



# Networking em Unity utilizando Mirror

Ministrante: Arthur Rosa de Souza

2

**Ministrante**

# 3



- ▶ Aluno de Engenharia de Computação
- ▶ Ingresso em 2015
- ▶ Membro do Fellowship of the Game
- ▶ 4 anos de desenvolvimento de jogos
- ▶ Mais de 10 jogos desenvolvidos
  - ▶ 3 Networked
- ▶ LinkedIn:
- ▶ Portifolio: <https://arsart.itch.io/>

4

# Motivação

# 5

Por que  
multiplayer?

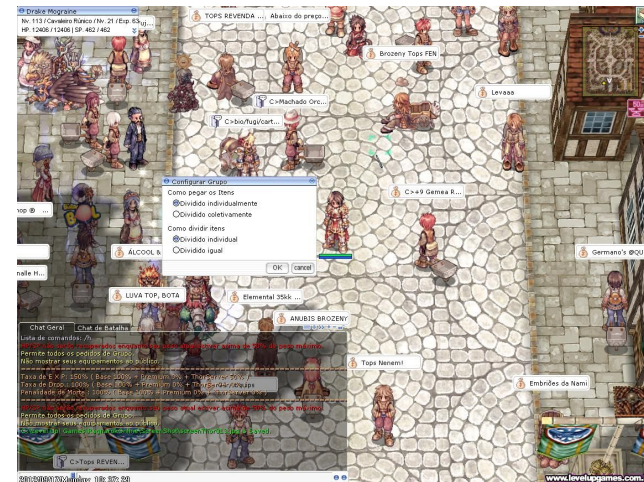
- Acrescenta fator de replay



# 6

Por que  
multiplayer?

## ► Acrescenta conteúdo ao jogo



Fonte: <https://www.techtudo.com.br/dicas-e-tutoriais/noticia/2012/09/como-criar-grupo-em-ragnarok.html>

# 7

Por que  
Mirror?

- ▶ Unity é uma das engines mais populares e com uma comunidade muito grande



# 8

## Por que Mirror?

### Unity Multiplayer ("UNet")

HLAPI, Critical Support

LLAPI, Critical Support

P2P Relay Service

Legacy Matchmaker Service

### New Connected Games Solution

New Network Transport Layer and Server Runtime

Game Server Hosting Service from Multiplay

New Integrated Matchmaker Service from Multiplay

Game Communication Service from Vivox

2017

2018

2019

2020

2021

2022

ALPHA

ALPHA

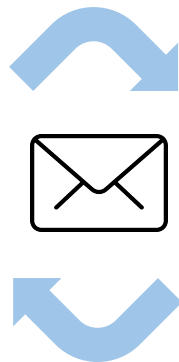


9

# Introdução

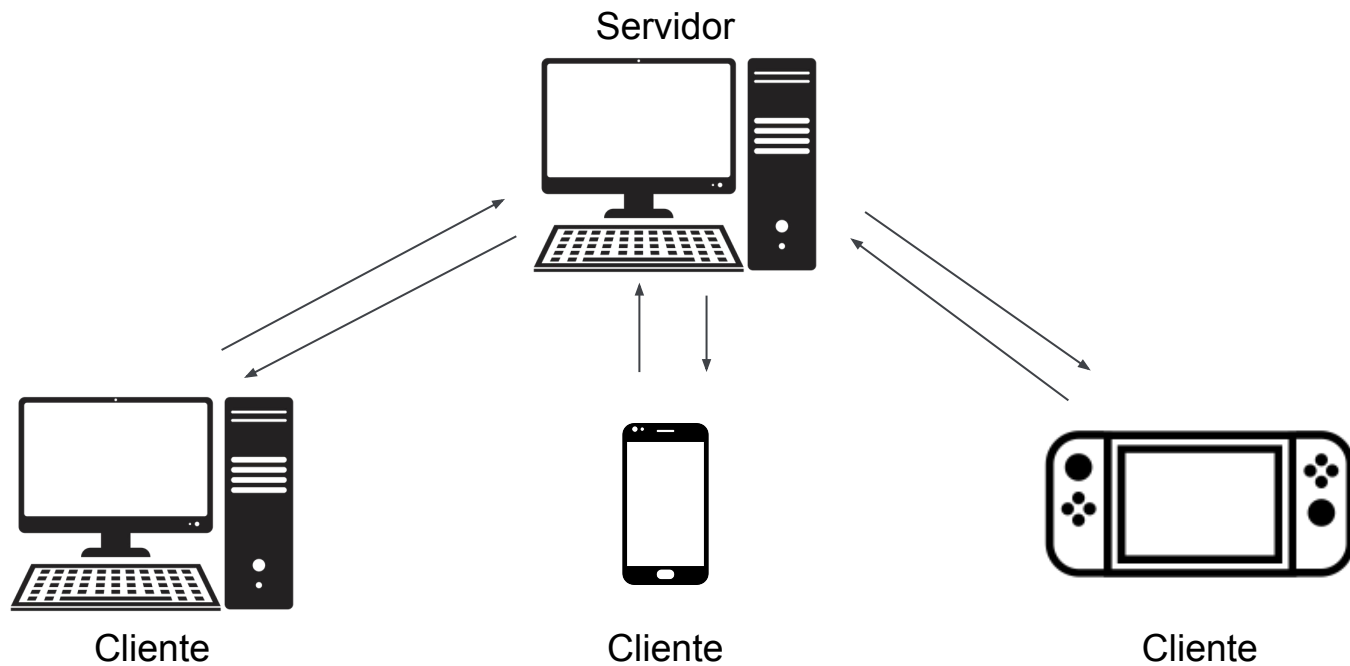
# 10

Como  
funciona  
multiplayer?



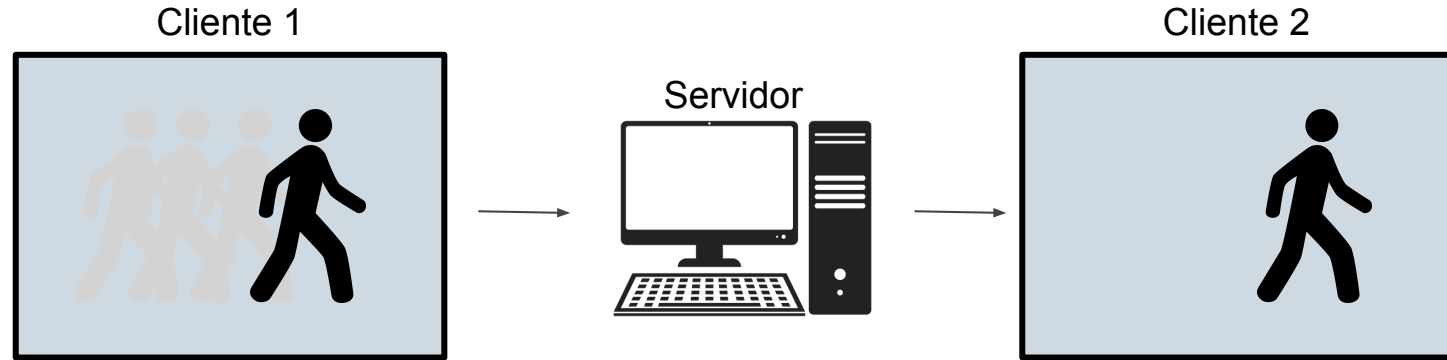
# 11

## Rede de computadores



# 12

## Exemplo



# 13

Mirror



Mirror

- ▶ Asset gratuito para Networking em Unity
- ▶ Replicar e melhorar antiga arquitetura de multiplayer da Unity: High Level API (HLAPI)
- ▶ Compatibilidade com multiplas versões da Unity
- ▶ Customizável
- ▶ Comunidade ativa



# Conceitos

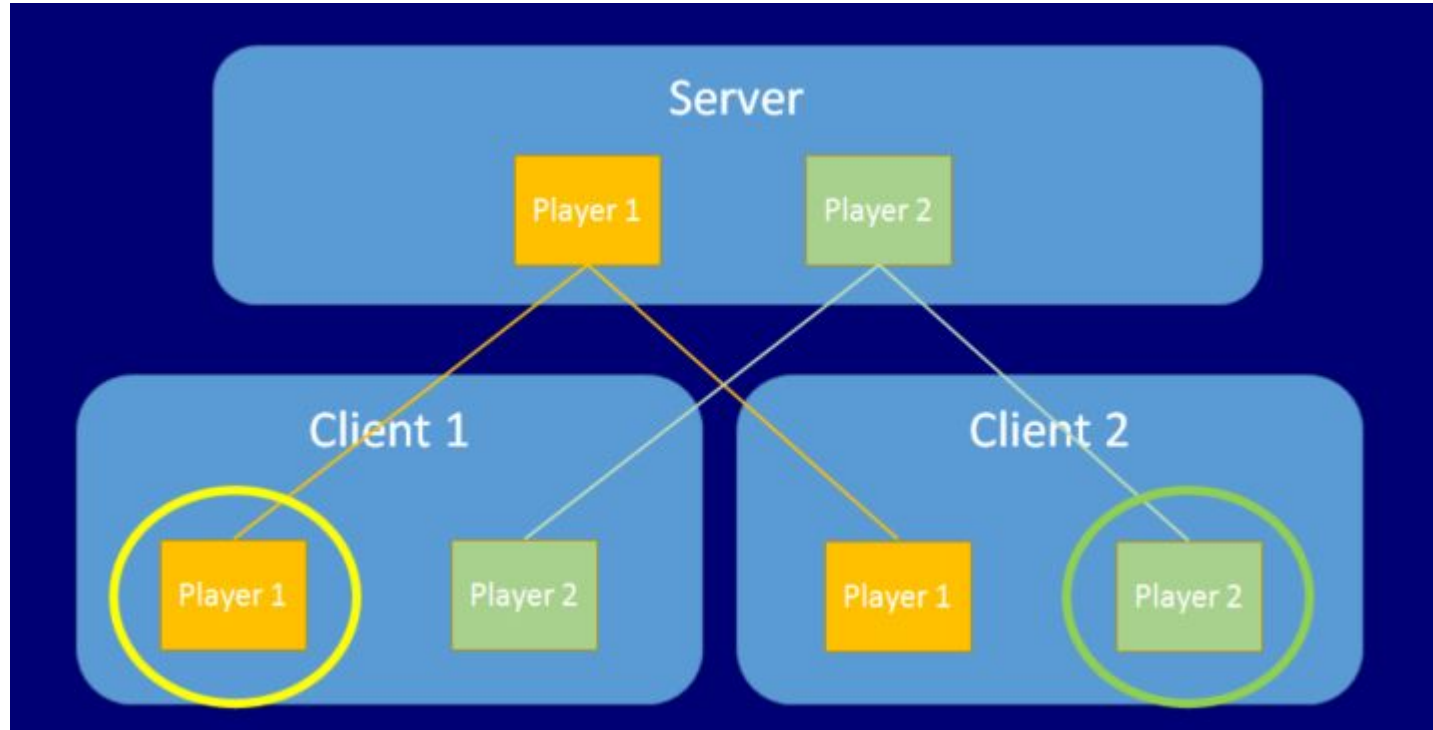
# 15

## Conceitos básicos

- ▶ Mesmo script para servidor e cliente (Mirror)
- ▶ Abstração de código (High level)
- ▶ Sincronismo por variáveis sincronizadas ou chamadas remotas
- ▶ Objetos são duplicados em todas as instâncias de jogo
- ▶ Uso de autoridade
- ▶ Tipos e componentes “Networked”

# 16

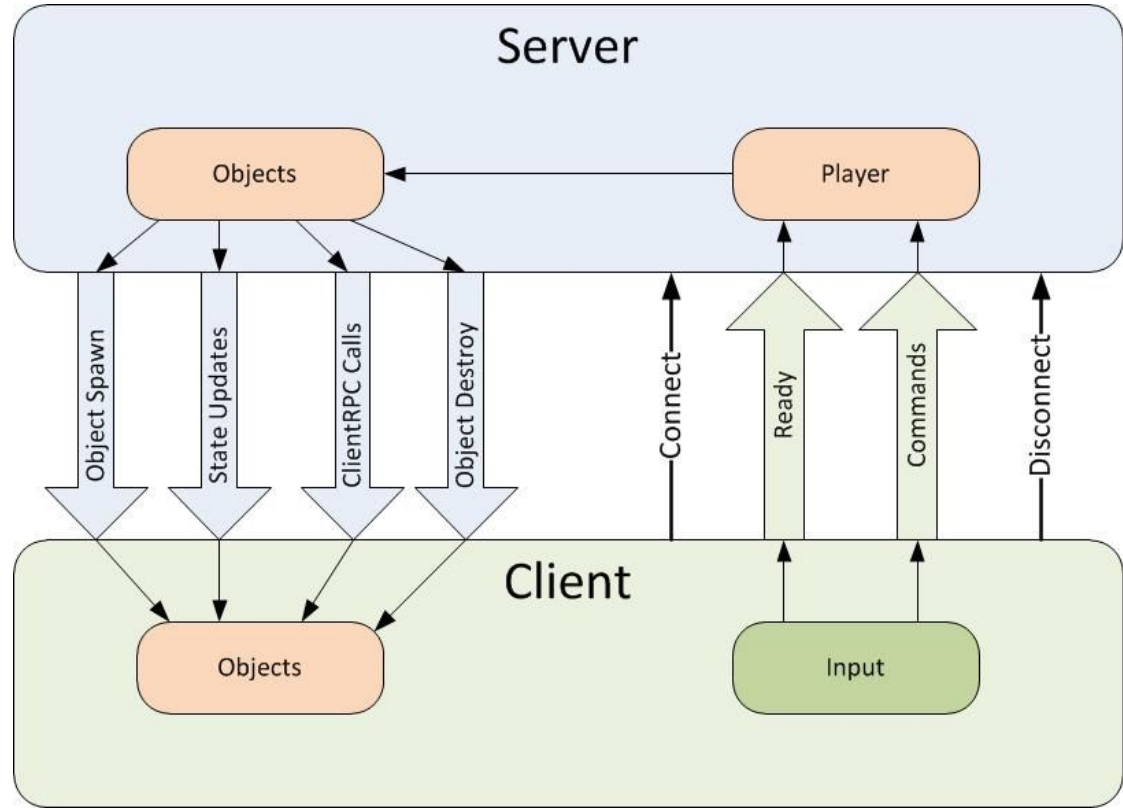
## Objetos sincronizado s





# 17

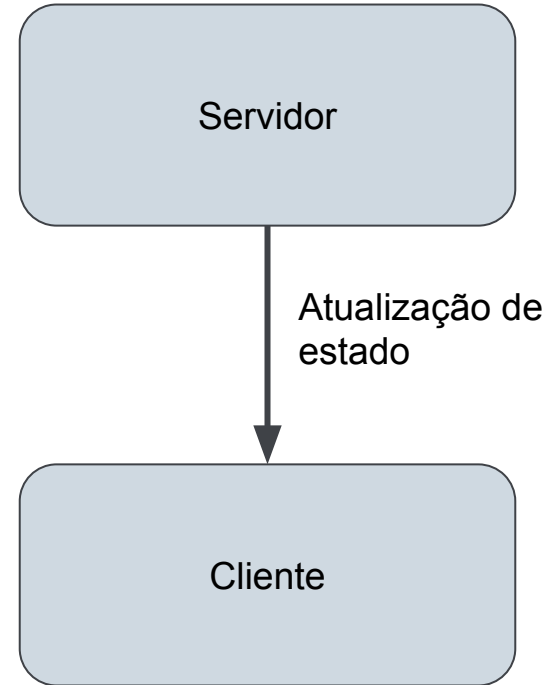
## Sincronismo



# 18

## Variáveis sincronizadas (Syncvars)

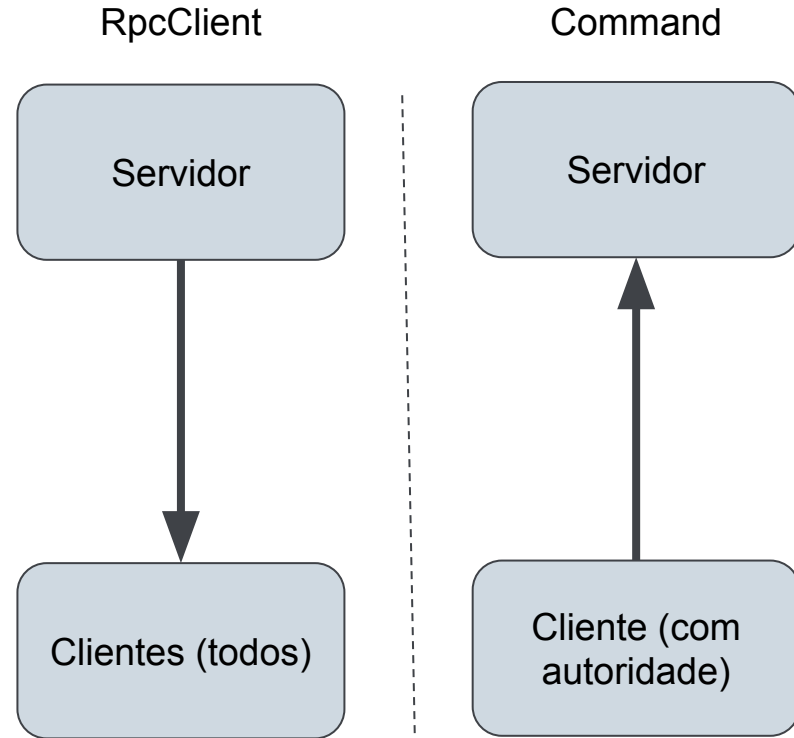
- ▶ Do servidor para o cliente
- ▶ Variáveis com atributo [SyncVar]
- ▶ Pode ser atribuído um callback se desejado
- ▶ É permitido também
  - ▷ SyncLists
  - ▷ SyncDictionary
  - ▷ SyncHashSet
  - ▷ SyncSortedSet



# 19

## Chamadas remotas

- ▶ Podem passar parâmetros
- ▶ RpcClient são chamadas do servidor em todos os clientes
- ▶ Command são chamadas de algum cliente com autoridade para o servidor
- ▶ Pode ser usado também SyncEvent
  - ▷ Similar ao RpcClient
  - ▷ Múltiplas funções inscritas



# 20

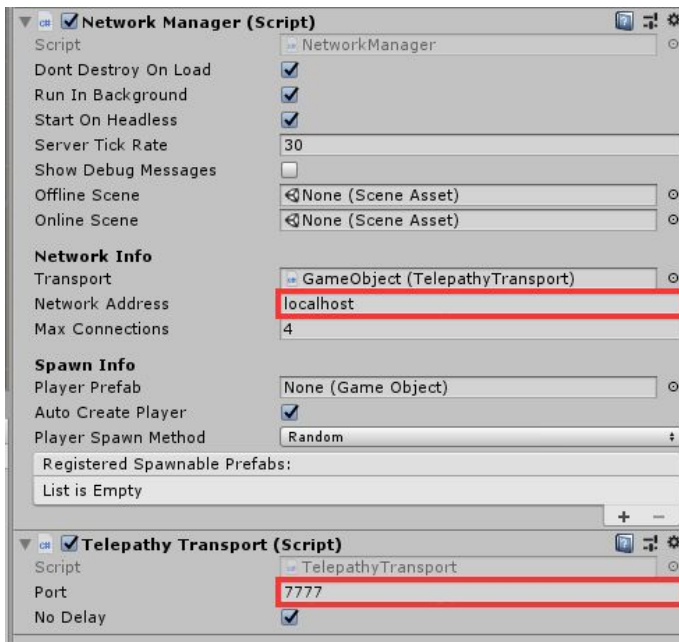
Tipos  
aceitos  
(Estado e  
parâmetros)

- ▶ Tipos básico C# (byte, int, char, uint, float, string, UInt64, etc)
- ▶ Tipos matemático Unity (Vector3, Quaternion, Rect, Plane, Vector3Int, etc)
- ▶ NetworkIdentity
- ▶ GameObject com um componente NetworkIdentity
- ▶ Structures com os tipos acima
- ▶ Arrays com os tipos acima

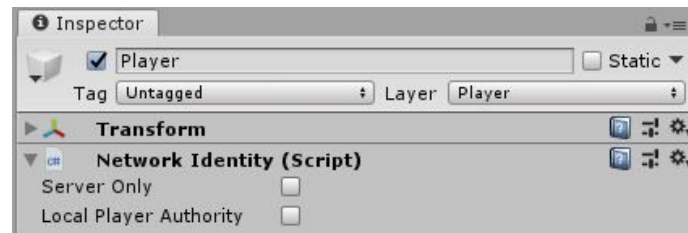
# 21

## Componentes Networked

### NetworkManager



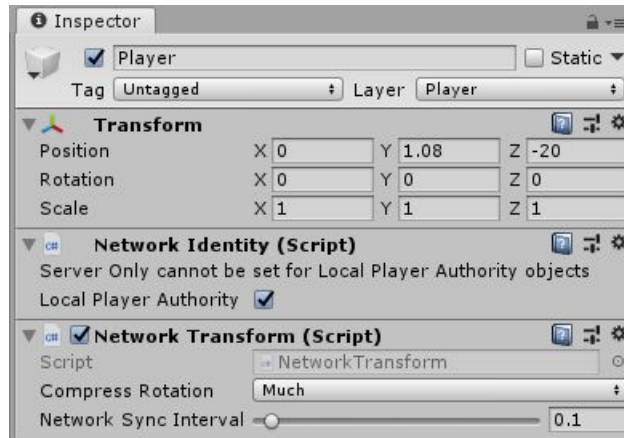
### NetworkIdentity



# 22

## Componentes Networked

### NetworkTransform

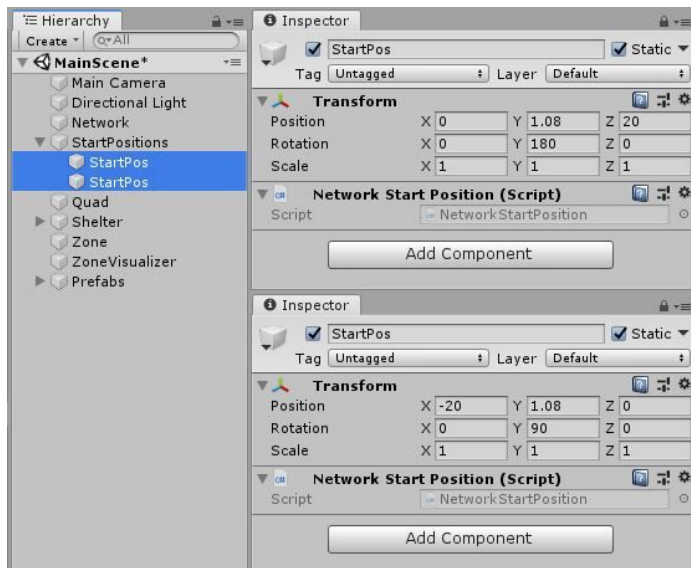


### NetworkTransformChild

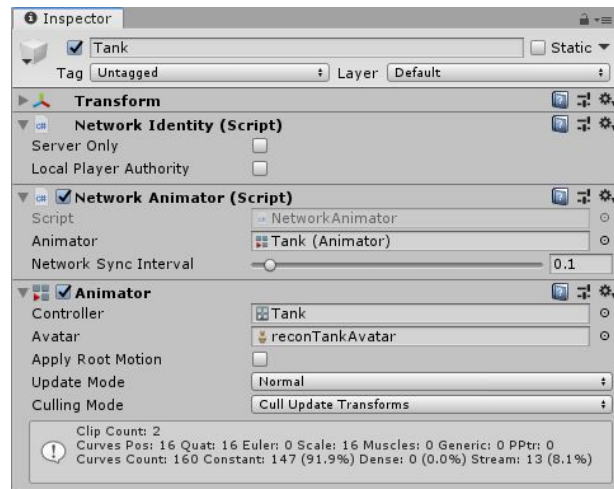


# 23

## NetworkStartPosition



## NetworkAnimator



Score: 290

# Conversão





# 25

Conversão  
de jogo  
singleplayer  
para  
multiplayer

- ▶ Jogo escolhido é o fornecido pelo tutorial:  
<https://learn.unity.com/project/survival-shooter-tutorial>
- ▶ Encontrado completo em:  
<https://assetstore.unity.com/packages/essentials/tutorial-projects/survival-shooter-tutorial-legacy-40756>
- ▶ Seguiremos o guia provido pelo mirror:  
<https://mirror-networking.com/docs/Guides/Conversion.html>

The header features a light blue background with a grid of various icons. The icons include a document, a tag, a puzzle piece, a magnifying glass, a smartphone, a document with lines, a target, a gear, a pie chart, an envelope, a speech bubble, a lightbulb, a clock, a checkmark, and a thumbs up. The number '26' is prominently displayed in a large, bold, blue font on the left side of the header.

# 26

## **Arquitetura recomendada (Unreal Engine 4)**

Servidor

Game Manager

Servidor + Cliente

Game Status  
(Autoridade:  
Servidor)

Player Controller  
(Autoridade: Cliente)

Cliente

Game Instance  
(persiste entre  
cenas)

Criado  
pelo  
usuário

28

# Referências

# 29

## Referências

- ▶ <https://mirror-networking.com/docs>