

INSTITUTO FEDERAL DO ESPÍRITO SANTO - CAMPUS SERRA  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

PEDRO MATHIAS NAKIBAR  
MATEUS DE CAMPOS

PROJETO DE BANCO DE DADOS:  
STARSPAWN STUDIOS  
INTERGALACTIC BREED

SERRA  
2014

# SUMÁRIO

- [1. MINI-MUNDO](#)
- [2. MODELO CONCEITUAL](#)
- [3. NORMALIZAÇÃO](#)
  - [3.1 0FN](#)
  - [3.2 3FN](#)
  - [3.3 Comentários](#)
- [4. MODELO ENTIDADE-RELACIONAMENTO](#)

## 1. MINI-MUNDO

A empresa Starspawn Studios está trabalhando em um jogo de exploração espacial chamado Intergalactic Breed. Para entrar no jogo, um jogador precisa se cadastrar, inserindo seu nome, email, “nickname” e a senha, ou fazer login com seus dados, se já tiver cadastro. Como usuário logado, no início do jogo ele deve escolher em qual dos três níveis de dificuldade deseja jogar, sendo eles “Easy”, “Medium” e “Hard”. Cada nível oferece uma quantia inicial de Gold, a moeda do jogo, com a qual pode comprar diversos itens durante a campanha.

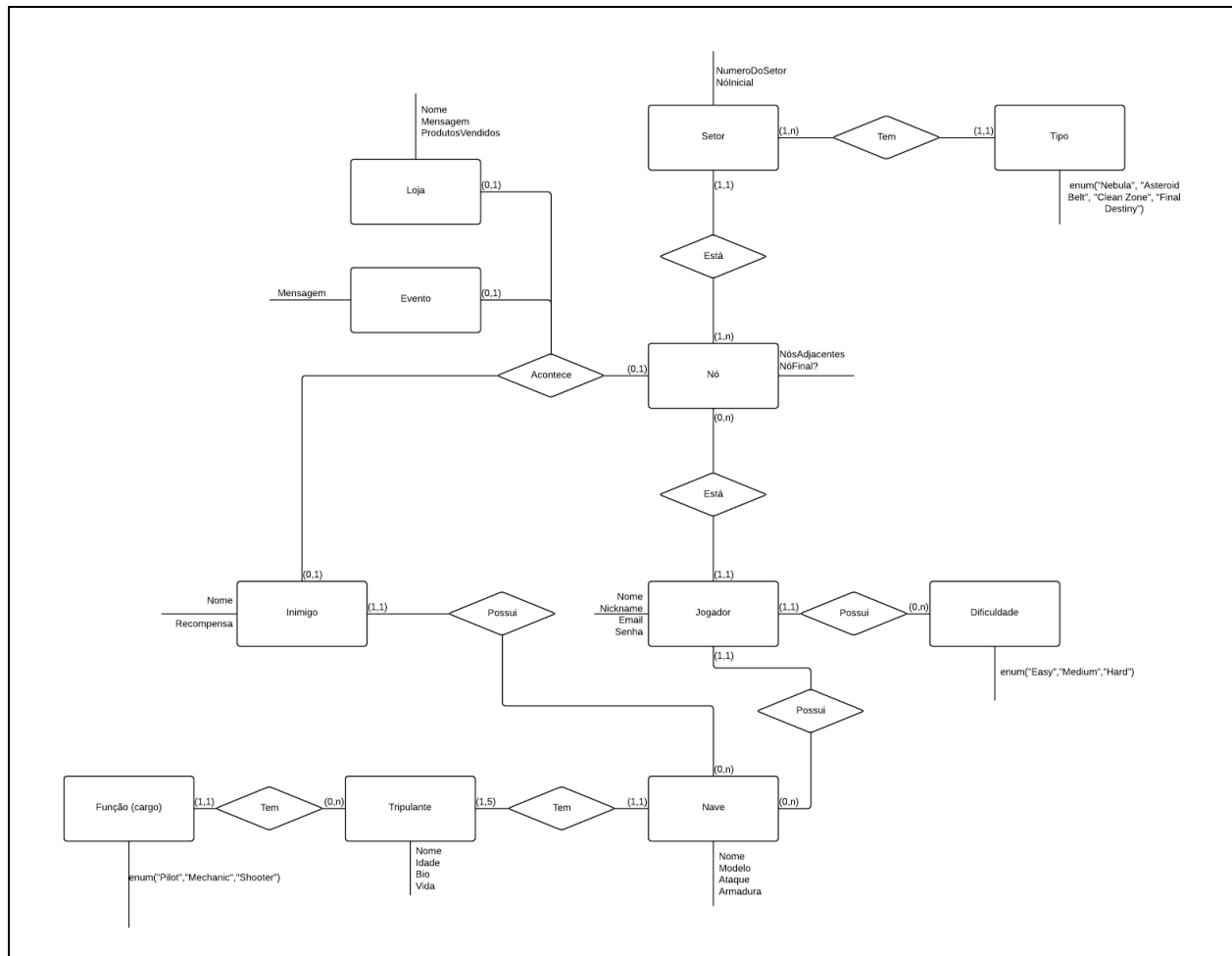
Cada jogador também escolhe uma nave entre modelos específicos para controlar no jogo. Cada nave possui duas qualidades: poder de ataque e armadura, sendo que, se o nível de armadura chegar a zero, a nave é destruída. Todas as naves possuem uma tripulação de 5 personagens aleatórios, que possuem nome, idade, uma breve biografia e pontos de vida, que quando chegam a zero, deixam o personagem fora de jogo até ser revivido ou substituído.

Cada membro da tripulação pode ser designado para uma função ou não, os que não forem designados viram automaticamente “Mechanic”. As funções são: “Pilot”, “Shooter” e “Mechanic”.

A campanha começa no setor 0 do universo do jogo e termina quando o “Emperor of the Galaxy” é derrotado no último setor. O jogador pode apenas seguir em frente, sem poder retornar aos setores já superados, que são indicados por números de 0 a 8. Cada setor pode ser de três tipos: “Nebula”, “Asteroid Belt” ou “Clean Zone”, e é dividido em uma quantidade aleatória de nós que podem ser acessados livremente pelo jogador. Nesses nós ocorrem as situações aleatórias do jogo, como encontrar com inimigos ou algum evento que pode ajudar ou atrapalhar o jogador. Os inimigos (naves e tripulantes) possuem qualidades semelhantes à nave do jogador: pontos de vida e pontos de ataque; sendo que ao destruí-los, o jogador recebe bonificações em dinheiro. Naves inimigas, porém, não tem tripulação visível. Os nós podem também ser lojas em que o jogador pode pagar com “Gold” adquirido em batalha, por aprimoramentos de armadura, armas e tripulação; reparar a nave; e recrutar ou reviver tripulantes.

Em cada setor se tem o nó inicial e o final, que permitirá ao jogador progredir para chegar nos setores seguintes, até o setor final do jogo. O setor final difere dos tipos dos demais, sendo definido como “Final Destiny”, que é quando ocorre o último confronto.

## 2. MODELO CONCEITUAL



Modelo 2.1 - Modelo Conceitual de um Banco de Dados

### 3. NORMALIZAÇÃO

#### 3.1. 0FN

```
Nó(id_Nó, Loja, Evento, {Nave}, Coordenada, Loja(id_Loja, Mensagem, ProdutosVendidos), Evento(id_Evento, Mensagem));  
Setor(id_Setor, {Nó}, id_TipoSetor, nome, NóFinal, NóInicial, NumSetor);  
Inimigo(id_Inimigo, id_nave, Recompensa, Nome);  
Nave(id_Nave, {Tripulação}, id_TipoNave, NomeModelo, Ataque, Armadura, Nome);  
Tripulante(id_Tripulante, id_Função, nome_função, Nome, Idade, Bio, Vida);  
Jogador(id_Jogador, id_nave, Nome, Nickname, Email, Senha);
```

*Tabela 3.1 - Modelo não normalizado*

#### 3.2. 1FN

Para Aplicar a 1FN precisamos eliminar os campos multivalorados, como {Tripulação} dentro Nave, {Nave} dentro de Nó e {Nó} dentro de Setor. Vale ressaltar que esses campos irão sumir das tabelas originais e serão criadas novas tabelas para eles.

```
Nó(id_Nó, id_setor, Coordenada, id_Loja, Nome, Mensagem, ProdutosVendidos, id_Evento, Mensagem);  
Setor(id_Setor, id_TipoSetor, nome, NóFinal, NóInicial, NumSetor);  
Inimigo(id_Inimigo, id_nave, Recompensa, Nome);  
Loja(id_Loja, Nome, Mensagem, ProdutosVendidos);  
Evento(id_Evento, Mensagem);  
Nave(id_Nave, id_nó, id_TipoNave, NomeModelo, Ataque, Armadura, Nome);  
Tripulante(id_Tripulante, id_nave, id_Função, nome_função, Nome, Idade, Bio, Vida);  
Jogador(id_Jogador, id_nave, Nome, Nickname, Email, Senha);
```

*Tabela 3.2 - Modelo na 1FN*

### 3.3. 2FN e 3FN

Para aplicar a 2FN é necessário que a tabela já esteja na 1FN e que todos os atributos não chave sejam dependentes da chave primária. A 3FN consiste em os campos que podem ser obtidos através de alguma operação matemática com outros campos pertencentes a mesma tabela. No modelo não existe esse tipo de campo, portanto o modelo, uma vez na 2FN, já está na FN3, como demonstrado na *Tabela 3.3*.

```
Loja(id_Loja, Nome, Mensagem, ProdutosVendidos);  
Evento(id_Evento, Mensagem)  
NósDeUmSetor(id_setor, id_nó);  
Nó(id_Nó, id_evento, id_Loja, Coordenada);  
Setor(id_Setor, id_NóFinal, id_NóInicial, NumSetor);  
TipoSetor(id_TipoSetor, nome);  
TipoSetorAss(id_tipoSetor, id_setor);  
TipoNave(id_TipoNave, NomeModelo, Ataque, Armadura);  
NaveNó(id_nó, id_nave);  
Nave(id_Nave, id_tiponave, Nome);  
Inimigo(id_Inimigo, Recompensa, Nome);  
InimigoNave(id_nave, id_inimigo);  
Jogador(id_Jogador, Nome, Nickname, Email, Senha);  
NaveJogador(id_nave, id_inimigo);  
Tripulante(id_Tripulante, id_Função, Nome, Idade, Bio, Vida);  
Função(id_Função, nome_função);
```

*Tabela 3.3 - Modelo Normalizado de acordo com a FN3, e consequentemente, com a FN2.*

## 4. MODELO ENTIDADE-RELACIONAMENTO

