

Modelagem de Dados

Os principais componentes do sistema são as **startups**. Cada **startup** recebe possui um nome, um slogan e um ano de fundação. Além disso, a cada **startup** está associada uma pontuação. A pontuação das **startups** depende:

1. da sua criação, quando toda startup recebe +70 pontos;
2. das batalhas vencidas; com +30 pontos por vitória; e
3. dos eventos acumulados durante batalhas, com valores diversos.

As **startups** competem entre si em **batalhas**. Durante uma batalha, o usuário/juiz pode registrar **eventos** a um ou a outro dos competidores. Cada evento possui um valor de pontos específico:

- "pitch convincente": +6 pontos;
- "produto com bugs": -4 pontos;
- "boa tração de usuários": +3 pontos;
- "investidor irritado": -6 pontos; e
- "fake news no pitch": -8 pontos

Somente é possível registrar, por batalha, um evento de determinado tipo. É possível registrar vários eventos para um mesmo participante, desde que os eventos sejam de tipos distintos.

Um **torneio** é um conjunto sequencial de batalhas. Cada torneio terá sua coleção de batalhas, bem como o **round** atual e uma **flag** de status para sabermos se o torneio está concluído, em andamento ou se ainda não começou. No banco de dados, o torneio também mantém armazenado o identificador da startup vencedora. De forma análoga, toda **batalha** também tem um status (**"not_started"**, **"completed"**) e, caso tenha sido concluída, armazena o identificador do vencedor.

Startup

PYTHON

```
class Startup(db.Model):
    id = db.Column(db.Integer, primary_key=True, autoincrement=True);
    name = db.Column(db.String(255), nullable=False);
    slogan = db.Column(db.String(255), nullable=False);
    year = db.Column(db.Integer, nullable=False);
    points = db.Column(db.Integer, nullable=False);
```

Batalhas (Battle)

PYTHON

```
class Battle(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    tournament_id = db.Column(db.Integer, db.ForeignKey('tournament.id'),
    nullable=False)
    round_number = db.Column(db.Integer, nullable=False)
    startup_a_id = db.Column(db.Integer, db.ForeignKey('startup.id'), nullable=False)
    startup_b_id = db.Column(db.Integer, db.ForeignKey('startup.id'), nullable=False)
    status = db.Column(db.String(20), nullable=False)
    winner_id = db.Column(db.Integer, db.ForeignKey('startup.id'), nullable=True)
```

Eventos (Event)

PYTHON

```
class Event(db.Model):
    id = db.Column(db.Integer, primary_key=True);
    user_id = db.Column(db.Integer, nullable=False);
    startup_id = db.Column(db.Integer, nullable=False);
    event_type = db.Column(db.String(20), nullable=False);
    battle_id = db.Column(db.Integer, db.ForeignKey('battle.id'), nullable=False)
```

Torneios (Tournament)

PYTHON

```
class Tournament(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(255), nullable=False)
    status = db.Column(db.String(20), nullable=False)
    current_round = db.Column(db.Integer, nullable=False)
    battles = db.relationship('Battle', backref='tournament', lazy=True)
```
