

Django para startups

O caso da amazonense Jukx App

Quem sou eu

FELIPE CARVALHO

- Graduando de Sistemas de Informação (Ufam);
 - Experiência profissional como desenvolvedor desde 2013;
 - Formado em Jornalismo; Trabalhei na área de 2006 até 2013;
 - Membro do PyNorte e da Comunidade Jaraqui Valley;
 - Já teve duas startups falidas, mas está persistindo na terceira!
-

Jukx

PORQUE A MÚSICA É IMPORTANTE

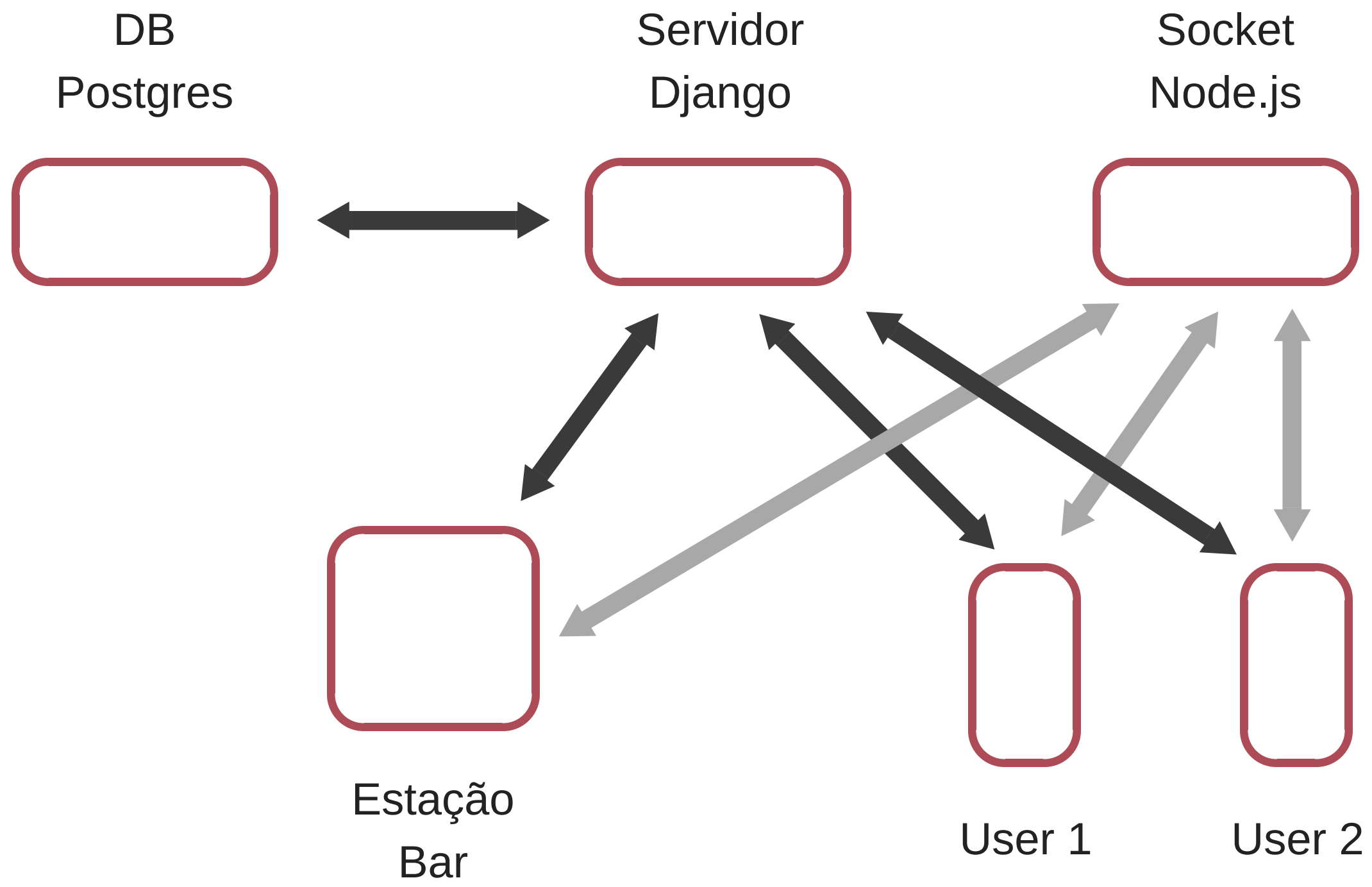
- O estabelecimento tinha uma playlist personalizada e sempre atualizada;
 - O cliente poderia escolher a música que desejava ouvir pelo app;
 - Estabelecimento gera renda extra com a venda de fichas virtuais para os usuários;
-

+700 usuários

- Duas versões lançadas;
 - Web + app Android + app desktop;
 - Finalistas do InovAtiva Brasil 2016.2;
-

Tecnologias

- Django + DRF
 - Bibliotecas Python
 - AngularJS
 - Socket.IO
 - Electron
 - Postgres
-



Por que Django?

UMA STARTUP PRECISA TESTAR HIPÓTESES E CRESCER O MAIS RÁPIDO POSSÍVEL

- Desenvolvimento rápido;
 - Muitos pacotes disponíveis tanto no Django quanto no Python;
 - Comunidade participante e engajada;
-

models/station.py

```
1 |from django.db import models
2 |from django.contrib.auth.models import User
3
4
5 |class JukxStation(models.Model):
6
7 |    class Meta:
8 |        app_label = 'version2'
9
10 |    TYPE = (
11 |        ('0', u'Pública'),
12 |        ('1', u'Particular'),
13 |        ('2', u'Secreta'),
14 |    )
15
16 |    title = models.CharField(max_length=32, unique=False)
17 |    description = models.CharField(max_length=128, null=True, unique=False)
18 |    allow = models.BooleanField(default=False)
19 |    picture = models.CharField(max_length=32, unique=True, null=True)
20 |    follower = models.ManyToManyField(User, blank=True, through='Followership', related_name='member')
21 |    owner = models.ManyToManyField(User, through='JukxOwnership', related_name='owner')
22 |    created = models.DateTimeField(auto_now_add=True, editable=False)
23 |    is_active = models.BooleanField(default=True)
24 |    privacy = models.CharField(max_length=1, choices=TYPE)
25 |    location = models.CharField(max_length=62, null=True)
26 |    url = models.CharField(max_length=18, unique=True, null=True)
27 |    visits = models.IntegerField(null=True)
28 |    last_update = models.DateTimeField(null=True)
29 |    last_propaganda = models.ForeignKey('Propaganda', null=True)
30 |    propaganda_view = models.DateTimeField(null=True)
31
32 |    def __str__(self):
33 |        return u'{0}'.format(self.title)
34
35 |    def get_absolute_url(self):
36 |        return "/s/%s/" % self.url
```


views/station.py

```
1 from version2.serializers.station import StationSerializer
2 from rest_framework import status, permissions, generics
3 from version2.models.station import JukxStation
4
5 class StationList(generics.ListAPIView):
6     """
7     List all stations.
8     """
9
10
11     queryset = JukxStation.objects.filter(is_active=True).order_by('-last_update', '-visits')
12     serializer_class = StationSerializer
13     permission_classes = (permissions.IsAuthenticatedOrReadOnly,)
14
15     def perform_create(self, serializer):
16         admin = self.request.user
17         serializer.save(admin=admin)
```

serializers/station.py

```
1  # -*- coding:utf-8 -*-
2  from django.db import transaction
3  from version2.models.station import JukxStation, JukxOwnership
4  from version2.models.lista import NowPlaying, MainList
5  from version2.serializers.address import AddressSerializer
6  from rest_framework import serializers
7
8
9  class StationSerializer(serializers.ModelSerializer):
10
11     address = AddressSerializer(read_only=True)
12     now_playing = serializers.SerializerMethodField()
13     is_owner = serializers.SerializerMethodField()
14
15     def get_now_playing(self, obj):
16         return NowPlaying().now_playing(self.context['request'], obj)
17
18     def get_is_owner(self, obj):
19         if self.context:
20             user = self.context['request'].user
21             if not user.is_anonymous():
22                 return JukxOwnership().is_owner(user, obj)
23         return False
24
25     class Meta:
26         model = JukxStation
27         exclude = ('owner', 'created', 'follower', 'last_update', 'is_active', 'visits', 'last_propaganda',
28
29
30     def create(self, validated_data):
31         """
32         Create and return a new Station instance, given the validated data.
33         """
34
35         return Station.objects.create(**validated_data)
```

urls.py

```
1 from django.conf.urls import include, url
2 from .views import station_mgmt
3
4 urlpatterns = [
5     .....
6     url(r'^stations/$', station_mgmt.StationList.as_view(), name='station_list'),
7
8 ]
```

Outras libs

- compressor;
 - python-decouple;
 - pytube;
 - ffmpeg;
-

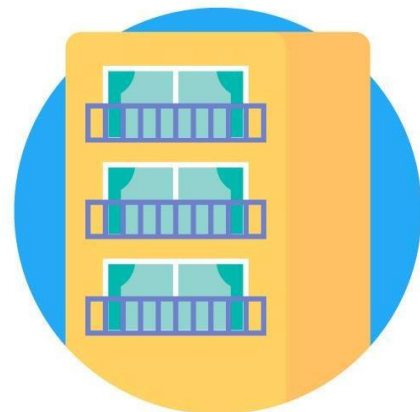
Nem tudo são flores...

- Problemas com o Tornado;
 - Sem muita adesão pela comunidade;
 - Documentação oficial muito densa;
 - Pouco suporte
-

Django Manaus



Mapsta



Communny



Conatus



Indikko



Obrigado

felipe.carvalho07@gmail.com

(92) 98420-1817

twitter.com/_07carvalho
