

SCAPS

Sistema de Controle de Acesso de Pessoas em um Shopping

Bruna Michelly de O. S. Cordeiro
Thalia Santos de Santana

2020

SCAPS



- O trabalho consiste no desenvolvimento de um *Sistema de Controle de Acesso de Pessoas em um Shopping* - SCAPS;
- O código fonte, imagens e relatórios estão disponíveis em:
<https://github.com/brunacordeiro/SCAPS>
- Quatro etapas foram definidas:
 - Núcleo funcional do sistema composto pelos componentes e suas interações;
 - Coordenação, consenso e controle de concorrência;
 - Tolerância a falhas e replicação (confiabilidade, disponibilidade, performance);
 - Aspectos de segurança da interação entre os componentes e apresentação final.

Cenário

Passeio das Águas Shopping

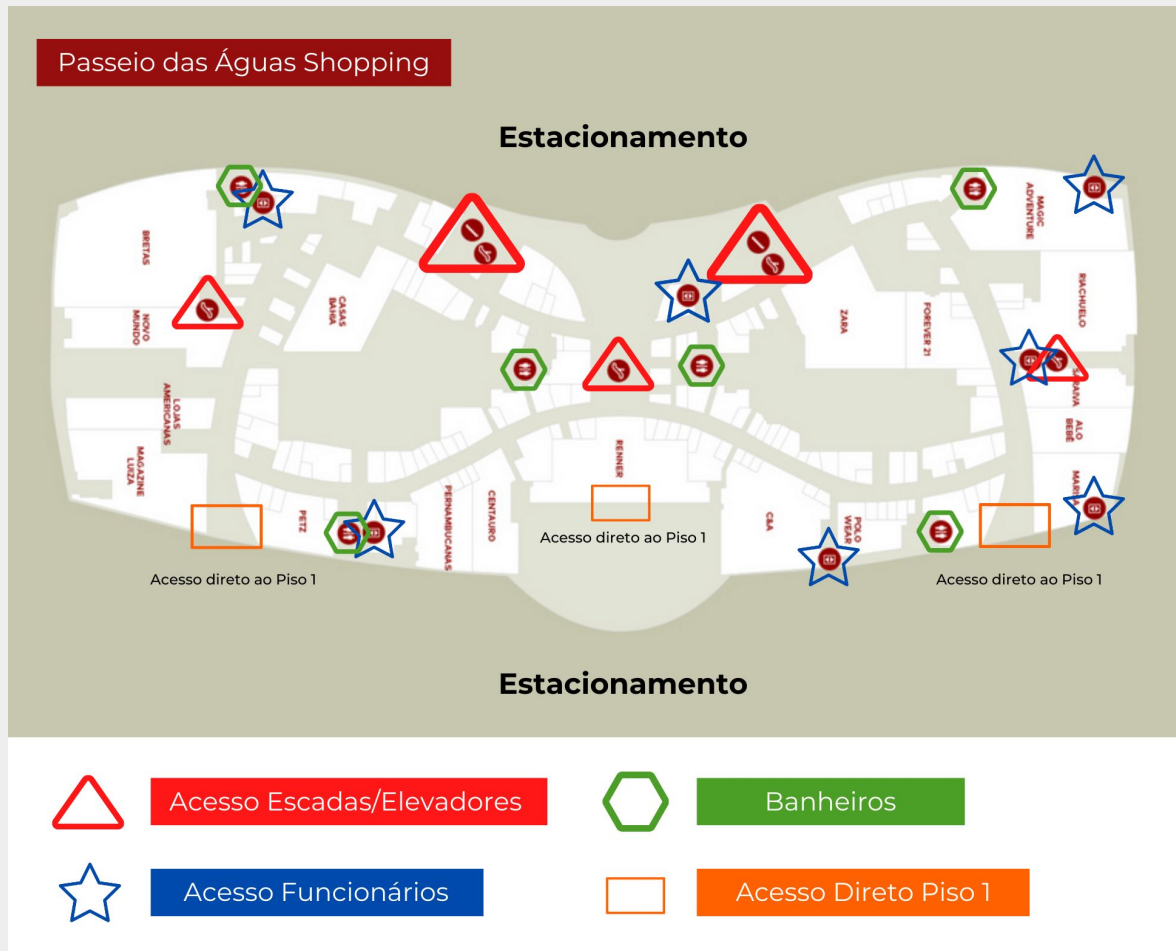
- 78 mil m²
- 283 lojas
- 1 hipermercado
- 7 salas de cinema



Cenário

Passeio das Águas Shopping

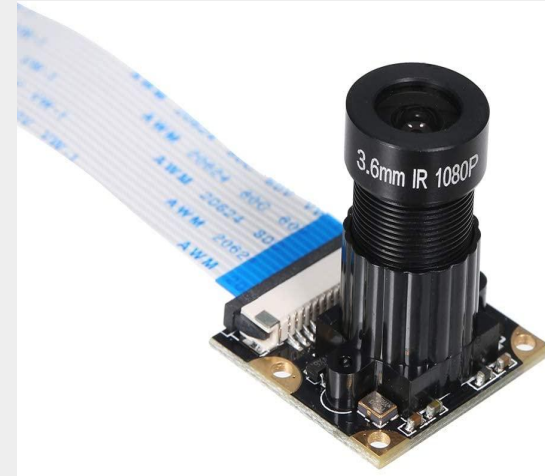
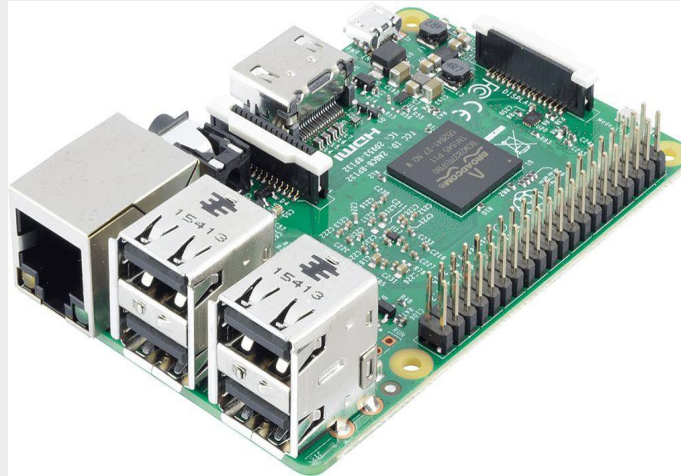
- 78 mil m²
- 283 lojas
- 1 hipermercado
- 7 salas de cinema



Dispositivo

Passeio das Águas Shopping

- Raspberry Pi
- Câmera ângulo 60°





Etapas



Etapa 1

Núcleo funcional do sistema composto pelos componentes e suas interações

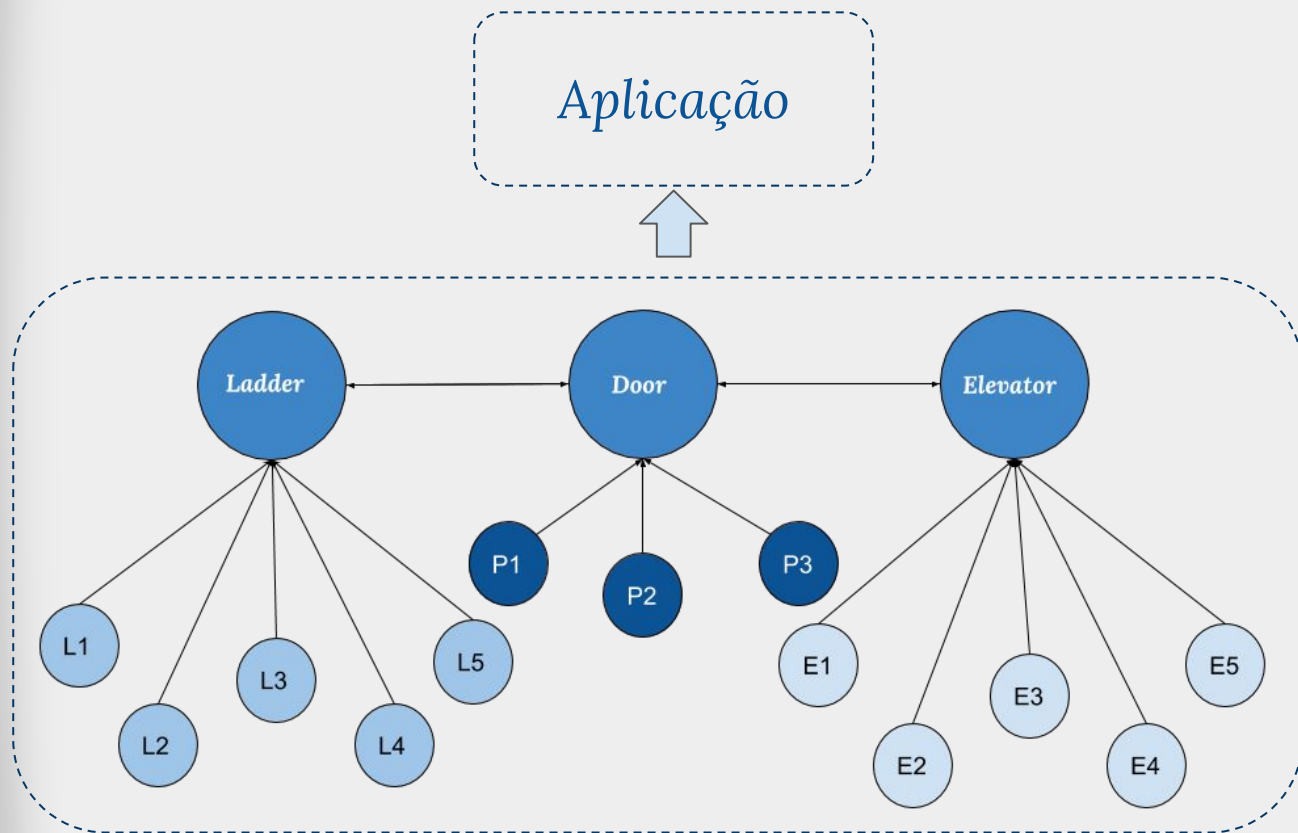
- *Middleware* RabbitMQ;
- Protocolo AMQP;
- Padrão *publish/subscribe*;
- Dados de entrada sendo as informações das câmeras (de modo simulado);
- Sequência de números binários, “0” para saída e “1” para entrada;
- Linguagem de programação Python, versão 3;
- Elementos:
 - Portas;
 - Escadas;
 - Elevadores.

Arquitetura

SCAPS

Acesso ao primeiro piso

- 3 portas
- 5 escadas
- 5 elevadores

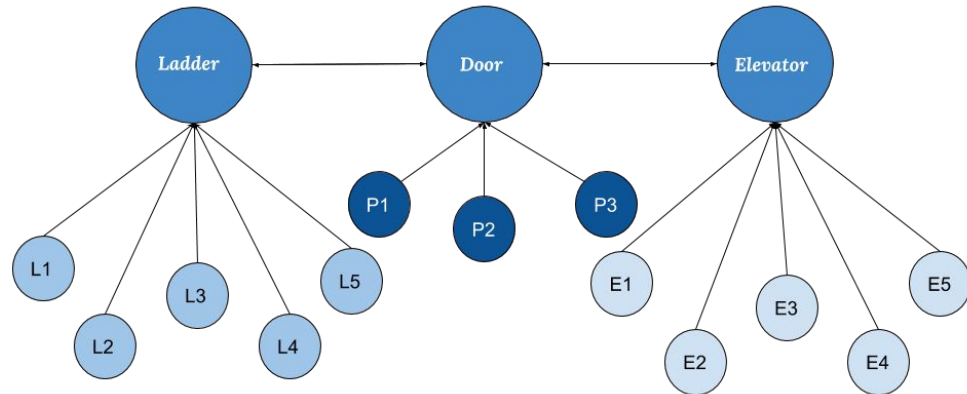




Etapa 2

Coordenação, consenso e controle de concorrência

- Manter consistência e coerência do estado de todos os componentes do sistema;
- **Senders**
 - SenderDoor1, SenderDoor2, SenderDoor3;
 - SenderElevator1, SenderElevator2, SenderElevator3, SenderElevator4 e SenderElevator5;
 - SenderLadder1, SenderLadder2, SenderLadder3, SenderLadder4 e SenderLadder5;
- **Consumers**
 - ConsumerDoors;
 - ConsumerElevators;
 - ConsumerLadders.





Etapa 2

Coordenação, consenso e controle de concorrência

```
bruna@bruna: ~/SCAPS/Partell
-----
Entradas Totais: 126 -> Portas: 54 Escadas: 31 Elevadores: 41
Saídas Totais: 100 -> Portas: 52 Escadas: 29 Elevadores: 19
-----
Total Pessoas no Shopping: 26

bruna@bruna: ~/SCAPS/Partell
-----
Entradas Totais: 127 -> Portas: 54 Escadas: 31 Elevadores: 42
Saídas Totais: 100 -> Portas: 52 Escadas: 29 Elevadores: 19
-----
Total Pessoas no Shopping: 27

bruna@bruna: ~/SCAPS/Partell
-----
Entradas Totais: 127 -> Portas: 54 Escadas: 31 Elevadores: 42
Saídas Totais: 101 -> Portas: 53 Escadas: 29 Elevadores: 19
-----
Total Pessoas no Shopping: 26

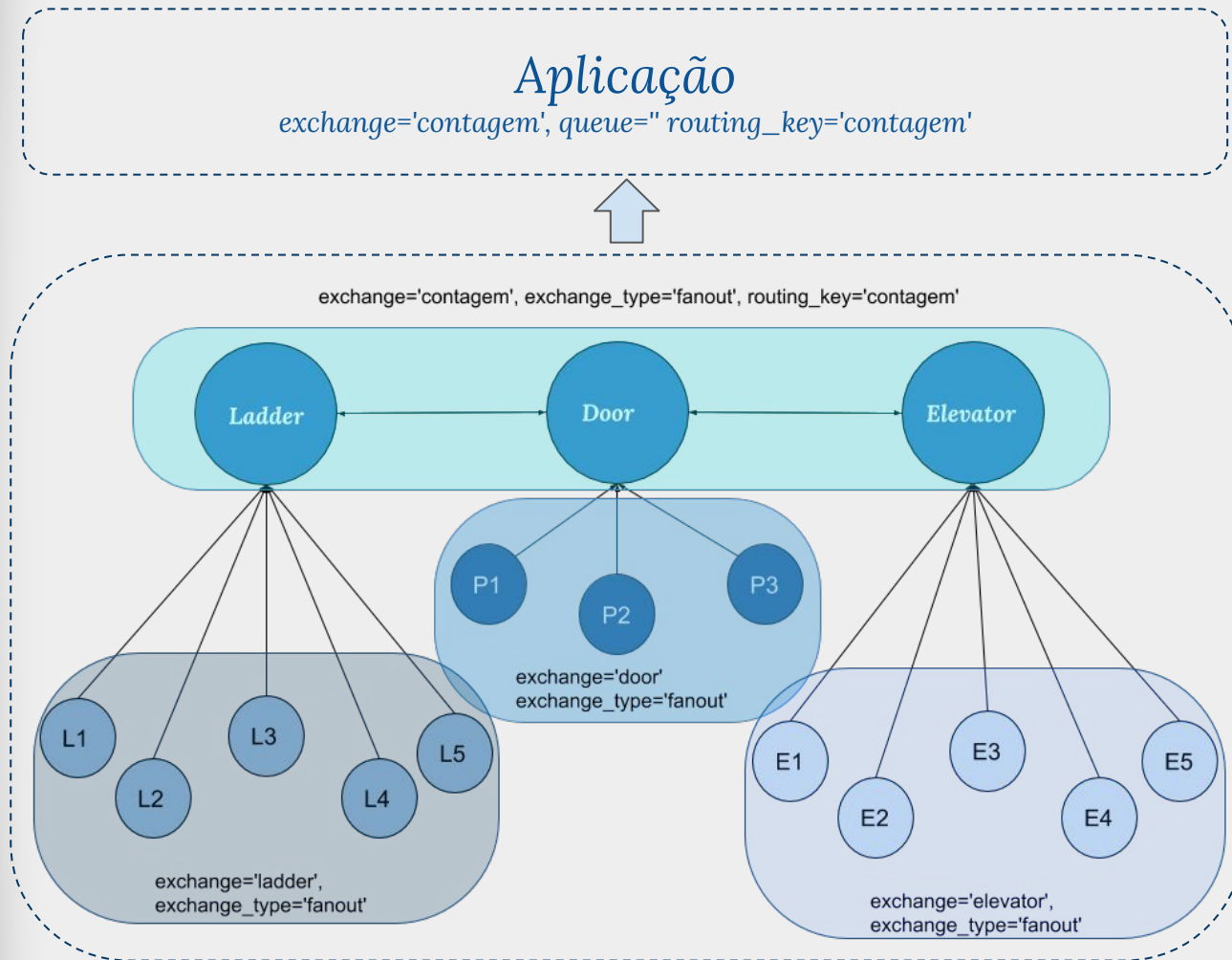
bruna@bruna: ~/SCAPS/Partell
-----
Entradas Totais: 127 -> Portas: 54 Escadas: 31 Elevadores: 42
Saídas Totais: 102 -> Portas: 53 Escadas: 30 Elevadores: 19
-----
Total Pessoas no Shopping: 25
```

Arquitetura

SCAPS

Acesso ao primeiro piso

- 3 portas
- 5 escadas
- 5 elevadores





Etapa 3

Tolerância a falhas e replicação (confiabilidade, disponibilidade, performance)

- Quando uma conexão falha, as mensagens podem estar em trânsito entre o cliente e servidor. Em tais eventos, as mensagens precisarão ser retransmitidas. Os *Acknowledgements* - Acks permitem que o servidor e os clientes saibam quando fazer isso.
- Os Acks podem ser usados em ambas as direções: para permitir que um *consumidor* indique ao servidor que recebeu e/ou processou uma entrega e para permitir que o *servidor* indique a mesma coisa ao *consumidor*. O uso de Ack garante pelo menos uma entrega.
- Do ponto de vista de balanceamento de cargas, ao utilizar o AMQP dentro do RabbitMQ, o balanceamento já é efetuado naturalmente entre os consumidores.

```
80 channel.basic_qos(prefetch_count=1)
81 channel.basic_consume(queue=queue_door, on_message_callback=callback, auto_ack=True)
82 channel.start_consuming()
```



Etapa 4

Aspectos de segurança da interação entre os componentes e apresentação final

No RabbitMQ, as filas têm propriedades que definem o comportamento e a segurança. No trabalho foram usadas:

- Nome
- Durável (a fila sobreviverá a uma reinicialização do corretor)
- Exclusivo (usado por apenas uma conexão e a fila será excluída quando a conexão for fechada)
- Tipo (fanout)
- Ligação (routing_key - chave de ligação)

Usuário e senha configurado para acesso ao RabbitMQ

```
sudo rabbitmqctl add_user scaps 123456789
sudo rabbitmqctl set_user_tags scaps administrator
sudo rabbitmqctl set_permissions -p / scaps "." "." ".*"
```



| **AWS**



Detalhes

- Características da máquina
 - Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0bcc094591f354be2 (64-bit x86) / ami-0bc556e0c71e1b467 (64-bit Arm)
 - Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)
- Grupo de segurança
 - ScapsSecurityGroup
 - Protocolo TCP, Intervalo de Portas 5672

EC2 > Grupos de segurança > sg-086c18a68f03386ab - ScapsSecurityGroup

sg-086c18a68f03386ab - ScapsSecurityGroup

Detalhes

Nome do grupo de segurança ScapsSecurityGroup	ID do grupo de segurança sg-086c18a68f03386ab	Descrição Projeto SCAPS de SD
Proprietário 315894590456	Número de regras de entrada 1 Entrada de permissão	Número de regras de saída 1 Entrada de permissão

Regras de entrada

Tipo	Protocolo	Intervalo de portas	Origem
TCP personalizado	TCP	5672	sg-086c18a68f03386ab (ScapsSecurityGroup)

SCAPS

Sistema de Controle de Acesso de Pessoas em um Shopping

Bruna Michelly de O. S. Cordeiro
Thalia Santos de Santana

2020