

Miguel Moura  **up201609149**@fe.up.pt

Antero Santos **up201604602**@fe.up.pt

Simão Santos **up201505695**@fe.up.pt

## Bases de Dados

ASM- Empresa de Videojogos

MIEIC06- Grupo 605

(6 de Março de 2018)

Descrição

Este projeto baseia-se na gestão de uma empresa de Videojogos semelhante á EA com as suas varias divisões.

A companhia possui varias divisões, jogos de desporto, jogos de ação, jogos de carros e etc., sendo que todas elas obedecem á mesma estrutura.

Em primeiro lugar consideremos a classe **Divisão** e as suas sub-relações. Uma **Divisão** é, portanto, composta pelo seu nome e pelo seu ID!

Cada divisão possui uma **Morada**, portanto um nome da rua, a cidade em que está baseada, o número e ultimamente o país.

Os funcionários que trabalharam no jogo podem ser do departamento **Financeiro**, do departamento de **Marketing**, do grupo de **Developers** e o grupo restrito de jogadores chamados de beta **Testers**.

Os **Developers** do jogo são o grupo de **Funcionários** que trabalha diretamente no desenvolvimento do jogo e os **Testers** são um grupo de jogadores que testa o jogo numa fase beta de modo a melhorar o produto final. Um jogador pode ser um **Tester** se for selecionado pelo departamento!

A classe **Jogo** tem, portanto, uma relação direta com a classe **Jogador**, estabelecendo-se por isso entre as duas uma **Playerbase**. Esta **Playerbase** é a comunidade de jogadores que efetivamente joga este jogo.

A comunidade de **Testers** vai atribuir-lhe um rating, estando por isso ligada á classe **Classificação**.

Um **Jogador** pode ainda adquirir uma conta **Premium**, ligada ao departamento **Financeiro** que lhe garante vantagens sobre os Jogadores normais e o acesso a fases *Beta* do jogo como foi referido acima.

Um jogo depois de ser lançado está por isso ligado á classe **Marketing** tendo em conta a sua popularidade e ao departamento **Financeiro** com base na receita gerada.

Tendo por base o rating do jogo, o dinheiro que gera e as recomendações do departamento de **Marketing** e **Financeiro**, a **Divisão** pode escolher terminar o desenvolvimento do **Jogo**.

O **Marketing** está também ligado ao **Financeiro** por causa do financiamento das campanhas de publicidade dos jogos.

Atributos

DIVISÃO

* Nome
* Morada
* Telefone

JOGO

* Nome
* Age Registration

JOGADOR

* In-gme ID
* Data de adesão

PLAYERBASE

* Nº de jogadores
* Idade Média

PREMIUM

* Metodo de pagamento
* Preço

Plataforma

* Nome
* Empresa Dona

FINANCEIRO

* Receita gerada pelo jogo

MARKETING

* Nº de Visualizações

Departamento

* Nº de Funcionários
* ID do departamento

Development

* Data prevista de lançamento

Tester

Classificação

* Rating(1-10)

Esquema Relacional e Dependências Funcionais

* Divisão(id, nome,moradaID->Morada)
  + {Id} - >{nome,moradaID }
* Morada(id, nomeRua, cidade, numeroPorta,país)
  + {Id}-> {nomeRua, cidade, numeroPorta,país}
* Funcionário(id, nome, idade, tempoServiço, depID->Departamento)
  + {Id}->{nome, idade, tempoServiço, depID }
* Departamento(id, nome, tamanhoEquipa)
  + {Id}->{ nome, tamanhoEquipa }
* Jogo(id, nome, playerbaseId->Playerbase)
  + {Id}->{ nome, playerbaseId}
* Playerbase(id, numeroTotal, idadeMédia)
  + {Id}->{ numeroTotal, idadeMédia }
* Jogador(id, username, idade, tester, premium)
  + {Id}->{ username, idade, tester, premium }
* Género(id, nome, idadeRecomendada)
  + {Id}->{ nome, idadeRecomendada }
* Plataforma(id,nome, marca)
  + {Id}->{ nome, marca }
* DepartamentoDivisão(depID->Departamento, divID->Divisão)
* FuncionárioDepartamento(funcID->Funcionário, depID->Departamento)
* JogadorPlayerbase(jogadorID->Jogador, playerbaseID->Playerbase)
* jogoDivisão(jogoId->Jogo, divID->Divisão)
* jogoGénero(jogoID->Jogo, generoID->Genero)
* jogoJogador(jogoId->Jogo, jogadorID->Jogador)
* jogoPlataforma(jogoID->Jogo, plataformaID->Plataforma)

Restrições