



UFC



# Systagram - Uma Aplicação Web para Gerenciar Posts de Fotos, implementada sobre os serviços da AWS

Desenvolvimento de Software para Nuvem - 2019.1 - UFC/MDCC

Professores: Fernando Trinta e Paulo Rego

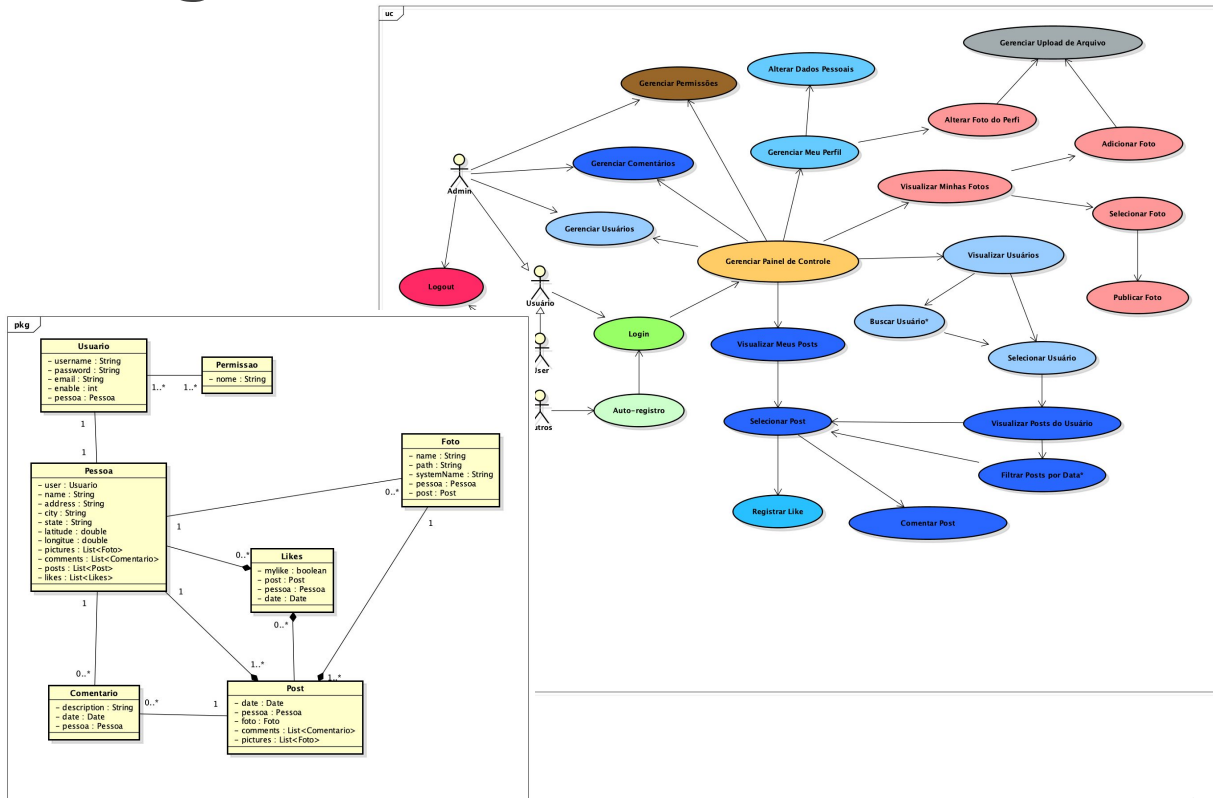
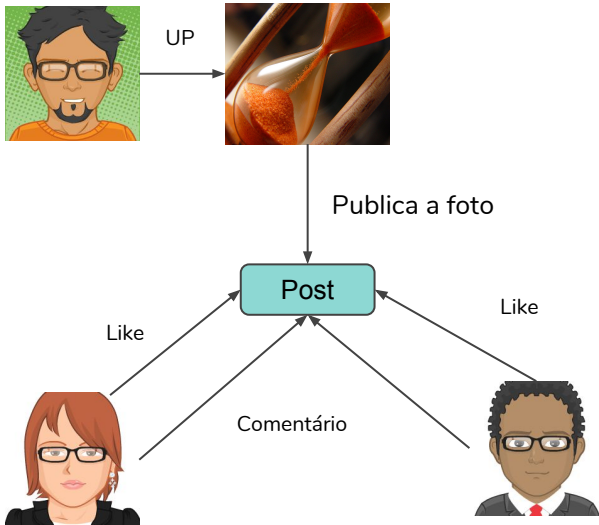
Alunos: Armando Soares e Juarez Meneses



# Agenda

- ❑ O que é Systagram?
- ❑ Requisitos Funcionais
- ❑ Requisitos Não Funcionais
- ❑ Arquitetura
  - ❑ Para uma instância
  - ❑ Balanceamento de Carga e Elasticidade
- ❑ Protótipo da Aplicação Web
- ❑ Considerações Finais

# O que é o Systagram?





# Requisitos Funcionais - Histórias do Usuário

HU001 - O usuário se registra na Aplicação

HU002 - O usuário faz login na Aplicação

HU003 - O usuário faz logout da Aplicação

HU004 - O usuário gerencia seu perfil

HU005 - O usuário visualiza seu painel de controle

HU006 - O usuário visualiza outros usuários

HU007 - O usuário busca um usuário

HU008 - O usuário seleciona um usuário

HU009 - O usuário visualiza posts do usuário selecionado

HU010 - O usuário filtra posts de por data

HU011 - O usuário seleciona um post

HU012 - O usuário registra um like em post selecionado

HU013 - O usuário registra um comentário em post selecionado

HU014 - O usuário visualiza suas fotos

HU015 - O usuário faz upload de foto

HU016 - O usuário seleciona uma foto

HU017 - O usuário publica uma foto

HU018 - O usuário seleciona outro usuário

HU019 - O usuário visualiza perfil de outro usuário



# Requisitos não funcionais

RNF001 - As informações dos usuários devem ser gravadas em uma instância de banco de dados relacional, criada pelo serviço Amazon RDS;

RNF002 - As fotos que serão enviadas devem ser armazenadas utilizando o serviço Amazon S3;

RNF003 - As curtidas que as imagens possuem deverão ser salvas usando o Amazon DynamoDB;

RNF004 - Deverá existir um balanceador de carga para gerenciar as requisições dos usuários

RNF005 - A aplicação deverá suportar um esquema de elasticidade horizontal. O controle de elasticidade será controlado via análise de consumo de CPU, quando passar de 80% de consumo de CPU deve-se criar uma nova instância, quando for menor que 10% deve remover uma instância. O esquema de elasticidade deverá variar entre 2 e 4 instâncias.



# Estrutura da Solução

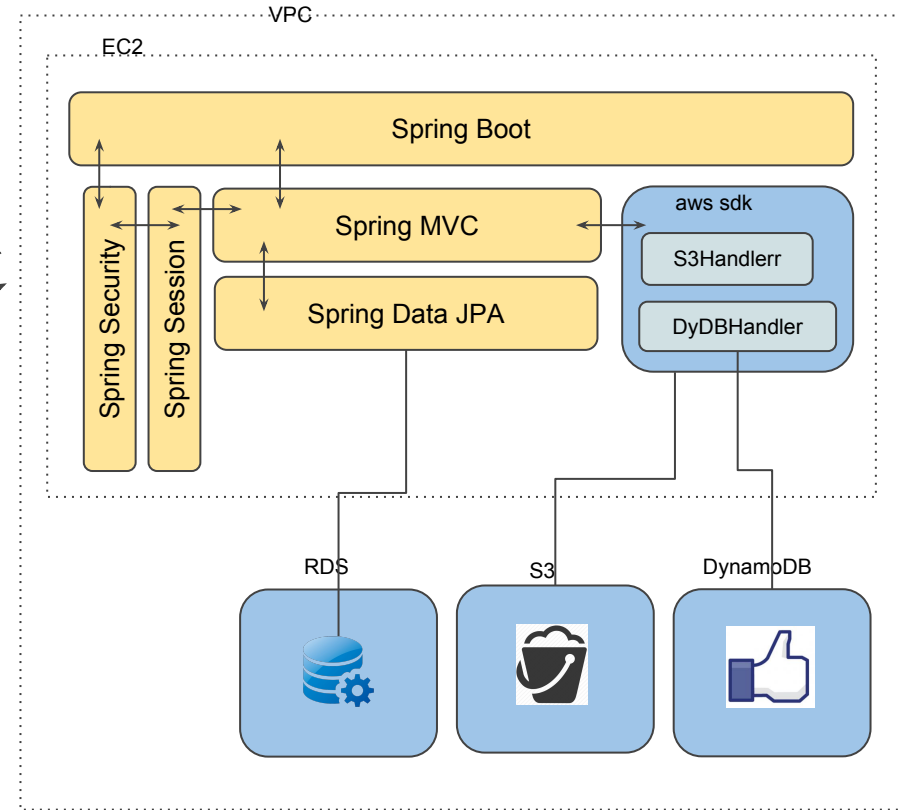
Solução implementada sobre a IaaS da AWS

- Servidor de Aplicação no EC2
  - Conjunto de serviços que serão consumidos pela aplicação web..
  - Componentes que se conectam com outros serviços da AWS
- RDS
  - *Relation Database Service*
- S3
  - *Simple Storage Service*
- DynamoDB
  - *NoSQL Database Service*
- LoadBalance
  - Balanceamento de carga entre as instâncias ativas
- Auto Scaling
  - Esquema de elasticidade horizontal baseado em consumo de CPU das instâncias ativas

# Arquitetura (Uma instância)

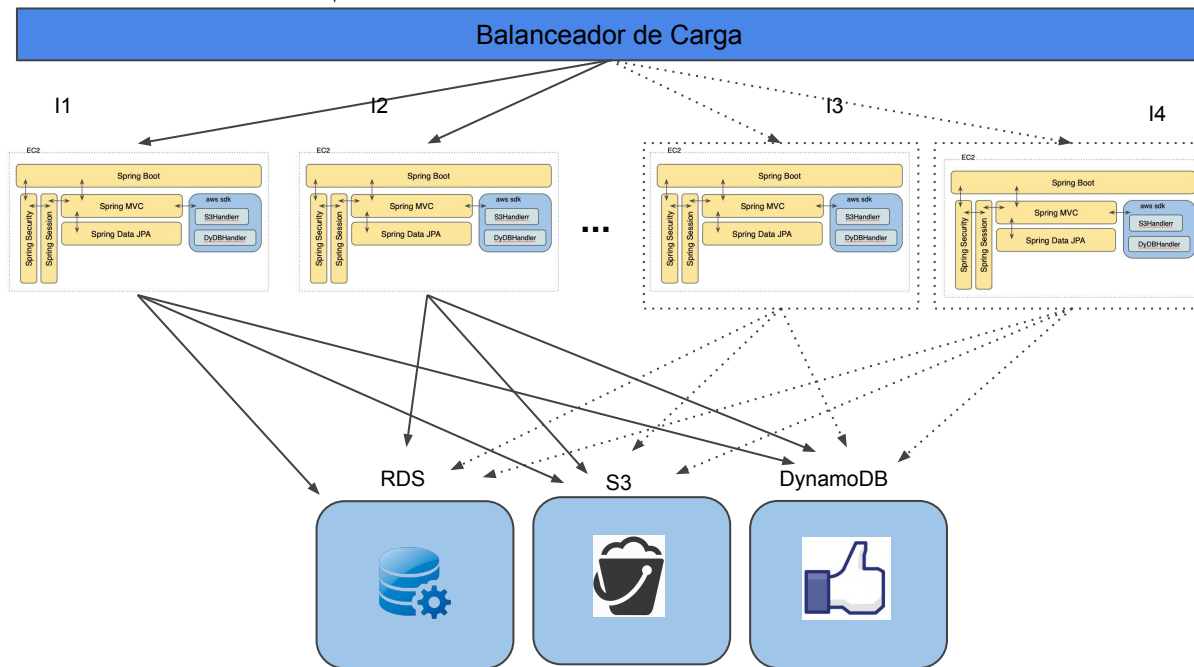


Spring Data JPA



# Esquema de Balanceamento de Carga

AWS



Cliente 1

Cliente 2

Cliente N

Cientes

http

AWS Cloud



# Esquema de Balanceamento de Carga

Name

DNS name

State

VPC ID

Availability Zones

Type

Created At

MyLoadBalancer2

MyLoadBalancer2-1116216541.us-east-1.elb.amazonaws.com

vpc-1cac2866

us-east-1e

classic

April 3, 2019 at 10:05:06 PM UT

Systagram3LoadBalance

Systagram3LoadBalance-796544882.us-east-1.elb.amazonaws.com

vpc-1cac2866

us-east-1a, us-east-1b

classic

April 4, 2019 at 2:01:14 PM UT

Load balancer: Systagram3LoadBalance

Description

Instances

Health check

Listeners

Monitoring

Tags

Migration

Basic Configuration

Name

Systagram3LoadBalance

Creation time

April 4, 2019 at 2:01:14 PM UTC-3

DNS name

Systagram3LoadBalance-796544882.us-east-1.elb.amazonaws.com  
(A Record)

Hosted zone

Z35SXDOTRQ7X7K

Type

Classic (Migrate Now)

Status

2 of 2 instances in service

Scheme

internet-facing

Availability Zones

subnet-82a69ec8 - us-east-1a,  
subnet-dbf8e87 - us-east-1b

Port Configuration

Port Configuration

80 (HTTP) forwarding to 80 (HTTP)

Stickiness: LBCookieStickinessPolicy, expirationPeriod=3

Edit stickiness

Security

Source Security Group

sg-02bfcf88f954dbb2b, AutoScaling-Security-Group-1  
AutoScaling-Security-Group-1 (2019-04-03 22:10:42)

Edit security groups

Attributes

Idle timeout

60 seconds

Edit idle timeout

Name

DNS name

State

VPC ID

Availability Zones

MyLoadBalancer2

MyLoadBalancer2-1116216541.us-east-1.elb.amazonaws.com

vpc-1cac2866

us-east-1e

Systagram3LoadBalance

Systagram3LoadBalance-796544882.us-east-1.elb.amazonaws.com

vpc-1cac2866

us-east-1a, us-east-1b

Load balancer: Systagram3LoadBalance

Description

Instances

Health check

Listeners

Monitoring

Tags

Migration

Connection Draining: Enabled, 300 seconds (Edit)

Edit Instances

Instance ID

Name

Availability Zone

Status

Actions

i-036088a48035588ca

us-east-1a

InService ⓘ

Remove from Load Balancer

i-0886e0150b7fd2813

us-east-1b

InService ⓘ

Remove from Load Balancer

Edit Availability Zones

Availability Zone

Subnet ID

Subnet CIDR

Instance Count

Healthy?

Actions

us-east-1a

subnet-82a69ec8

172.31.16.0/20

1

Yes

Remove from Load Balancer

us-east-1b

subnet-dbf8e87

172.31.32.0/20

1

Yes

Remove from Load Balancer

# Esquema de Auto Scaling

Name	AMI ID	Instance Type	Spot Price	Creation Time
Systagram3LaunchConfiguration	ami-0f84cf36c...	t2.micro		April 4, 2019 at 2:02:41 PM UTC-3

Launch Configuration: Systagram3LaunchConfiguration

## Details

Copy launch configuration

AMI ID  
ami-0f84cf36c615d3dbb

IAM Instance Profile

Key Name  
desenvolvimento2

EBS Optimized  
false

Spot Price

RAM Disk ID

User data

Instance Type  
t2.micro

Kernel ID

Monitoring  
false

Security Groups  
sg-05f46657fe72f64e7

Creation Time  
Thu Apr 04 14:02:41 GMT-300 2019

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
Systagram3AutoScalingGroup	Systagram3LaunchConf...	2	2	2	4	us-east-1a, us-east-1b	300	300

Auto Scaling Group: Systagram3AutoScalingGroup

Details Activity History Scaling Policies Instances Monitoring Notifications Tags Scheduled Actions Lifecycle Hooks

Launch Configuration ⓘ Systagram3LaunchConfiguration

Desired Capacity ⓘ 2

Min ⓘ 2

Max ⓘ 4

Availability Zone(s) ⓘ us-east-1a, us-east-1b

Subnet(s) ⓘ subnet-82a69ec8,subnet-dbf6e87

Classic Load Balancers ⓘ Systagram3LoadBalance

Target Groups ⓘ

Health Check Type ⓘ EC2

Health Check Grace Period ⓘ 300

Instance Protection ⓘ

Termination Policies ⓘ Default

Suspended Processes ⓘ

Placement Groups ⓘ

Default Cooldown ⓘ 300

Enabled Metrics ⓘ

Creation Time ⓘ Thu Apr 04 14:04:51 GMT-300 2019

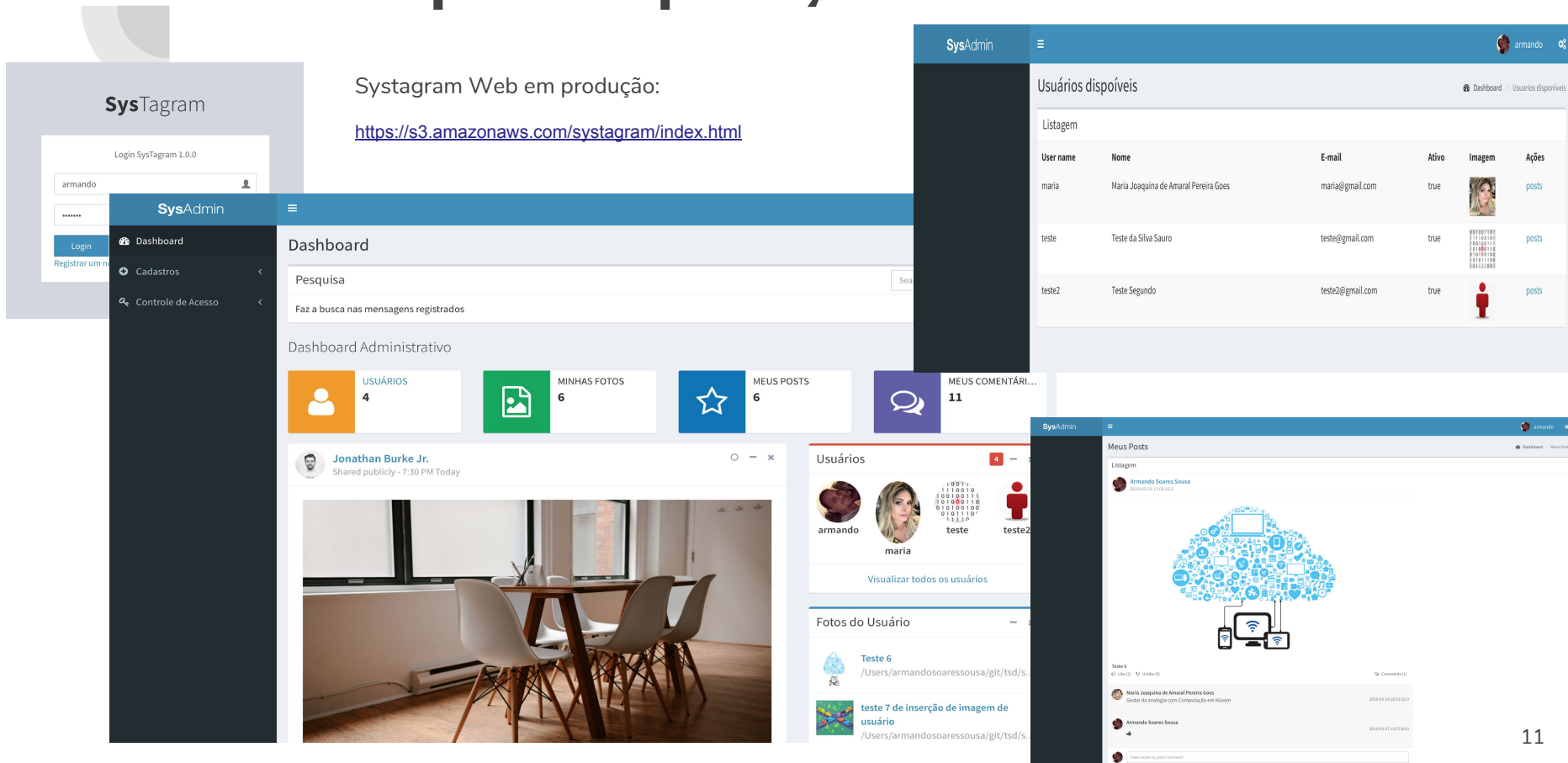
Service-Linked Role ⓘ  
arn:aws:iam::405802614091:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling

Edit

# Protótipo da Aplicação Web

Systagram Web em produção:

<https://s3.amazonaws.com/systagram/index.html>





## Considerações Finais

- Criar aplicações para a nuvem não é trivial.
- Para criar aplicações na nuvem, no esquema IaaS, é preciso ter bons conhecimentos de infraestrutura de rede e internet disponível.
  - Foi necessário trabalhar com vários frameworks para o desenvolvimento da aplicação e usar uma API específica da Amazon AWS para se comunicar com os outros serviços como o S3 e DynamoDB.
  - É preciso um bom conhecimento de redes de computadores para fazer as devidas configurações dos serviços com relação a segurança, portas, esquema de balanceamento e elasticidade.



# Referências

Repositório de código do Systagram App Web

<https://github.com/topicos-sistemas-distribuidos/systagram>



**Dúvidas?**





# Contato



**Armando Soares Sousa**

[armando@ufpi.edu.br](mailto:armando@ufpi.edu.br)

**Juarez L. Meneses Filho**

[juarezmeneses@great.ufc.br](mailto:juarezmeneses@great.ufc.br)

<http://www.mdcc.ufc.br>

Mestrado e Doutorado DC/UFC