

Trabalho de Banco de Dados

Desenvolvimento de um banco de dados relacional para o um jogo de RPG

André Pacheco

José Nilton

Filipe Favoreto

Introdução

- ▶ RPG: *Role-Playing Game*, em português: “Jogo de interpretação de personagens”
- ▶ É um mercado bem aquecido e de alto crescimento
- ▶ Inspirado no League of heroes (Android e iOS)
- ▶ Não temos acesso a modelagem original do jogo



Contexto

▶ Conta:

- ▶ Uma conta deve possuir informações em geral do usuário, como nome, e-mail, país, nickname, senha etc.
- ▶ A partir do momento que o usuário possui uma conta, ele está apto a participar do jogo

▶ Herói:

- ▶ Cada conta possui um herói
- ▶ O herói é o personagem no jogo.
- ▶ Contém informações pertinentes ao jogo como arma, energia, magia, sangue level etc.



Contexto



Contexto

▶ Arma:

- ▶ É a arma que algum usuário pode ter
- ▶ Possui informações de danos, preço, level de pré-requisito...



► **Armadura:**

- ▶ É a armadura que algum usuário pode ter
- ▶ Possui informações de defesa, preço, level de pré-requisito...



Contexto

► Botas:

- É a bota que algum usuário pode ter
- Possui informações de nome, defesa, preço, level de pré-requisito...



Contexto

► Capacetes:

- É o capacete que algum usuário pode ter
- Possui informações de defesa, preço, level de pré-requisito...



Contexto

► Magia:

- É a magia que algum usuário pode ter
- Possui descrição da mesma, nome, preço....



Contexto

► Técnicas:

- É a técnica que algum usuário pode ter
- Possui descrição da mesma, nome, preço....



Contexto

▶ **Mapas:**

- ▶ É o local onde ocorre as batalhas contra os monstros
- ▶ Possuem nome, level de pré-requisito, recompensa, número de monstros...
- ▶ Existe um mapa especial chama do Vila. A vila não existe monstros e é como se fosse o home do jogo



Contexto

► Mapas:



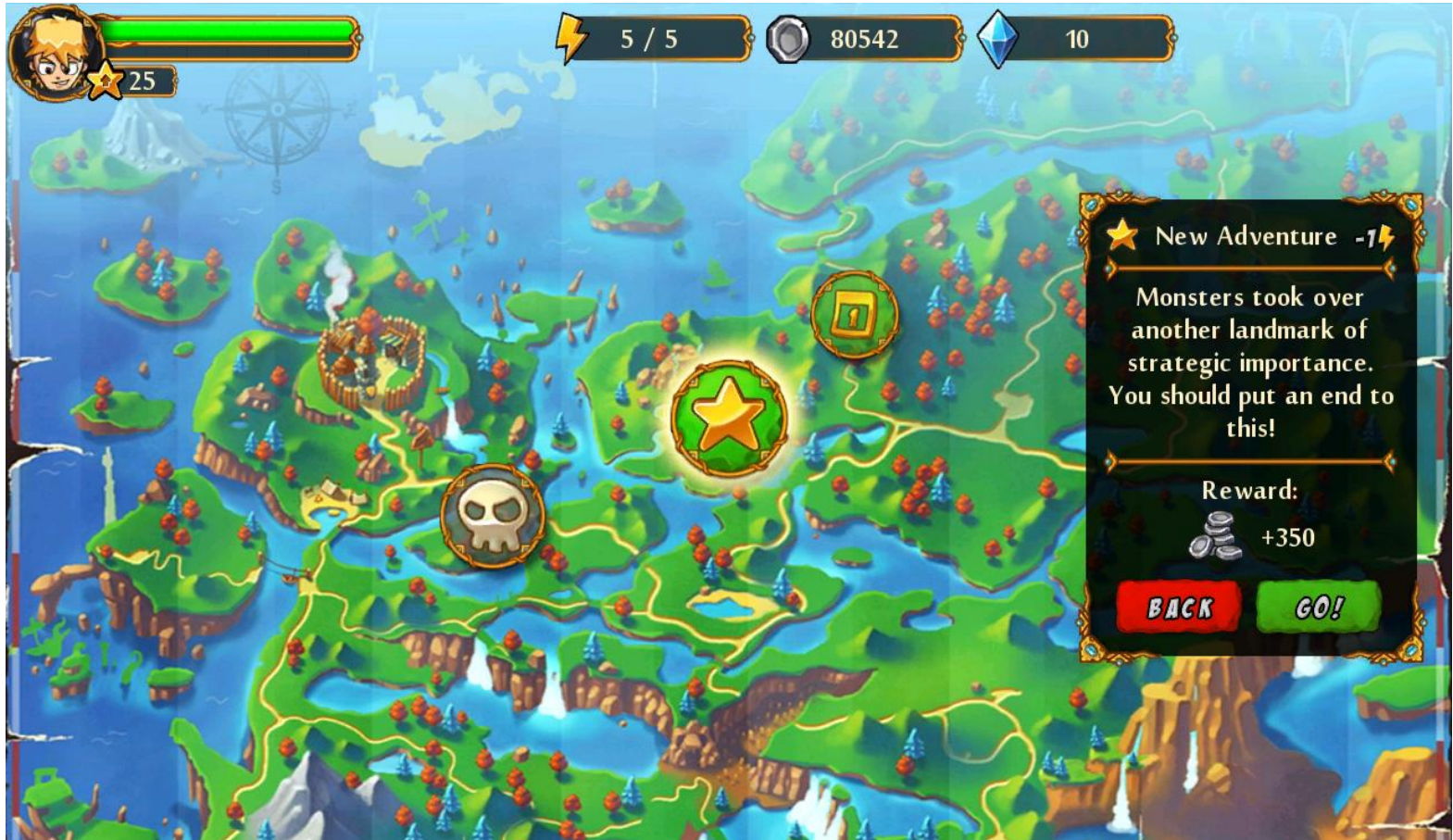
Contexto

► Mapas:



Contexto

► Mapas:



Contexto

▶ **Monstros:**

- ▶ São os inimigos dos heróis
- ▶ Devem possuir níveis de defesa e ataque
- ▶ Deve especificar o mapa em que estão
- ▶ Quando derrotados deixa 'cair' moedas e experiência (representado por estrelas)



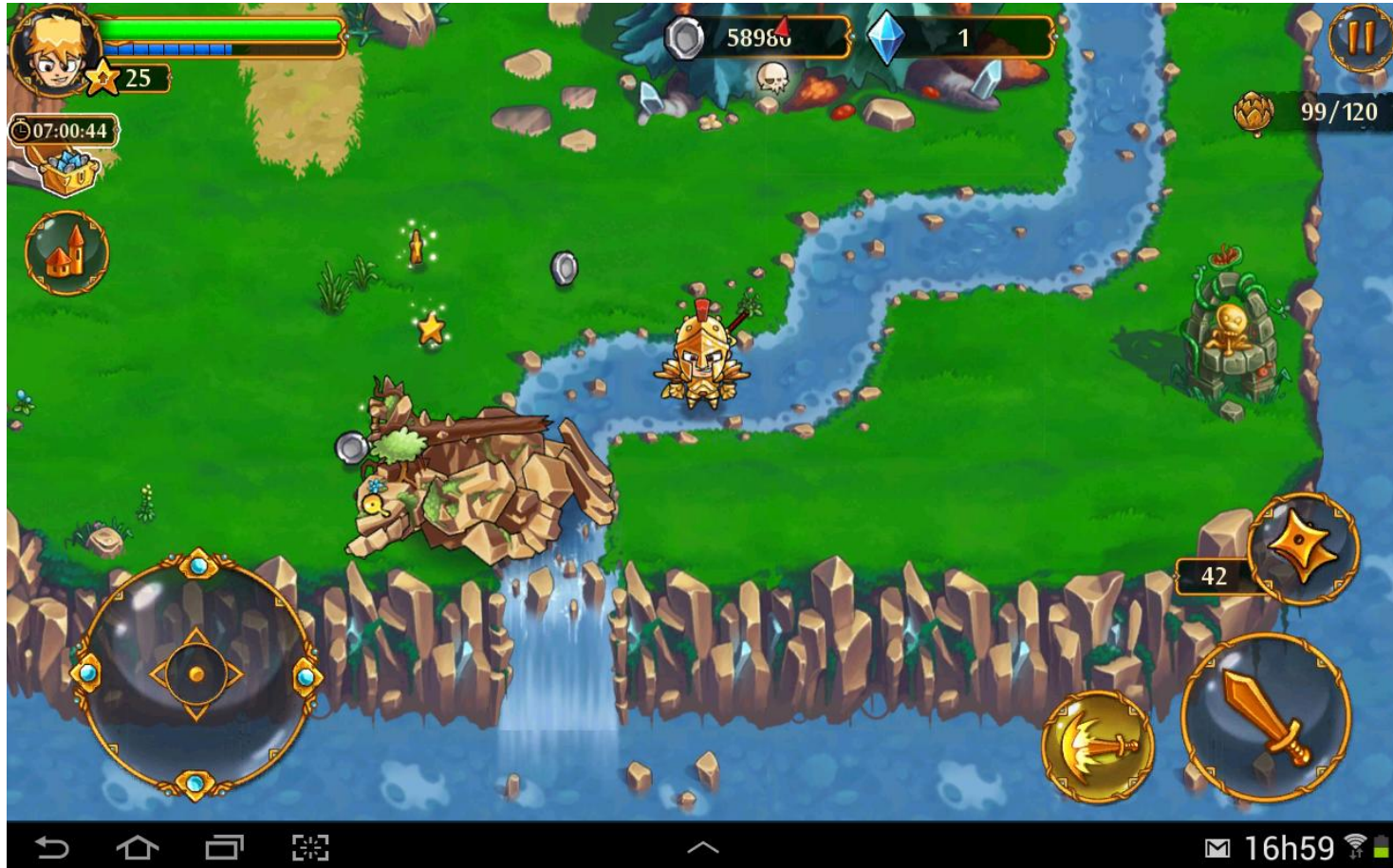
Contexto

► Monstros:



Contexto

► Monstros:



Contexto

▶ **Baús:**

- ▶ São 'tesouros' que o herói pode encontrar no mapa
- ▶ Teoricamente ficam escondidos no mapa
- ▶ Quando encontrados, dão uma recompensa ao herói
- ▶ Deve possuir uma cor e quantas moedas dará de recompensa



Contexto

► Baús:



Contexto

▶ **Desafio diário:**

- ▶ Diariamente o herói possui um desafio
- ▶ Esse desafio possui uma descrição e três missões
- ▶ Ao ser cumprido, ele dará uma recompensa em diamantes para o herói



Contexto

► Desafio diário:



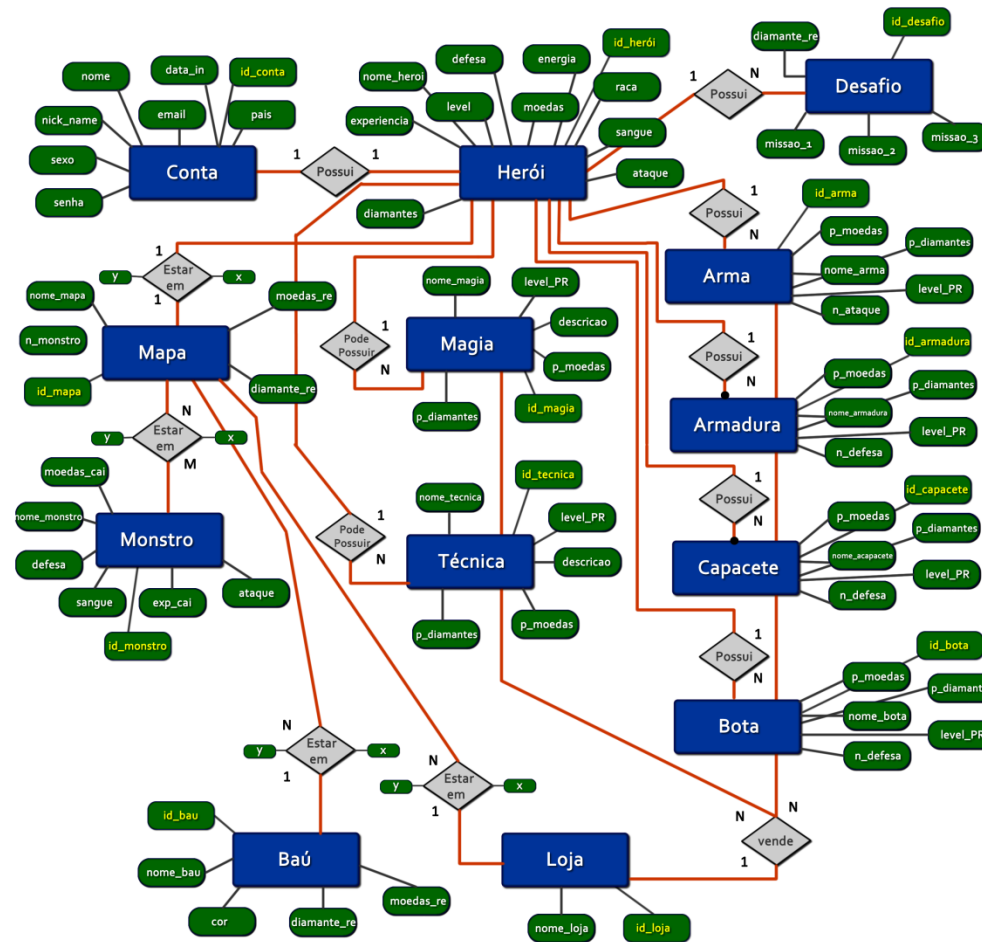
Contexto

► Loja:

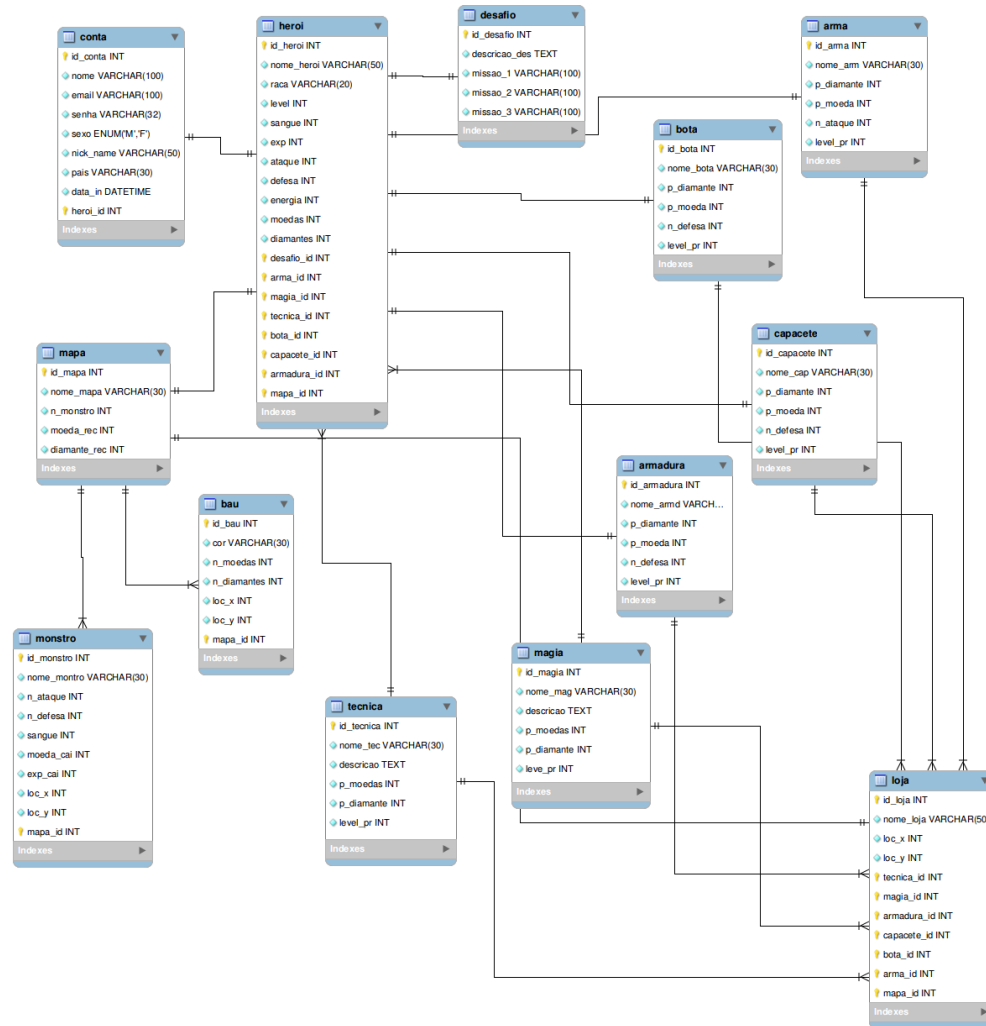
- É o local onde é vendido armadura, magia, técnicas...
- Possui um mapa de localização
- Possui também um nome



Projeto conceitual



Projeto lógico



Implementação física

```
CREATE TABLE IF NOT EXISTS `arma` (  
    `id_arma` int(10) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `nome_arm` varchar(30) NOT NULL,  
    `p_diamante` int(10) unsigned NOT NULL,  
    `p_moeda` int(10) unsigned NOT NULL,  
    `n_ataque` int(10) unsigned NOT NULL,  
    `level_pr` int(10) unsigned NOT NULL,  
    PRIMARY KEY (`id_arma`),  
    UNIQUE KEY (`nome_arm`)  
);
```



Implementação física

```
CREATE TABLE IF NOT EXISTS `armadura` (  
    `id_armadura` int(10) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `nome_armadura` varchar(30) NOT NULL,  
    `p_diamante` int(10) unsigned NOT NULL,  
    `p_moeda` int(10) unsigned NOT NULL,  
    `n_defesa` int(10) unsigned NOT NULL,  
    `level_pr` int(10) unsigned NOT NULL,  
    PRIMARY KEY (`id_armadura`),  
    UNIQUE KEY (`nome_armadura`)  
);
```



Implementação física

```
CREATE TABLE IF NOT EXISTS `bota` (  
    `id_bota` int(10) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `nome_bota` varchar(30) NOT NULL,  
    `p_diamante` int(10) unsigned NOT NULL,  
    `p_moeda` int(10) unsigned NOT NULL,  
    `n_defesa` int(10) unsigned NOT NULL,  
    `level_pr` int(10) unsigned NOT NULL,  
    PRIMARY KEY (`id_bota`),  
    UNIQUE KEY (`nome_bota`)  
);
```



Implementação física

```
CREATE TABLE IF NOT EXISTS `capacete` (  
    `id_capacete` int(10) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `nome_cap` varchar(30) NOT NULL,  
    `p_diamante` int(10) unsigned NOT NULL,  
    `p_moeda` int(10) unsigned NOT NULL,  
    `n_defesa` int(10) unsigned NOT NULL,  
    `level_pr` int(10) unsigned NOT NULL,  
    PRIMARY KEY (`id_capacete`),  
    UNIQUE KEY (`nome_cap`)  
);
```



Implementação física

```
CREATE TABLE IF NOT EXISTS `desafio` (  
    `id_desafio` int(10) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `descricao_des` text NOT NULL,  
    `missao_1` varchar(100) NOT NULL,  
    `missao_2` varchar(100) NOT NULL,  
    `missa_3` varchar(100) NOT NULL,  
    PRIMARY KEY (`id_desafio`),  
);
```



Implementação física

```
CREATE TABLE IF NOT EXISTS `magia` (  
    `id_magia` int(10) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `nome_magia` varchar(30) NOT NULL,  
    `descricao` text NOT NULL,  
    `p_moedas` int(10) unsigned NOT NULL,  
    `p_diamantes` int(10) unsigned NOT NULL,  
    `level_pr` int(10) unsigned NOT NULL,  
    PRIMARY KEY (`id_magia`),  
    UNIQUE KEY (`nome_magia`),  
);
```



Implementação física

```
CREATE TABLE IF NOT EXISTS `tecnica` (  
    `id_tecnica` int(10) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `nome_tec` varchar(30) NOT NULL,  
    `descricao` text NOT NULL,  
    `p_moedas` int(10) unsigned NOT NULL,  
    `p_diamante` int(10) unsigned NOT NULL,  
    `level_pr` int(10) unsigned NOT NULL,  
    PRIMARY KEY (`id_tecnica`),  
    UNIQUE KEY (`nome_tec`)  
);
```



Implementação física

```
CREATE TABLE IF NOT EXISTS `mapa` (  
    `id_mapa` int(10) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `nome_mapa` varchar(30) NOT NULL,  
    `n_monstro` int(10) unsigned NOT NULL,  
    `moeda_rec` int(10) unsigned NOT NULL,  
    `diamante_rec` int(10) unsigned NOT  
NULL,  
    PRIMARY KEY (`id_mapa`),  
    UNIQUE KEY (`nome_mapa`)  
);
```



Implementação física

```
CREATE TABLE IF NOT EXISTS `bau` (  
    `id_bau` int(10) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `cor` varchar(30) NOT NULL,  
    `n_moedas` int(10) unsigned NOT NULL,  
    `n_diamantes` int(10) unsigned NOT NULL,  
    `loc_x` int(10) unsigned NOT NULL,  
    `loc_y` int(10) unsigned NOT NULL,  
    `mapa_id` int(10) unsigned NOT NULL,  
    PRIMARY KEY (`id_bau`),  
    UNIQUE KEY (`cor`),  
    FOREIGN KEY (`mapa_id`) REFERENCES `mapa`  
    (`id_mapa`) ON DELETE RESTRICT  
);
```



Implementação física

```
CREATE TABLE IF NOT EXISTS `loja` (  
    `id_loja` int(10) unsigned NOT NULL AUTO_INCREMENT,  
    `nome_loja` varchar(50) NOT NULL,  
    `loc_x` int(10) unsigned NOT NULL,  
    `loc_y` int(10) unsigned NOT NULL,  
    `tecnica_id` int(10) unsigned NOT NULL,  
    `magia_id` int(10) unsigned NOT NULL,  
    `armadura_id` int(10) unsigned NOT NULL,  
    `arma_id` int(10) unsigned NOT NULL,  
    `capacete_id` int(10) unsigned NOT NULL,  
    `bota_id` int(10) unsigned NOT NULL,  
    PRIMARY KEY (`id_loja`),  
    UNIQUE KEY (`nome_loja`),  
    FOREIGN KEY (`tecnica_id`) REFERENCES `tecnica` (`id_tecnica`) ON DELETE  
RESTRICT,  
    FOREIGN KEY (`magia_id`) REFERENCES `magia` (`id_magia`) ON DELETE  
RESTRICT,  
    FOREIGN KEY (`armadura_id`) REFERENCES `armadura` (`id_armadura`) ON DELETE  
RESTRICT,  
    FOREIGN KEY (`arma_id`) REFERENCES `arma` (`id_arma`) ON DELETE RESTRICT,  
    FOREIGN KEY (`capacete_id`) REFERENCES `capacete` (`id_capacete`) ON DELETE  
RESTRICT,  
    FOREIGN KEY (`bota_id`) REFERENCES `bota` (`id_bota`) ON DELETE RESTRICT  
    FOREIGN KEY (`mapa_id`) REFERENCES `bota` (`id_bota`) ON DELETE RESTRICT  
);
```

Implementação física

```
CREATE TABLE IF NOT EXISTS `monstro` (  
    `id_monstro` int(10) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `nome_monstro` varchar(30) NOT NULL,  
    `n_ataque` int(10) unsigned NOT NULL,  
    `n_defesa` int(10) unsigned NOT NULL,  
    `sangue` int(10) unsigned NOT NULL,  
    `moeda_cai` int(10) unsigned NOT NULL,  
    `exp_cai` int(10) unsigned NOT NULL,  
    `loc_x` int(10) unsigned NOT NULL,  
    `loc_y` int(10) unsigned NOT NULL,  
    `mapa_id` int(10) unsigned NOT NULL,  
    PRIMARY KEY (`id_monstro`),  
    UNIQUE KEY (`nome_monstro`),  
    FOREIGN KEY (`mapa_id`) REFERENCES  
    `mapa` (`id_mapa`) ON DELETE RESTRICT  
);
```



Implementação física

```
CREATE TABLE IF NOT EXISTS `heroi` (  
  `id_heroi` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `nome_heroi` varchar(50) NOT NULL,  
  `raca` varchar(20) NOT NULL,  
  `level` int(10) unsigned NOT NULL,  
  `sangue` int(10) unsigned NOT NULL,  
  `exp` int(10) unsigned NOT NULL,  
  `ataque` int(10) unsigned NOT NULL,  
  `defesa` int(10) unsigned NOT NULL,  
  `energia` int(10) unsigned NOT NULL,  
  `moedas` int(10) unsigned NOT NULL,  
  `diamantes` int(10) unsigned NOT NULL,  
  `desafio_id` int(10) unsigned NOT NULL,  
  `arma_id` int(10) unsigned NOT NULL,  
  `armadura_id` int(10) unsigned NOT NULL,  
  `magia_id` int(10) unsigned NOT NULL,  
  `tecnica_id` int(10) unsigned NOT NULL,  
  `bota_id` int(10) unsigned NOT NULL,  
  `capacete_id` int(10) unsigned NOT NULL,  
  `mapa_id` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`id_heroi`),  
  UNIQUE KEY (`nome_heroi`),  
  FOREIGN KEY (`desafio_id`) REFERENCES `desafio` (`id_desafio`) ON DELETE RESTRICT,  
  FOREIGN KEY (`arma_id`) REFERENCES `arma` (`id_arma`) ON DELETE RESTRICT,  
  FOREIGN KEY (`armadura_id`) REFERENCES `armadura` (`id_armadura`) ON DELETE RESTRICT,  
  FOREIGN KEY (`magia_id`) REFERENCES `magia` (`id_magia`) ON DELETE RESTRICT,  
  FOREIGN KEY (`tecnica_id`) REFERENCES `tecnica` (`id_tecnica`) ON DELETE RESTRICT,  
  FOREIGN KEY (`bota_id`) REFERENCES `bota` (`id_bota`) ON DELETE RESTRICT,  
  FOREIGN KEY (`capacete_id`) REFERENCES `capacete` (`id_capacete`) ON DELETE RESTRICT,  
  FOREIGN KEY (`mapa_id`) REFERENCES `mapa` (`id_mapa`) ON DELETE RESTRICT  
);
```



Implementação física

```
CREATE TABLE IF NOT EXISTS `conta` (  
    `id_conta` int(10) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `nome` varchar(100) NOT NULL,  
    `email` varchar(100) NOT NULL,  
    `senha` varchar(32) NOT NULL,  
    `nick_mane` varchar(50) NOT NULL,  
    `sexo` enum('M','F') NOT NULL,  
    `pais` varchar(30) NOT NULL,  
    `data_in` datetime NOT NULL,  
    `heroi_id` int(10) unsigned NOT NULL,  
    PRIMARY KEY (`id_conta`),  
    UNIQUE KEY (`email`,`nick_mane`),  
    FOREIGN KEY (`heroi_id`) REFERENCES `heroi`  
    (`id_heroi`) ON DELETE RESTRICT  
);
```



Inserção de dados nas tabela

```
INSERT INTO `arma` (`id_arma`, `nome_arm`,  
`p_diamante`, `p_moeda`, `n_ataque`, `level_pr`) VALUES  
(1, 'Espada de aço', 5, 1000, 6, 1),  
(2, 'Espada de fogo', 15, 5000, 14, 5),  
(3, 'Machado bolado', 16, 7000, 17, 7),  
(4, 'Arco do Legolas', 50, 50000, 50, 20),  
(5, 'Martelo de Thor', 30, 35000, 41, 15),  
(6, 'Machado simples', 2, 500, 3, 1),  
(7, 'Cacetete do BME', 1, 200, 1, 1),  
(8, 'Sabre de luz', 60, 70000, 73, 25),  
(9, 'Purrete', 5, 1200, 8, 2),  
(10, 'Faquinha do CS', 2, 800, 3, 1);
```



Inserção de dados nas tabela

```
INSERT INTO `conta` (`id_conta`, `nome`, `email`, `senha`, `nick_mane`, `sexo`,  
`pais`, `data_in`, `heroi_id`) VALUES  
(1, 'Andre', 'andre@gmail.com', '1234', 'andre', 'M', 'Brasil', '2010-07-01 04:00:00', 6),  
(2, 'Alvaro', 'alvaro@inf.com', '1234', 'alvs', 'M', 'Brasil', '2013-07-07 19:00:00', 9),  
(3, 'Lucas', 'lucas@gmail.com', '1234', 'lusk', 'M', 'USA', '2009-04-09 15:00:00', 1),  
(4, 'Maria', 'maria@gmail.com', '1234', 'mariazinha', 'F', 'Brasil', '2011-07-07 10:00:00', 2),  
(5, 'Paulo', 'paulo@gmail.com', '1234', 'paul', 'M', 'USA', '2011-07-07 00:00:00', 7),  
(6, 'Tiago', 'tiago@gmail.com', '1234', 't2', 'M', 'Brasil', '2012-07-07 00:00:00', 10),  
(7, 'Henrique', 'hq@gmail.com', '1234', 'h4e1', 'M', 'USA', '2011-07-07 00:00:00', 8),  
(8, 'Joana', 'jojo@gmail.com', '1234', 'jojo', 'F', 'Brasil', '2012-07-07 00:00:00', 3),  
(9, 'Luisa', 'lulu@gmail.com', '1234', 'lulu', 'F', 'Canada', '2011-07-07 00:00:00', 5),  
(10, 'Artur', 'tutu@gmail.com', '1234', 'tutu', 'M', 'Brasil', '2010-07-07 00:00:00', 4);
```



Inserção de dados nas tabela

INSERT INTO `desafio` (`id_desafio`, `descricao_des`, `missao_1`, `missao_2`, `missao_3`) VALUES

(1, 'Desafio do dia: Este é apenas um aquecimento. Cumpra as missões a seguir:', 'Mate 20 monstros', 'Não perca sangue', 'Gaste toda sua energia'),
(2, 'Desafio do dia: A vila está em perigo. Você deve proteja-la o mais rápido possível. As missões são:', 'Colete 5000 moedas', 'Mate 50 monstros', 'Fique 1h na vila'),
(3, 'Desafio do dia: O quão rápido você é? Proteja-se!', 'Encontre 10 baús', 'Conquiste 2 diamantes', 'Obtenha uma nova técnica'),
(4, 'Desafio do dia: A vila foi atacada enquanto você esteve fora. E agora José?', 'Mate 100 monstros', 'Utilize uma magia', 'Compre uma nova arma'),
(5, 'Desafio do dia: Você se considera forte o suficiente? Saiba que nunca está preparado o suficiente.', 'Compre uma nova bota', 'Compre um novo capacete', 'Compre uma nova armadura'),
(7, 'Desafio do dia: A magia é tão importante quanto a técnica.', 'Compre uma nova técnica', 'Compre uma nova magia', 'Utilize uma técnica'),
(8, 'Desafio do dia: Sua vila está falindo. Salve-a', 'Recolha 20 mil moedas', 'Recolha 5 diamantes', 'Visite 3 mapas diferentes'),
(9, 'Desafio diario: Subir de level é sempre bom. Você precisa de mais experiência.', 'Conquiste 100 de experiência', 'Encontre 3 baús azuis', 'Suba de level'),
(10, 'Desafio diario: Os monstros estão se aproximando. Haja!', 'Mate 50 monstros', 'Gaste toda sua energia', 'Use uma magia'),
(11, 'Desafio diario: Você se tornou um bom guerreiro, porém pode melhorar.', 'Suba de level', 'Compre uma nova arma', 'Aprenda uma nova técnica');



Inserção de dados nas tabela

```
INSERT INTO `heroi` (`id_heroi`, `nome_heroi`, `raca`, `level`, `sangue`, `exp`, `ataque`,  
`defesa`, `energia`, `moedas`, `diamantes`, `desafio_id`, `arma_id`, `armadura_id`, `magia_id`,  
`tecnica_id`, `bota_id`, `capacete_id`, `mapa_id`) VALUES  
(1, 'Goku', 'Saiyajin', 37, 100, 22, 39, 25, 90, 107000, 25, 1, 1, 1, 1, 1, 1, 1, 1),  
(2, 'Brandon', 'Humano', 20, 89, 15, 22, 15, 92, 40000, 14, 2, 2, 2, 2, 2, 2, 2, 2),  
(3, 'Legolas', 'Elfo', 44, 70, 40, 45, 30, 67, 44599, 28, 7, 4, 9, 5, 7, 8, 5, 6),  
(4, 'Hodor', 'Orc', 10, 87, 5, 12, 8, 90, 4500, 3, 8, 1, 2, 8, 5, 2, 3, 4),  
(5, 'Thor', 'Deus', 15, 98, 10, 38, 20, 77, 30000, 39, 10, 5, 7, 6, 8, 10, 9, 8),  
(6, 'Darth Vader', 'Jedi_Sith', 50, 100, 40, 83, 70, 100, 200000, 50, 5, 8, 10, 1, 9, 9, 9, 10),  
(7, 'Anakin', 'Jedi', 60, 100, 45, 93, 50, 100, 300000, 55, 9, 8, 5, 10, 5, 4, 6, 8),  
(8, 'Batman', 'Morcego', 25, 94, 18, 21, 10, 85, 40000, 15, 2, 3, 6, 8, 1, 2, 1, 1),  
(9, 'Joãozinho', 'Humano', 5, 75, 2, 5, 2, 48, 300, 1, 2, 6, 1, 2, 2, 1, 1, 2),  
(10, 'Vegeta', 'Saiyajin', 20, 57, 12, 19, 15, 78, 34500, 16, 9, 2, 7, 6, 3, 9, 5, 9);
```



Inserção de dados nas tabela

```
INSERT INTO `mapa` (`id_mapa`, `nome_mapa`, `n_monstro`,  
`moeda_rec`, `diamante_rec`) VALUES  
(1, 'Vila', 0, 0, 0),  
(2, 'Caverna do dragão', 5, 500, 1),  
(3, 'Winterfell', 7, 1000, 2),  
(4, 'Castelo Negro', 10, 2000, 2),  
(5, 'Colatina', 14, 3000, 3),  
(6, 'Ilha da perdição', 13, 3500, 2),  
(7, 'Zumbiland', 20, 4000, 4),  
(8, 'Serra', 25, 4500, 4),  
(9, 'Parque da cebola', 5, 100, 0),  
(10, 'UFES', 50, 5000, 5);
```



Inserção de dados nas tabela

```
INSERT INTO `monstro` (`id_monstro`, `nome_monstro`, `n_ataque`,  
`n_defesa`, `sangue`, `moeda_cai`, `exp_cai`, `loc_x`, `loc_y`, `mapa_id`) VALUES  
(1, 'Freeza', 13, 5, 20, 20, 5, 4, 8, 2),  
(2, 'Lobo', 15, 10, 15, 15, 3, 5, 9, 10),  
(3, 'Kymera', 30, 30, 30, 25, 6, 1, 0, 3),  
(4, 'Porco Assassino', 15, 10, 20, 5, 4, 8, 11, 4),  
(5, 'Boto Rosa', 50, 50, 50, 10, 10, 10, 3, 5),  
(6, 'Urso Negro', 40, 20, 35, 17, 6, 3, 9, 6),  
(7, 'Zumbi', 40, 10, 40, 7, 3, 9, 9, 7),  
(8, 'Funkeiro do transcol', 50, 15, 20, 0, 0, 3, 2, 8),  
(9, 'Mulher de algodão', 43, 24, 30, 4, 4, 6, 7, 9),  
(10, 'Buosi', 100, 100, 100, 5, 10, 3, 2, 10);
```



Consultas

1 - Quais os principais dados cadastrais de usuários acima do level 10?

```
SELECT NICK_MANE,NOME,PAIS,EMAIL  
FROM CONTA JOIN HEROI ON HEROI.ID_HEROI = CONTA.HEROI_ID  
WHERE HEROI.LEVEL > 10 ORDER BY PAIS;
```

2 - Qual o número de mulheres jogando em cada país?

```
SELECT COUNT(ID_CONTA) AS QTDE, SEXO, PAIS  
FROM CONTA  
WHERE SEXO LIKE "F"  
GROUP BY PAIS  
ORDER BY QTDE;
```

3 - Quantas são as técnicas que o herói ainda não possui pré-requisito?

```
SELECT ID_HEROI, LEVEL, LEVEL_PR  
FROM CONTA,HEROI,TECNICA  
WHERE ID_HEROI = HEROI_ID  
HAVING CAST(TECNICA.LEVEL_PR AS SIGNED) - CAST(HEROI.LEVEL AS  
SIGNED) > 0;
```



Consultas

► 4 - Quantos heróis estão na vila?

```
SELECT DISTINCT NOME_HEROI, EMAIL, PAIS  
FROM HEROI, CONTA  
WHERE ID_HEROI = HEROI_ID  
AND NOT EXISTS (SELECT * FROM MONSTRO  
WHERE HEROI.MAPA_ID = MONSTRO.MAPA_ID);
```

5 - Qual a quantidade de heróis que estão aptos a comprar determinada arma?

```
SELECT COUNT(NOME_HEROI), NOME_ARM  
FROM HEROI, ARMA  
WHERE MOEDAS > P_MOEDA GROUP BY NOME_ARM;
```



Consultas

6 – Quais as distâncias dos monstros para cada baú?

```
SELECT ID_BAU,COR,NOME_MONSTRO,  
SQRT( POW(CAST(BAU.LOC_X AS SIGNED) - CAST(MONSTRO.LOC_X AS SIGNED),  
2 ) + POW(CAST(BAU.LOC_Y AS SIGNED) - CAST(MONSTRO.LOC_Y AS SIGNED),  
2 ) ) AS DISTANCIA  
FROM MONSTRO, BAU  
WHERE BAU.MAPA_ID = MONSTRO.MAPA_ID  
ORDER BY ID_BAU,DISTANCIA ASC;
```

7 - Quais os monstros mais próximos de cada baú?

```
SELECT ID_BAU,COR,NOME_MONSTRO,  
min(SQRT(POW(CAST(BAU.LOC_X AS SIGNED) - CAST(MONSTRO.LOC_X AS  
SIGNED), 2) + POW(CAST(BAU.LOC_Y AS SIGNED) - CAST(MONSTRO.LOC_Y AS  
SIGNED), 2 ) )) AS DISTANCIA  
FROM MONSTRO, BAU  
WHERE BAU.MAPA_ID = MONSTRO.MAPA_ID  
group by id_bau  
ORDER BY ID_BAU,DISTANCIA ASC;
```



Consultas

8 - Quantas contas foram criadas entre as datas X e Y?

```
SELECT NOME, EMAIL, DATA_IN, NOME_HEROI, HEROI.EXP  
FROM CONTA  
LEFT OUTER JOIN HEROI ON ID_HEROI = HEROI_ID  
WHERE DATA_IN BETWEEN '2011-12-01' AND NOW();
```

9 – Quanto tem em experiência e moedas as raças que contém técnicas acima do nível 5?

```
SELECT RACA, SUM(EXP) AS EXPERIENCIA, SUM(MOEDAS) AS TOTAL,  
    LEVEL_PR  
FROM HEROI, TECNICA  
WHERE ID_TECNICA = TECNICA_ID  
GROUP BY RACA  
HAVING LEVEL_PR > 5;
```



Consultas

10 - Quais equipamentos os heróis usam em cada mapa de batalha?

```
SELECT NOME_HEROI, NOME_CAP, ARMA.NOME_ARM,  
        ARMADURA.NOME_ARM, NOME_BOTA, NOME_TEC, NOME_MAGIA  
FROM HEROI  
LEFT OUTER JOIN CAPACETE ON ID_CAPACETE = CAPACETE_ID  
LEFT OUTER JOIN BOTA ON ID_BOTA = BOTA_ID  
LEFT OUTER JOIN ARMA ON ID_ARMA = ARMA_ID  
LEFT OUTER JOIN ARMADURA ON ID_ARMADURA = ARMADURA_ID  
LEFT OUTER JOIN TECNICA ON ID_TECNICA = TECNICA_ID  
LEFT OUTER JOIN MAGIA ON ID_MAGIA = MAGIA_ID  
WHERE ID_HEROI IN (SELECT ID_HEROI FROM HEROI, MONSTRO  
WHERE HEROI.MAPA_ID = MONSTRO.MAPA_ID);
```



Consultas

11 - Qual o nome do usuário e de seu herói, que estão em mapas que possuem monstros com nível de ataque acima de 5?

```
SELECT NICK_MANE, NOME_HEROI, LEVEL, ATAQUE  
FROM CONTA, HEROI  
WHERE ID_HEROI = HEROI_ID AND MAPA_ID  
IN (SELECT MONSTRO.MAPA_ID FROM MONSTRO WHERE N_ATAQUE > 5);
```



Consultas

12 – Qual nome e level dos heróis que suas contas possuem cadastro de e-mail de domínio @inf?

```
SELECT NOME_HEROI, LEVEL  
FROM HEROI  
WHERE ID_HEROI  
IN (SELECT heroi_id FROM conta WHERE email like '%inf%');
```

12 (2) - Quais os heroes pertencentes a contas com email de dominio inf?

```
SELECT NOME_HEROI, LEVEL  
FROM HEROI  
WHERE exists (SELECT 1 FROM conta WHERE email like '%inf%' and id_heroi  
= heroi_id);
```



Consultas

13 - VIEW Para um usuario poder acompanhar que heroi tem o level mais alto

CREATE VIEW maior_level **AS**

SELECT NOME_HEROI, NOME_CAP, ARMA.NOME_ARM **as** Arma,
ARMADURA.NOME_ARM **as** armadura, NOME_BOTA, NOME_TEC,
NOME_MAGIA, **max (LEVEL)**

FROM HEROI

LEFT OUTER JOIN CAPACETE **ON** ID_CAPACETE = CAPACETE_ID

LEFT OUTER JOIN BOTA **ON** ID_BOTA = BOTA_ID

LEFT OUTER JOIN ARMA **ON** ID_ARMA = ARMA_ID

LEFT OUTER JOIN ARMADURA **ON** ID_ARMADURA = ARMADURA_ID

LEFT OUTER JOIN TECNICA **ON** ID_TECNICA = TECNICA_ID

LEFT OUTER JOIN MAGIA **ON** ID_MAGIA = MAGIA_ID;



Consultas

14 - Ver os usuarios que se cadastraram no dia atual

```
CREATE VIEW cadastros_dia AS  
SELECT NOME, EMAIL, DATA_IN, NOME_HEROI, PAIS  
FROM CONTA  
LEFT OUTER JOIN HEROI ON ID_HEROI = HEROI_ID  
WHERE DATA_IN BETWEEN date_sub(now (), interval 1 day) AND NOW ();
```



Algebra Relacional

1 - Email das contas que contêm heróis humanos:

$\Pi \text{ conta.email } (\sigma \text{ heroi.raca} = \text{'Humanos'} (\text{heroi } \bowtie (\text{heroi.id_heroi} = \text{conta.heroi_id}) \text{ conta}))$

2 - Nome do mapa em que está a conta de ID = 7

$\Pi \text{ mapa.nome_mapa } (\sigma \text{ conta.id_conta} = \text{'7'} (\text{conta } \bowtie (\text{conta.heroi_id} = \text{heroi.id_heroi}) \text{ heroi })$
 $\bowtie (\text{heroi.mapa_id} = \text{mapa.id_mapa}) \text{ mapa})$

3 - Qual a ultima missao do desafio de Goku?

$\Pi \text{ desafio.missao_3 } (\sigma \text{ heroi.nome_heroi} = \text{'Goku'} (\text{heroi } \bowtie (\text{heroi.desafio_id} =$
 $\text{desafio.id_desafio}) \text{ desafio})$

