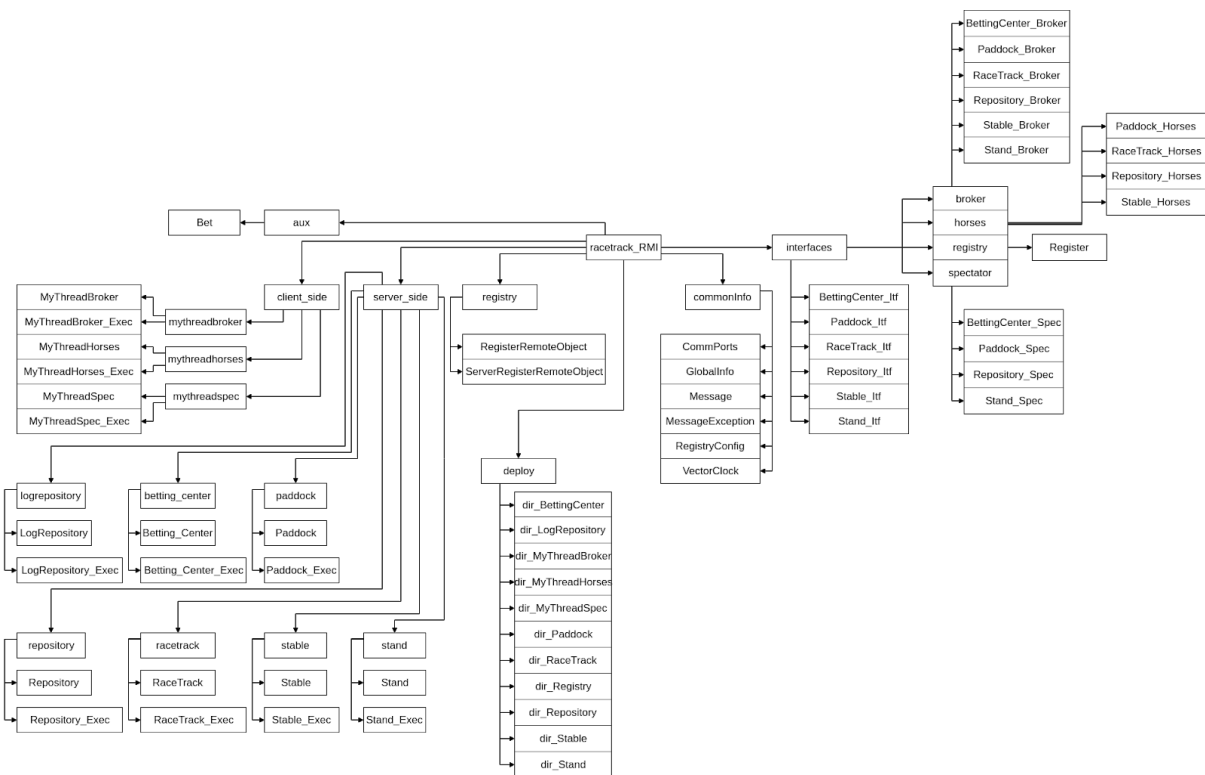
Horse / jockey life cycle



Spectator life cycle

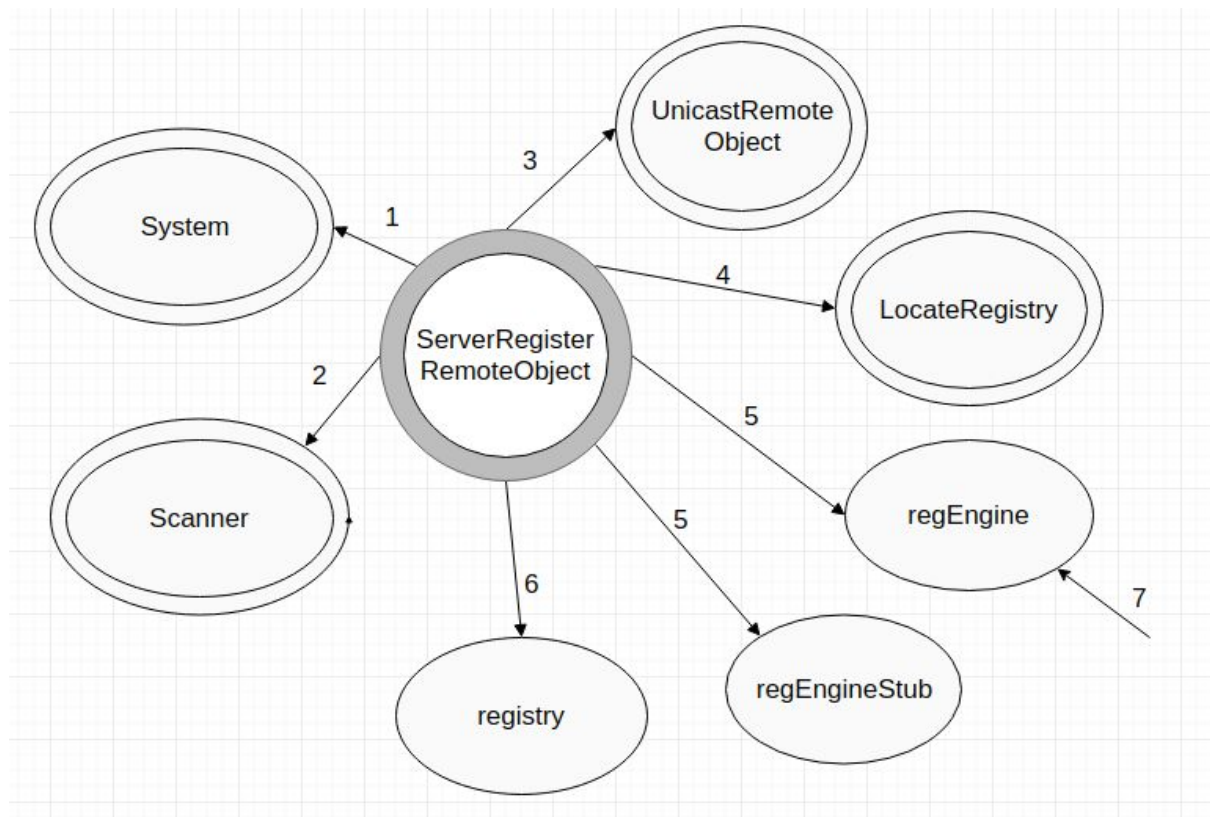


Organização do pacote racetrack_RMI



Diagramas de Interação - Lado do Servidor.

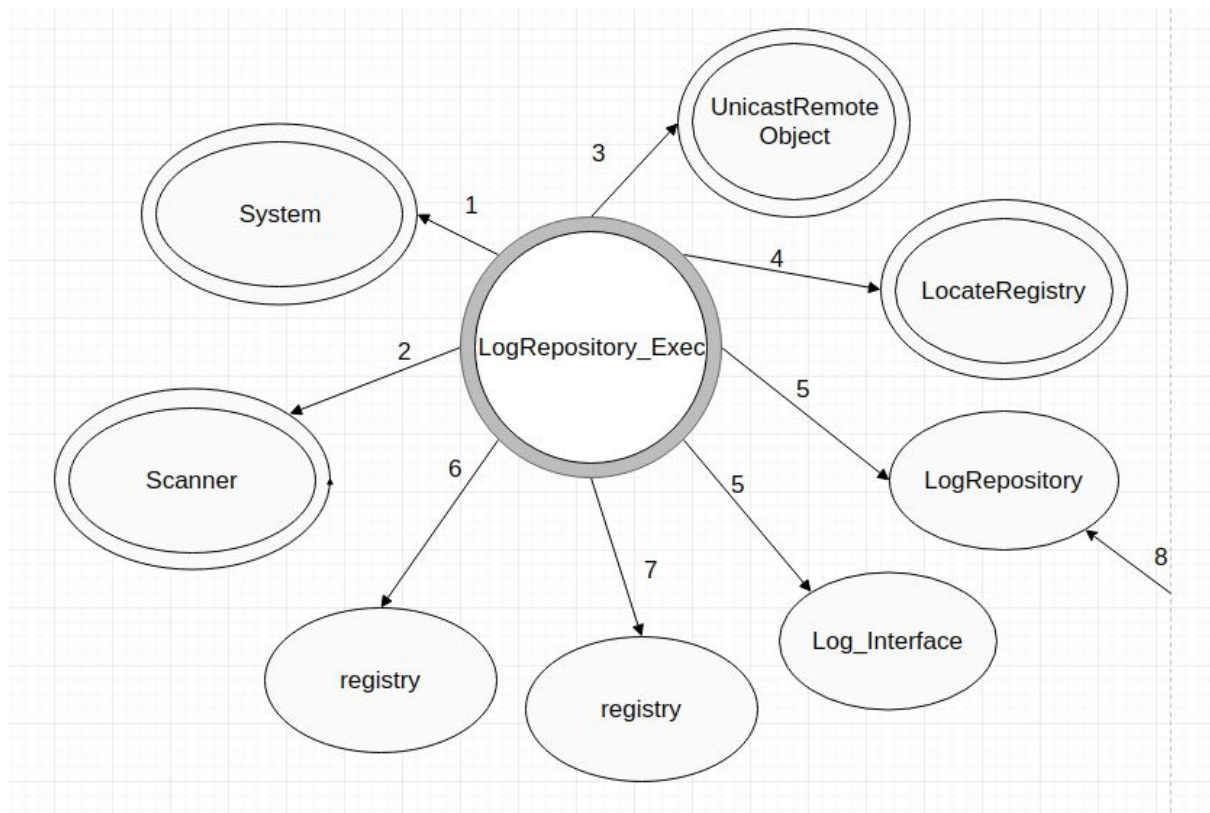
Diagrama do lado do ServerRegisterRemoteObject



Legenda:

- 1 - getSecurityManager, setSecurityManager.
- 2 - nextLine, nextInt.
- 3 - exportObject.
- 4 - LocateRegistry.
- 5 - instantiate.
- 6 - instantiate, rebind.
- 7 - bind, unbind, rebind.

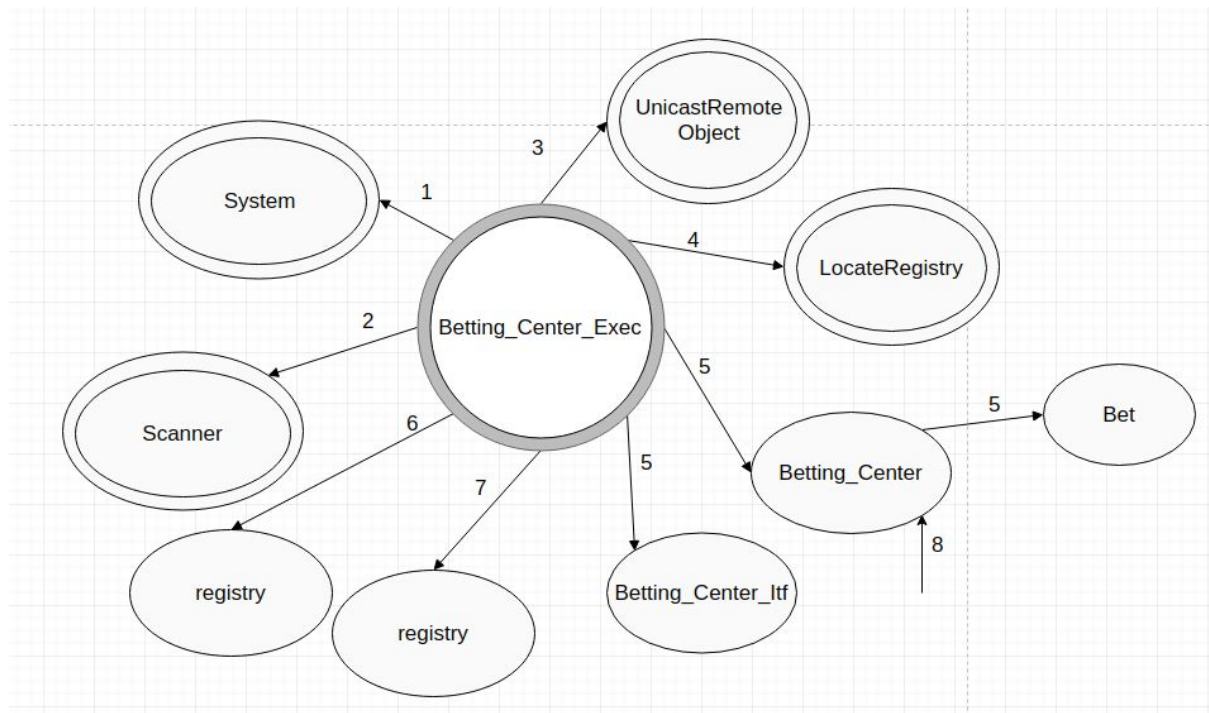
Diagrama do lado do LogRepository



Legenda:

- 1 - getSecurityManager, setSecurityManager.
- 2 - nextLine, nextInt.
- 3 - exportObject.
- 4 - LocateRegistry.
- 5 - instantiate.
- 6 - instantiate, locate.
- 7 - instantiate, bind.
- 8 - printFirst, changeLog, getRaceNumber, getTravelDistance, setBrokerState, setSpectatorState, setSpectatorMoney, setSpectatorMoneyToBet, setSpectatorHorseSel, setHorseState, setHorseLength, setHorseOdds, setHorseIT, setHorsePos, setHorsesAtTheEnd, setRaceNumber, finishLog, terminate.

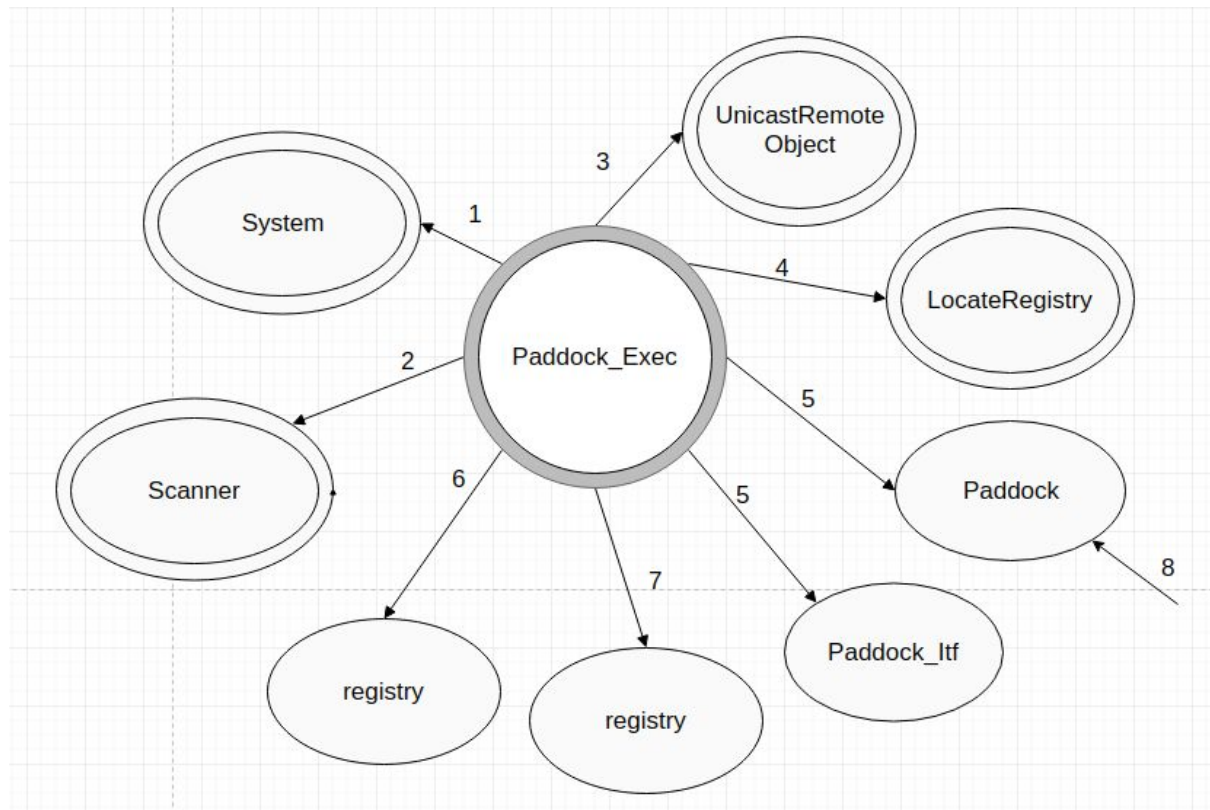
Diagrama do lado do BettingCenter



Legenda:

- 1 - getSecurityManager, setSecurityManager, println.
- 2 - nextLine, nextInt.
- 3 - exportObject.
- 4 - LocateRegistry.
- 5 - instantiate.
- 6 - instantiate, locate.
- 7 - instantiate, bind.
- 8 - waitForBets, goWaitForHorses, settlingAccounts, horsesToPaddock, entertain, areThereAnyWinners, placeABet, goWatchRace, collectGains, goRelax, getRelaxCount, waitForNextRace, terminate.

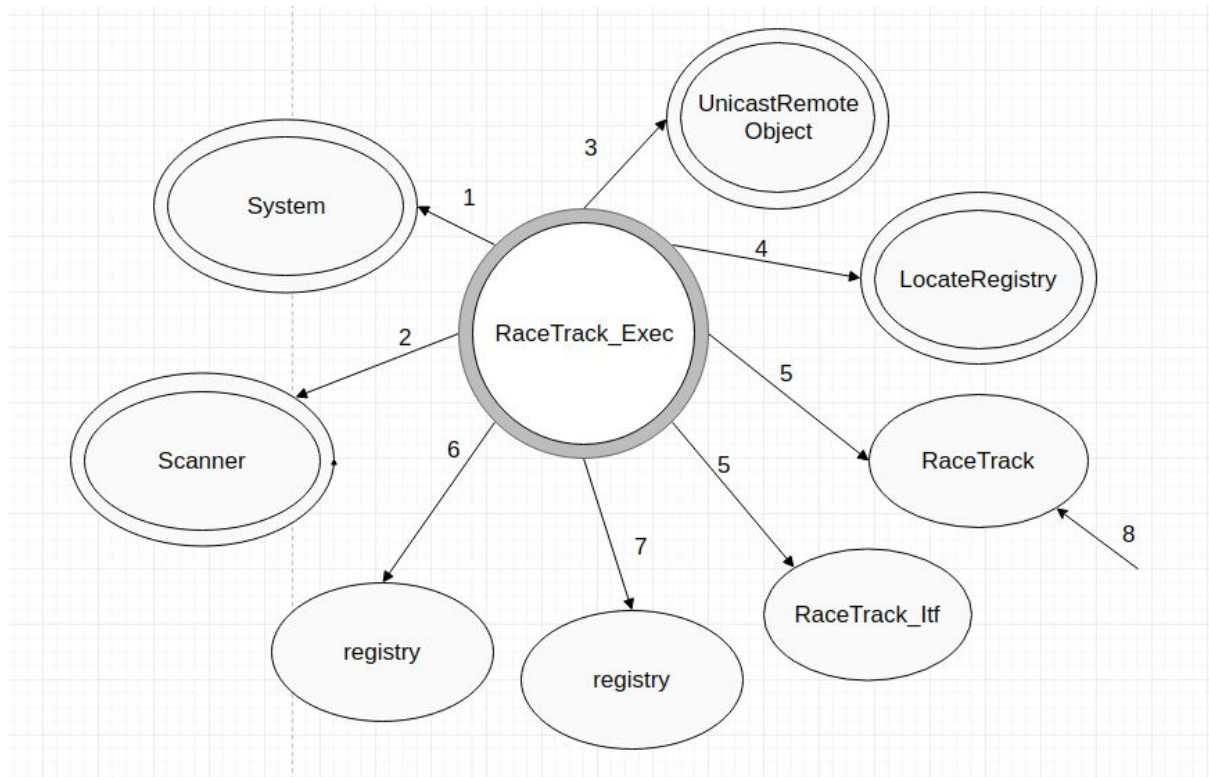
Diagrama do lado do Paddock



Legenda:

- 1 - getSecurityManager, setSecurityManager, println.
- 2 - nextLine, nextInt.
- 3 - exportObject.
- 4 - LocateRegistry.
- 5 - instantiate.
- 6 - instantiate, locate.
- 7 - instantiate, bind.
- 8 - announceNextRace, checkSpecCount, acceptTheBets, proceedToStartLine, proceedToPaddock, checkForSpectators, waitForNextRace, appraiseHorses, goPlaceABet, terminate.

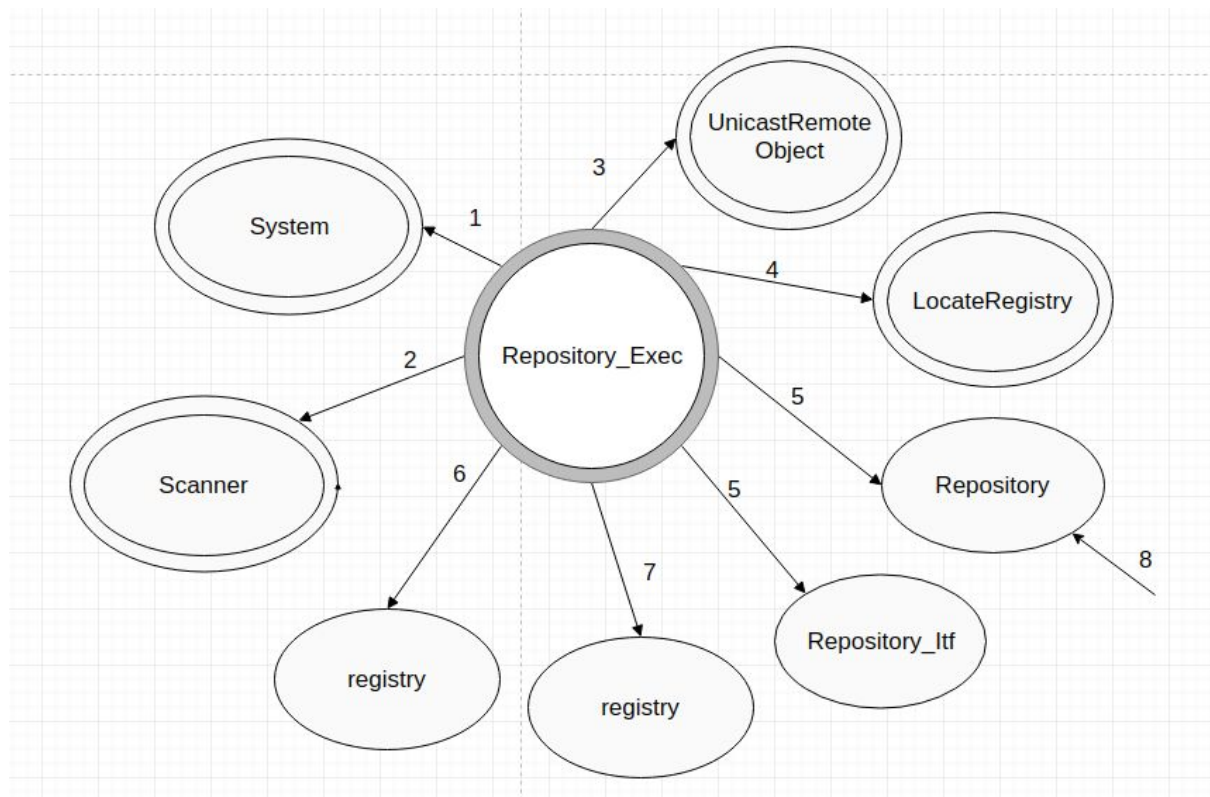
Diagrama do lado do RaceTrack



Legenda:

- 1 - getSecurityManager, setSecurityManager, println.
- 2 - nextLine, nextInt.
- 3 - exportObject.
- 4 - LocateRegistry.
- 5 - instantiate.
- 6 - instantiate, locate.
- 7 - instantiate, bind.
- 8 - waitForHorses, startRace, superviseTheRace, reportResults, checkOnHorses, atTheStartLine, makeAMove, hasFinishLineBeenCrossed, haveAllHorsesFinished, proceedToStable, terminate.

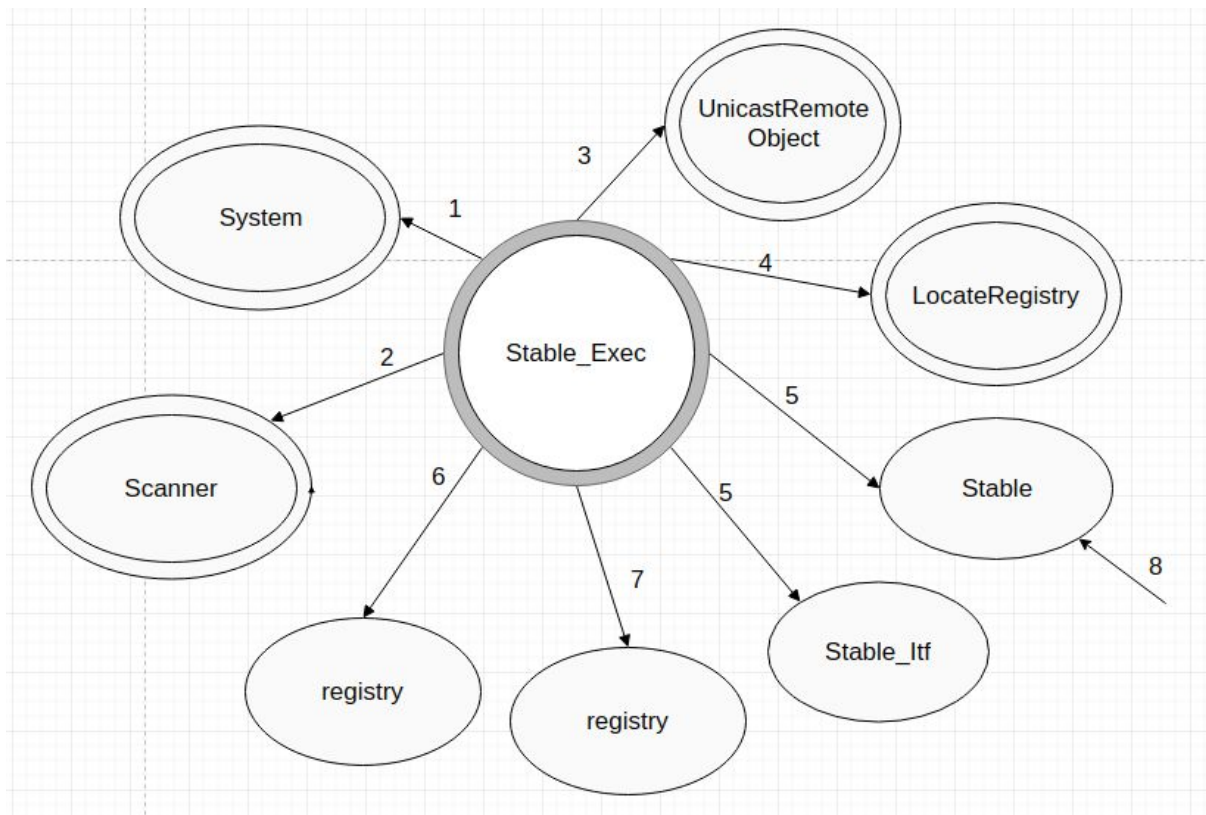
Diagrama do lado do Repository



Legenda:

- 1 - getSecurityManager, setSecurityManager, println.
- 2 - nextLine, nextInt.
- 3 - exportObject.
- 4 - LocateRegistry.
- 5 - instantiate.
- 6 - instantiate, locate.
- 7 - instantiate, bind.
- 8 - reportResults, alertWinners, checkWin, goCollectGains, checkFinalRace, terminate.

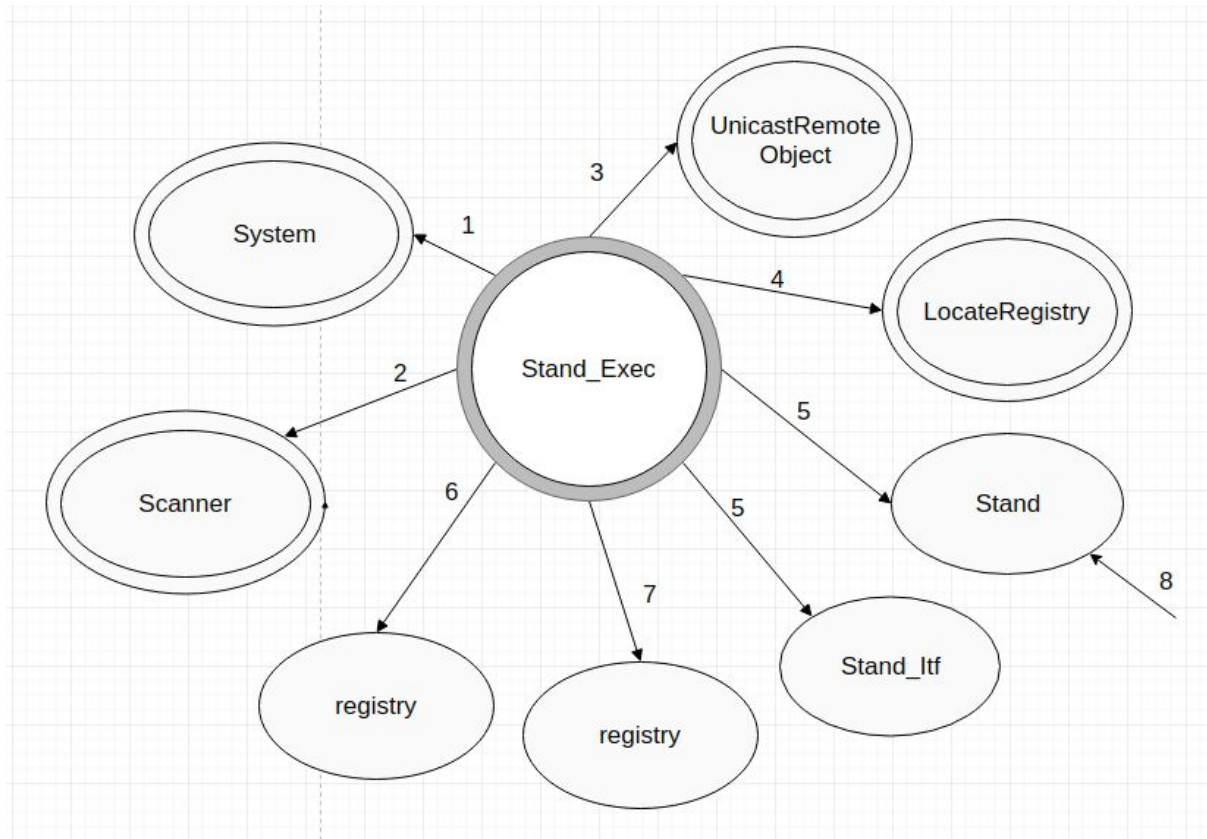
Diagrama do lado do Stable



Legenda:

- 1 - getSecurityManager, setSecurityManager, println.
- 2 - nextLine, nextInt.
- 3 - exportObject.
- 4 - LocateRegistry.
- 5 - instantiate.
- 6 - instantiate, locate.
- 7 - instantiate, bind.
- 8 - openEvent, summonHorsesToPaddock, SendHorsesAway, checkHorses, checkHorsesAway, proceedToPaddock, returnToStable, leaveStableFinal, terminate.

Diagrama do lado do Stand

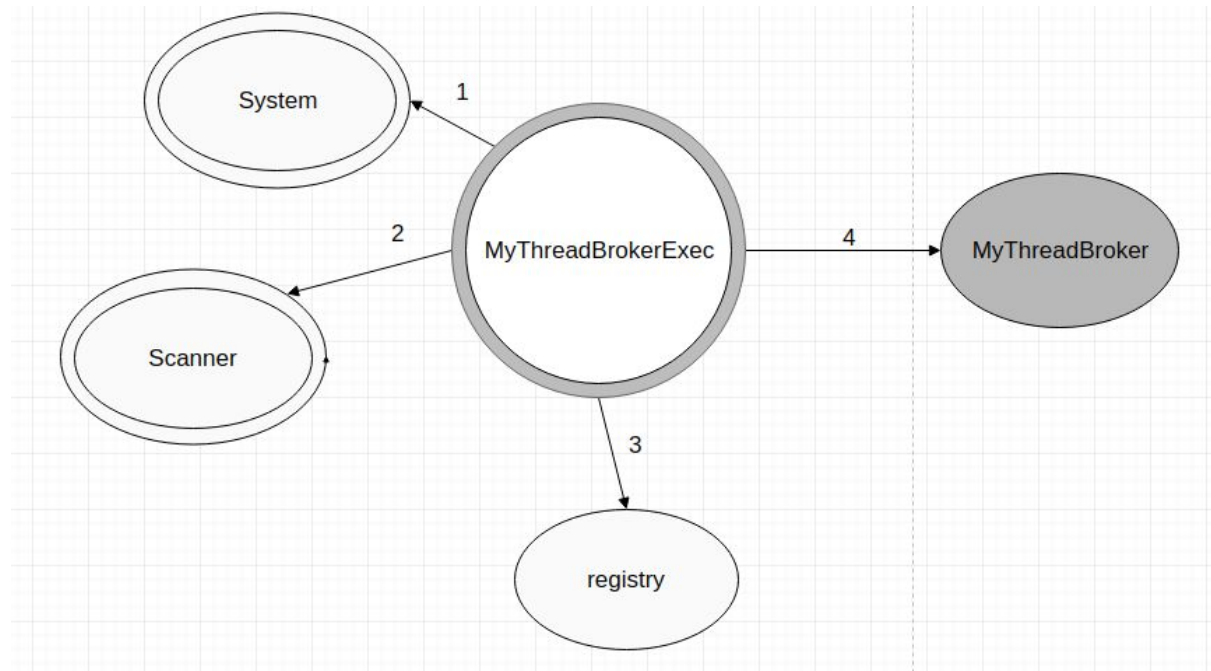


Legenda:

- 1 - getSecurityManager, setSecurityManager, println.
- 2 - nextLine, nextInt.
- 3 - exportObject.
- 4 - LocateRegistry.
- 5 - instantiate.
- 6 - instantiate, locate.
- 7 - instantiate, bind.
- 8 - goReportResults, entertain, honourBets, horsesToPaddock, checkSpectators, waitForNextRace, goCheckHorses, watchingTheRace, goCheckWin, celebrate, checkCelSpectators, requestFinishLog, theEnd, terminate.

Diagramas de Interação - Lado do Cliente.

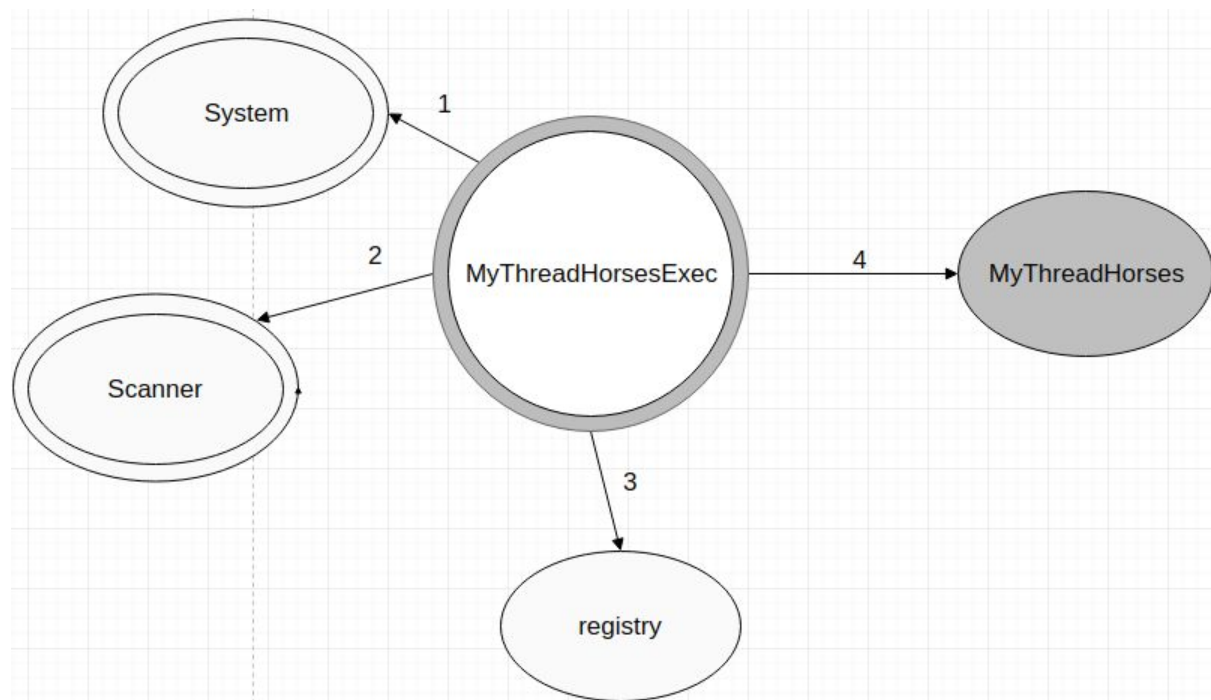
Diagrama do lado do Broker



Legenda:

- 1 - getSecurityManager, setSecurityManager, println.
- 2 - nextLine, nextInt.
- 3 - instantiate, lookup
- 4 - instantiate, start, join, terminate.

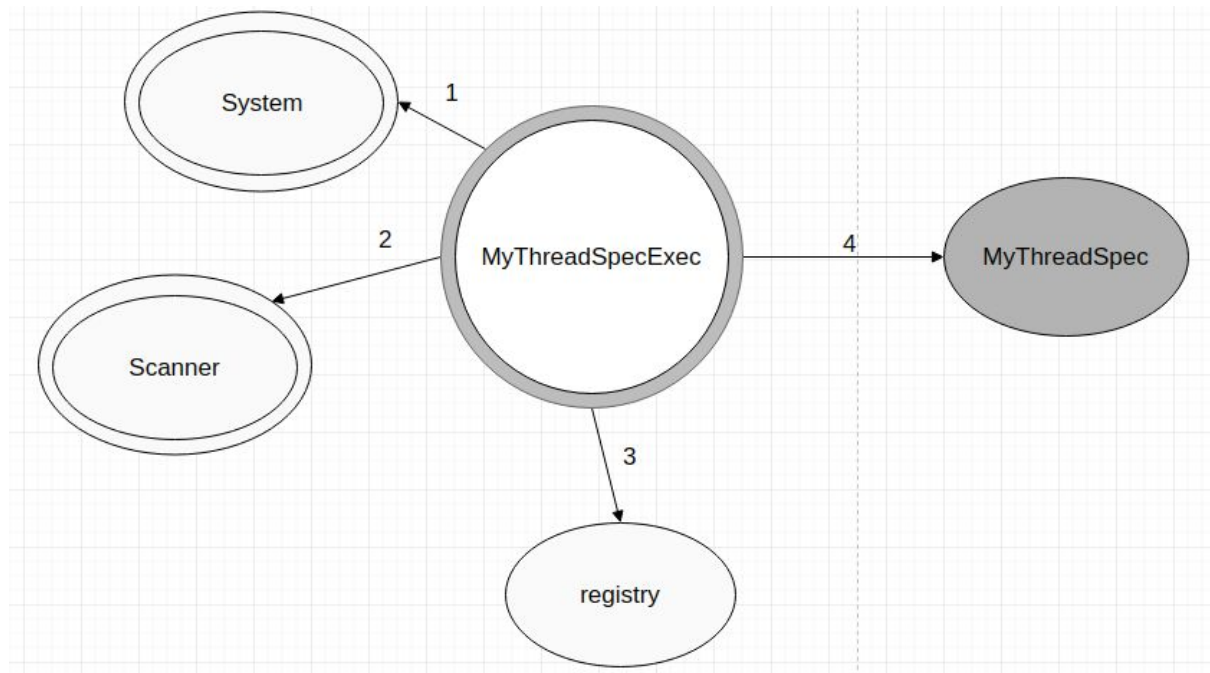
Diagrama do lado do Horses



Legenda:

- 1 - getSecurityManager, setSecurityManager, println.
- 2 - nextLine, nextInt.
- 3 - instantiate, lookup
- 4 - instantiate, start, join.

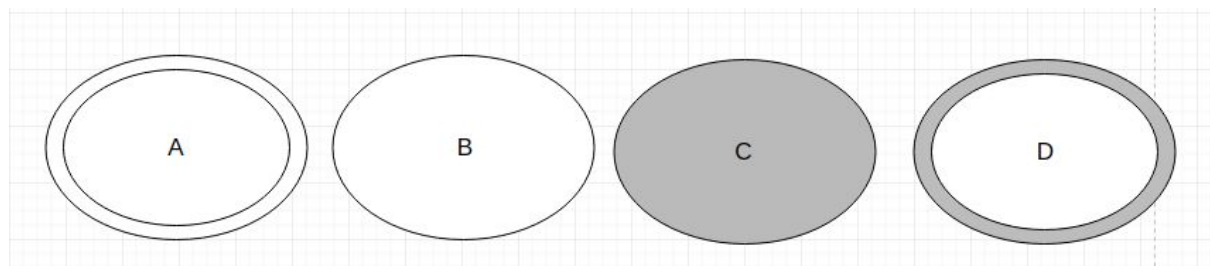
Diagrama do lado do Spectator



Legenda:

- 1 - getSecurityManager, setSecurityManager, println.
- 2 - nextLine, nextInt.
- 3 - instantiate, lookup
- 4 - instantiate, start, join.

Diagramas de Interação - Legenda Geral.



Legenda:

- **A** - Biblioteca utilizada (não pertence à Oracle).
- **B** - Classe java normal.
- **C** - Thread.
- **D** - Executável.