

# Projeto Integrador II – RPGnow

## *Documento Análise e Design*

Repositório: <https://github.com/volponi121/projeto-integrador-2019.git>

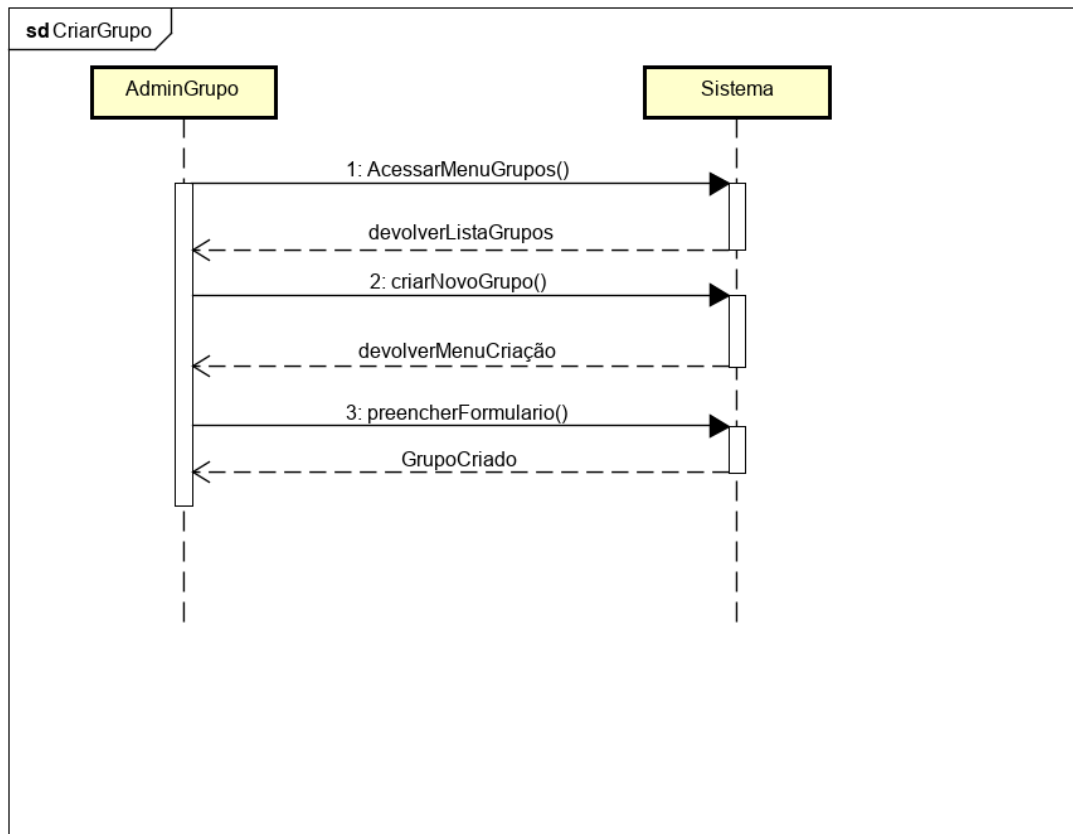
### **Integrantes:**

- Elielson Cardoso da Silva – R.A.: 1722105-2
- Fabio Augusto Volponi de Souza – R.A: 1702511-2
- Jéssica Santiago Ferreira – R.A.: 1735337-2
- José Carlos Soares Junior – R.A.: 1701479-2
- Matheus Nicolielo Serafim dos Santos – R.A: 1722071-2

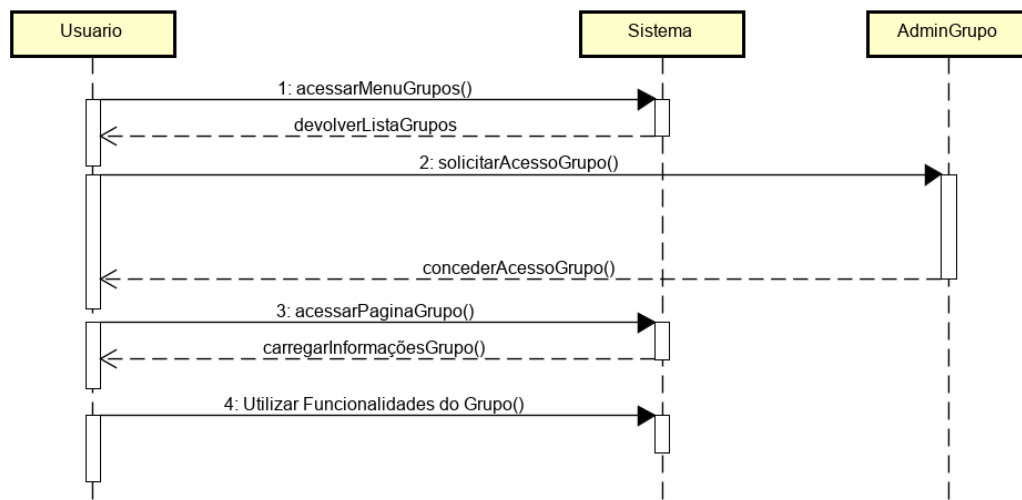
## 1. Arquitetura do Sistema

Vamos utilizar a arquitetura serverless, assim como API Rest. Para o deploy e controle utilizaremos o AWS(Amazon Web Services), como será utilizado o AWS as conexões serão feitas pela própria plataforma, aumentando as instâncias de acordo com a necessidade.

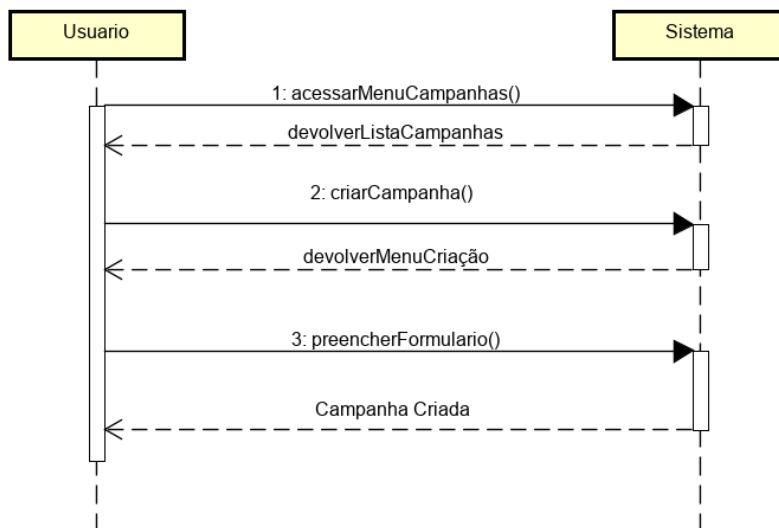
## 2. Diagrama de Sequência



**sd** AcessarGrupo

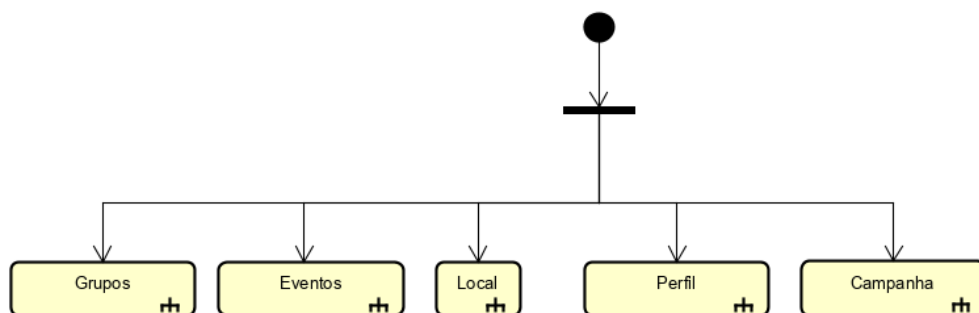
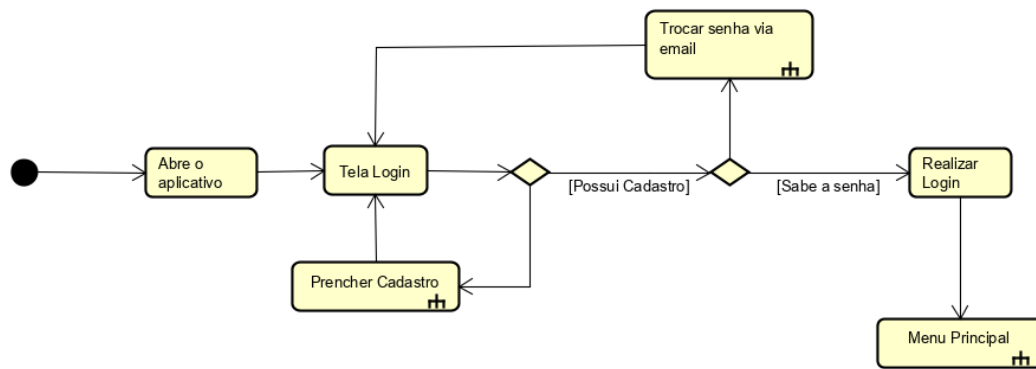


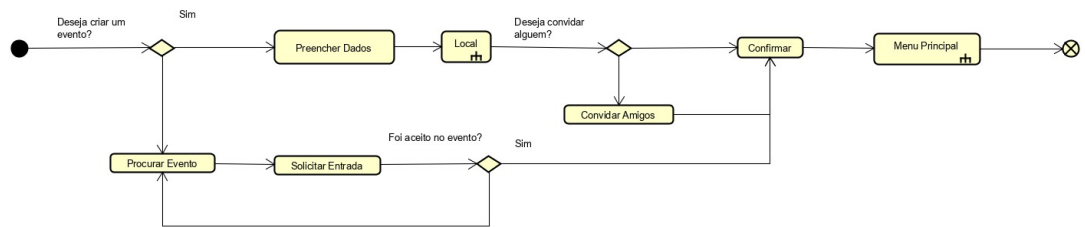
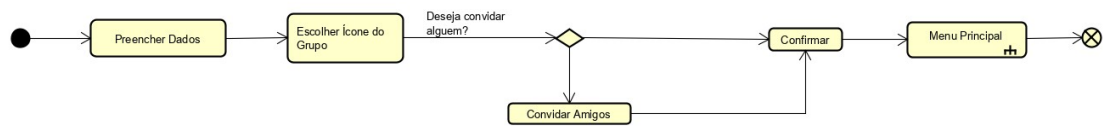
**sd** CriarCampanha

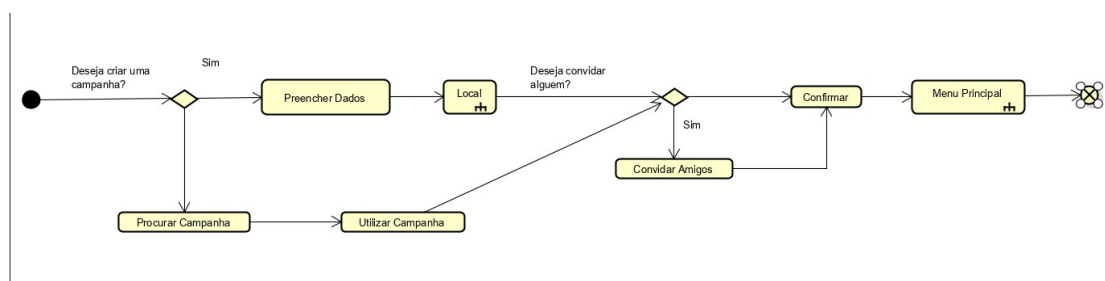


### 3. Diagrama de Atividades

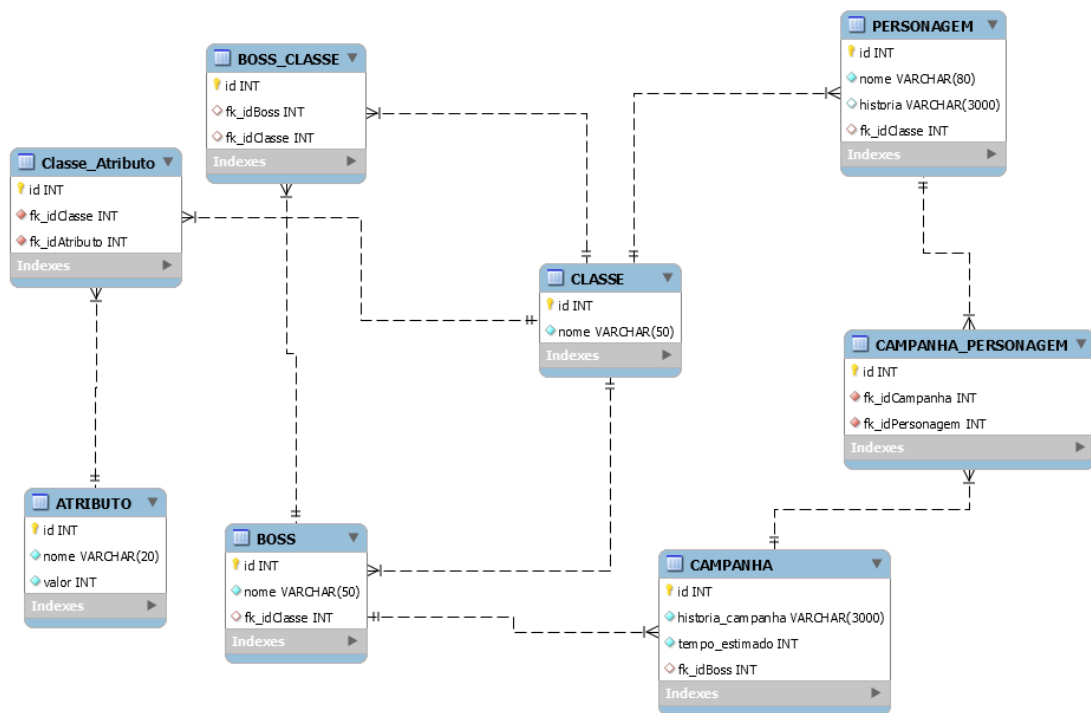
---

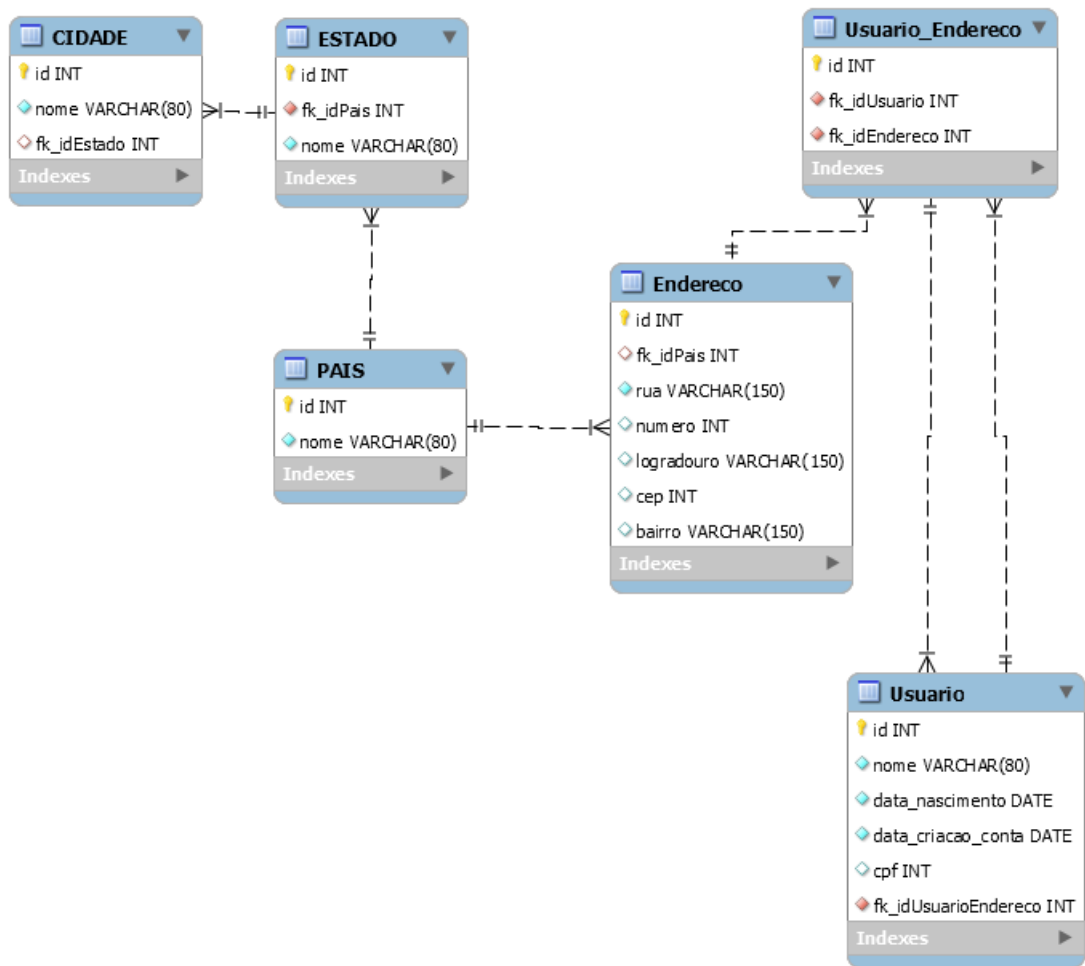






## 4. Modelo Lógico da Base de Dados







## 5. Criação Física do Modelo de Dados

```
/*----- Criando Usuário e suas Ligações -----*/
CREATE TABLE PAIS (
  id int not null,
  nome varchar(80) not null
);

ALTER TABLE PAIS ADD CONSTRAINT PK_PAIS PRIMARY KEY (ID);

CREATE TABLE ESTADO (
  id int not null,
  fk_idPais int not null,
  nome varchar(80) not null
);

ALTER TABLE ESTADO ADD CONSTRAINT PK_ESTADO PRIMARY KEY (ID);
ALTER TABLE ESTADO ADD CONSTRAINT FK_PAIS_ESTADO FOREIGN KEY (fk_idPais) REFERENCES PAIS(id);

CREATE TABLE CIDADE (
  ALTER TABLE Usuario_Endereco ADD CONSTRAINT PK_Usuario_Endereco PRIMARY KEY (ID);

  ALTER TABLE USUARIO ADD CONSTRAINT FK_USUARIOEND_ENDereco FOREIGN KEY (fk_idUsuarioEndereco) REFERENCES Usuario_Endereco(id);
  ALTER TABLE Usuario_Endereco ADD CONSTRAINT FK_USUARIO_USUARIOEND FOREIGN KEY (fk_idUsuario) REFERENCES USUARIO(id);
  ALTER TABLE Usuario_Endereco ADD CONSTRAINT FK_ENDereco_USUARIOEND FOREIGN KEY (fk_idEndereco) REFERENCES ENDereco(id);

/*----- Criando Campanha e personagens -----*/

= CREATE TABLE CLASSE (
  id int not null,
  nome varchar(50) not null
);

ALTER TABLE CLASSE ADD CONSTRAINT PK_CLASSE PRIMARY KEY (ID);

= CREATE TABLE BOSS (
  id int not null,
  nome varchar(50) not null,
  fk_idClasse int
);

ALTER TABLE BOSS ADD CONSTRAINT PK_BOSS PRIMARY KEY (ID);
ALTER TABLE BOSS ADD CONSTRAINT FK_CLASSE_BOSS FOREIGN KEY (fk_idClasse) REFERENCES CLASSE(id);

= CREATE TABLE CAMPANHA (
  id int not null,
  historia_campanha varchar(3000) not null,
  tempo_estimado int not null,
  fk_idBoss int
);

ALTER TABLE CAMPANHA ADD CONSTRAINT PK_CAMPANHA PRIMARY KEY (ID);
ALTER TABLE CAMPANHA ADD CONSTRAINT FK_BOSS_CAMPANHA FOREIGN KEY (fk_idBoss) REFERENCES BOSS(id);

= create table BOSS_CLASSE(
  id int not null,
  fk_idBoss int,
  fk_idClasse int
);

ALTER TABLE BOSS_CLASSE ADD CONSTRAINT PK_BOSS_CLASSE PRIMARY KEY (ID);
ALTER TABLE BOSS_CLASSE ADD CONSTRAINT FK_BOSS_BOSSCLASSE FOREIGN KEY (fk_idBoss) REFERENCES BOSS(id);
ALTER TABLE BOSS_CLASSE ADD CONSTRAINT FK_CLASSE_BOSSCLASSE FOREIGN KEY (fk_idClasse) REFERENCES CLASSE(id);

= CREATE TABLE PERSONAGEM (
  id int not null,
  nome varchar(80) not null,
  historia varchar(3000),
  fk_idClasse int
);

ALTER TABLE PERSONAGEM ADD CONSTRAINT FK_CLASSE_PERSONAGEM FOREIGN KEY (fk_idClasse) REFERENCES CLASSE(id);

CREATE TABLE ATRIBUTO (
  id int not null,
  nome varchar(20) not null,
  valor int not null
);

ALTER TABLE ATRIBUTO ADD CONSTRAINT PK_ATRIBUTO PRIMARY KEY (ID);

create table Classe_Atributo(
  id int not null,
  fk_idClasse int not null,
  fk_idAtributo int not null
);

ALTER TABLE Classe_Atributo ADD CONSTRAINT PK_Classe_Atributo PRIMARY KEY (ID);
ALTER TABLE Classe_Atributo ADD CONSTRAINT FK_ATRIBUTO_Classe_Atributo FOREIGN KEY (fk_idAtributo) REFERENCES ATRIBUTO(id);
ALTER TABLE Classe_Atributo ADD CONSTRAINT FK_CLASSE_Classe_Atributo FOREIGN KEY (fk_idClasse) REFERENCES CLASSE(id);

CREATE TABLE CAMPANHA_PERSONAGEM (
  id int not null,
  fk_idCampanha int not null,
  fk_idPersonagem int not null
);

ALTER TABLE CAMPANHA_PERSONAGEM ADD CONSTRAINT PK_CAMPANHA_PERSONAGEM PRIMARY KEY (ID);
ALTER TABLE CAMPANHA_PERSONAGEM ADD CONSTRAINT FK_CAMPANHA_CAMPANHA_PERSONAGEM FOREIGN KEY (fk_idCampanha) REFERENCES CAMPANHA(id);
ALTER TABLE CAMPANHA_PERSONAGEM ADD CONSTRAINT FK_PERSONAGEM_CAMPANHA_PERSONAGEM FOREIGN KEY (fk_idPersonagem) REFERENCES PERSONAGEM(id);
```

## 6. Ambiente de Desenvolvimento

NodeJs no Back-end com React Native no Front-end, banco de dados utilizaremos PostgreSQL, Git e GitHub para o versionamento de código, Docker para utilizar a imagem do banco de Dados PostgreSQL, Postman para a requisição da API REST.

## 7. Sistemas e componentes externos utilizados

VsCode(Editor de texto <https://code.visualstudio.com/>)

Git (Versionador de código <https://git-scm.com/>)

Docker(Para quem usa Windows <https://docs.docker.com/docker-for-windows/>)

PgAdmin3 para utilizar o banco(<https://www.pgadmin.org/>)

nodeJs(<https://nodejs.org/en/> Ultima versão LTS)

Postman(<https://www.getpostman.com/>)

Discord(<https://discordapp.com/>)

Trello(<https://trello.com/>)