The background of the slide features a close-up, artistic shot of a glass being filled with a vibrant orange liquid. The glass is partially visible on the right side, with the liquid's surface creating a smooth, curved line. The lighting is soft, highlighting the texture of the glass and the rich color of the liquid. The overall aesthetic is clean and modern.

REORGANIZAÇÃO INTELIGENTE DE CLUSTERS EM COMPUTAÇÃO EM NÉVOA UTILIZANDO APRENDIZADO POR REFORÇO

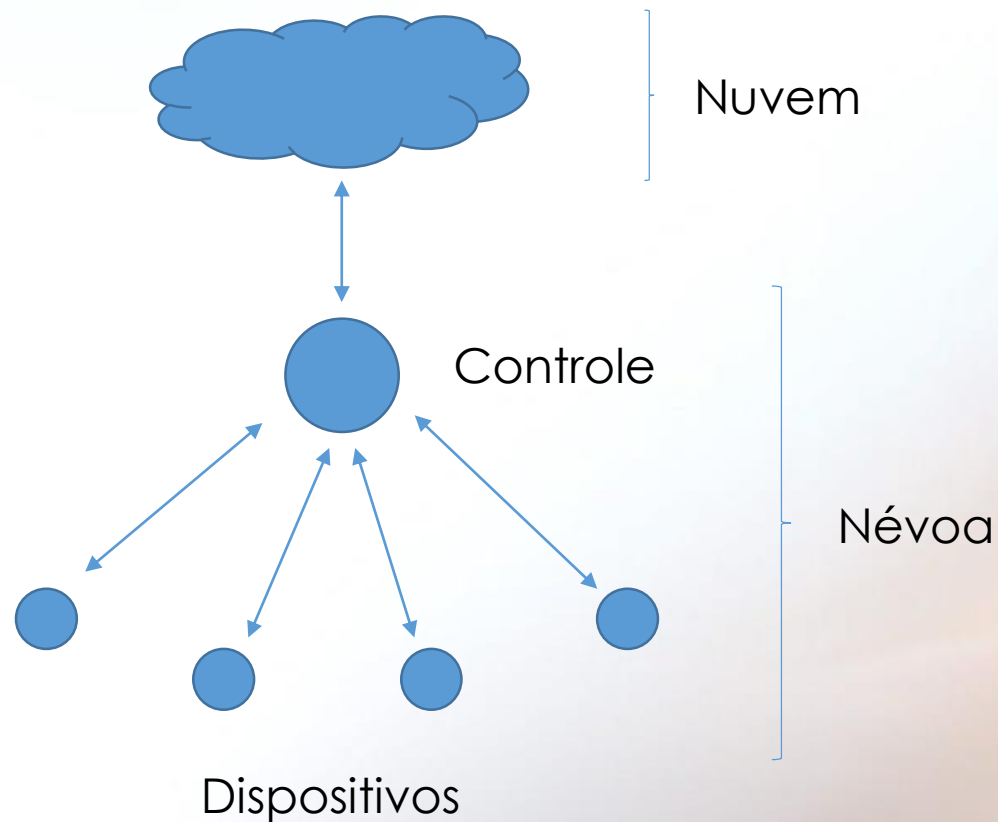
Aluno: Moises H. Pereira - 83390

Orientador: Vitor Barbosa Carlos de Souza

Sumário

- Névoa
- Motivação
- Projeto
- Desafios
- O que foi feito - Em andamento
- Aprendizado por reforço
- Modelagem - Testes
- Cronograma
- Referências

Névoa



- Nuvem próxima do usuário
- Capacidade de processamento menor
- Mais rápida
- Diminui o tráfego de rede

Motivação

- Popularização da IoT (Internet of Things)
- Aplicações de tempo real
- Complementação da Nuvem
- Flexibilidade quanto a tamanho e recurso

Projeto

Estudar uma abordagem para reorganização inteligente dos dispositivos disponíveis para compor a Névoa, utilizando aprendizagem por reforço

Desafios

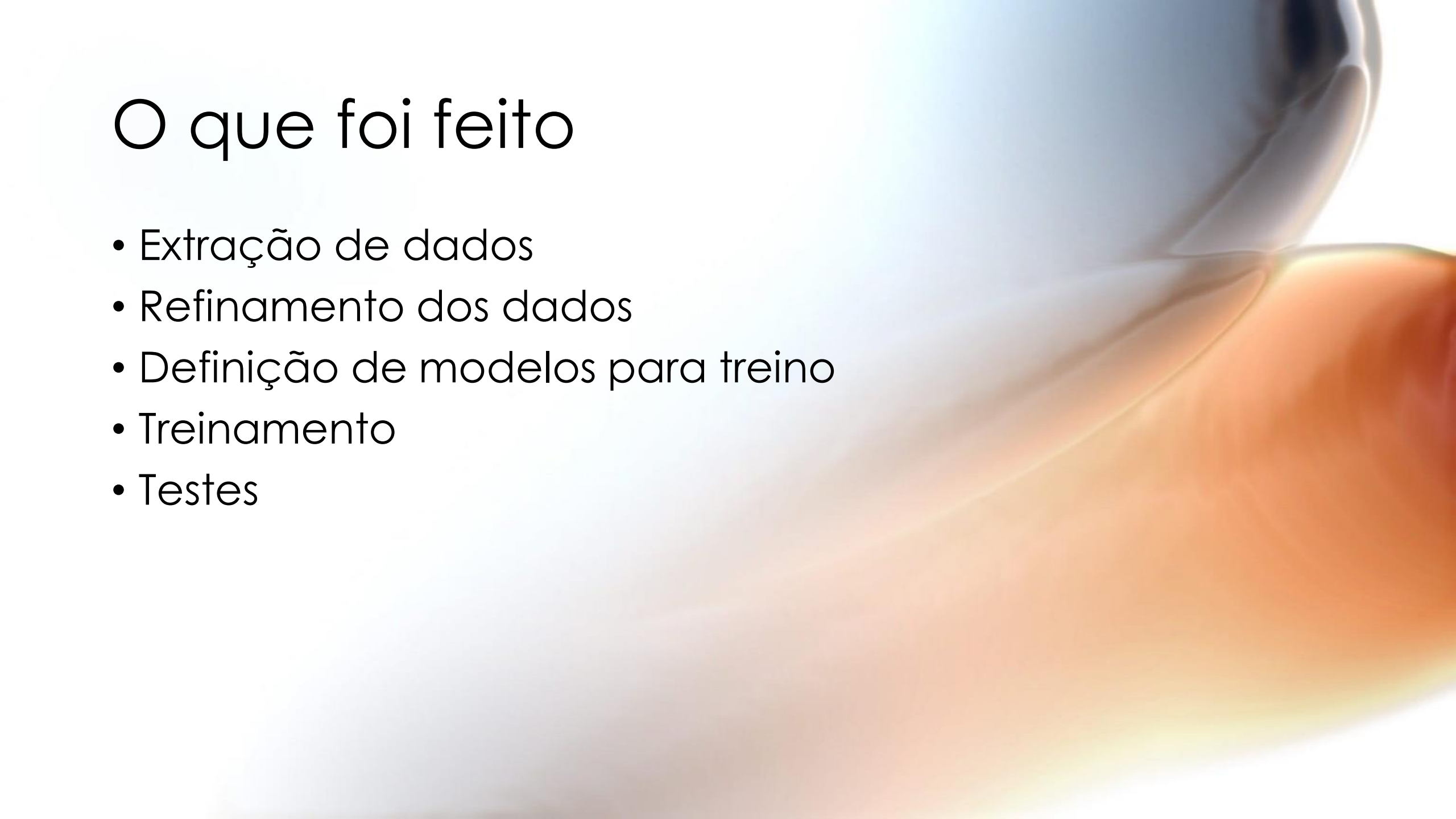
- Estudar e escolher o simulador que será usado
- Desenvolver um ambiente de testes
- Definir variáveis que serão necessárias
- Extrair dados
- Refinamento
- Modelagem do problema
- Utilizar aprendizagem por reforço para clusterização

O que foi feito

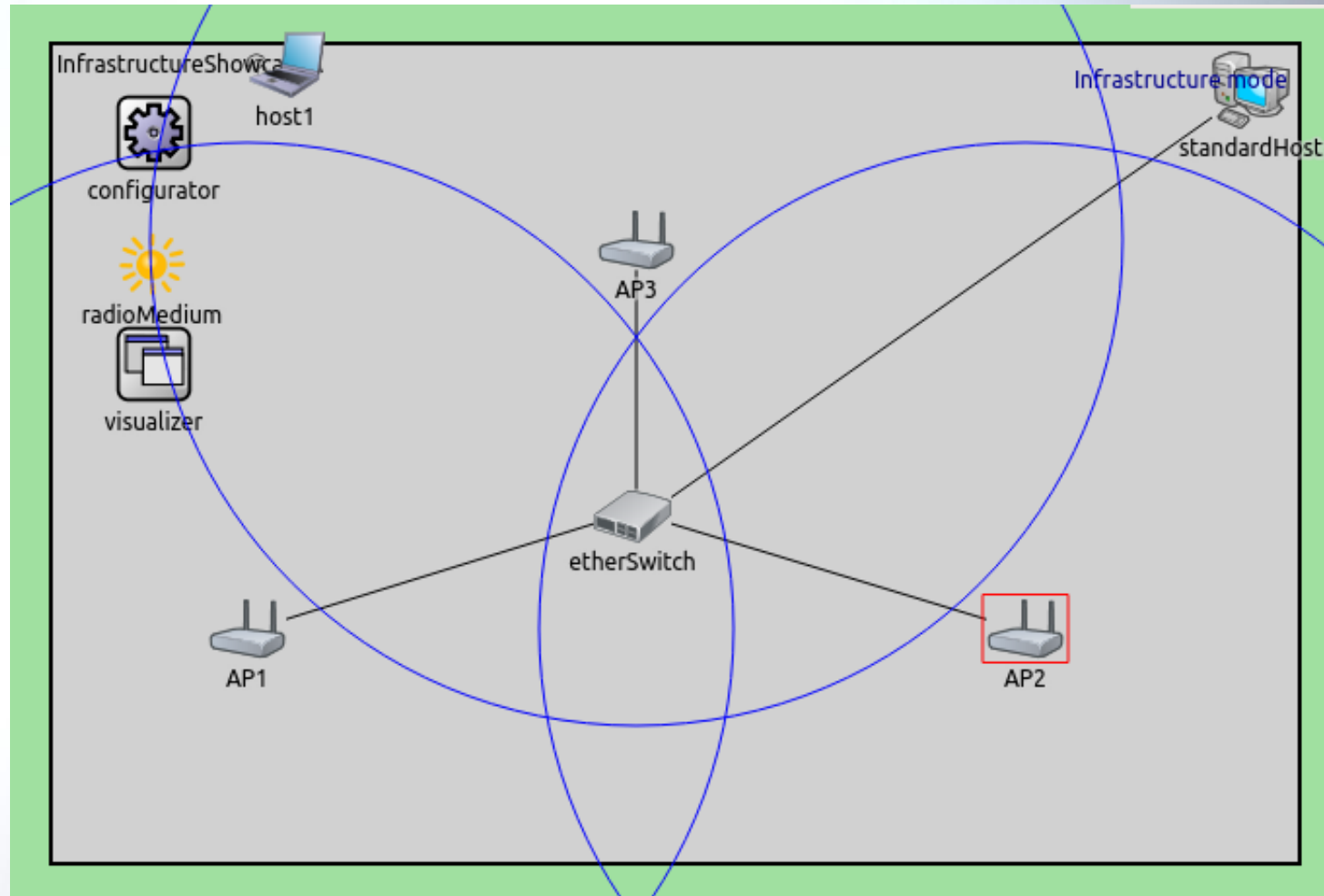
- Estudar e escolher o simulador que será usado
 - OMNeT++
- Desenvolver um ambiente de testes
- Definir variáveis que serão necessárias
 - Posição do host
 - Tempo conectado
 - Tempo desconectado
 - Ponto de acesso que esta conectado

O que foi feito

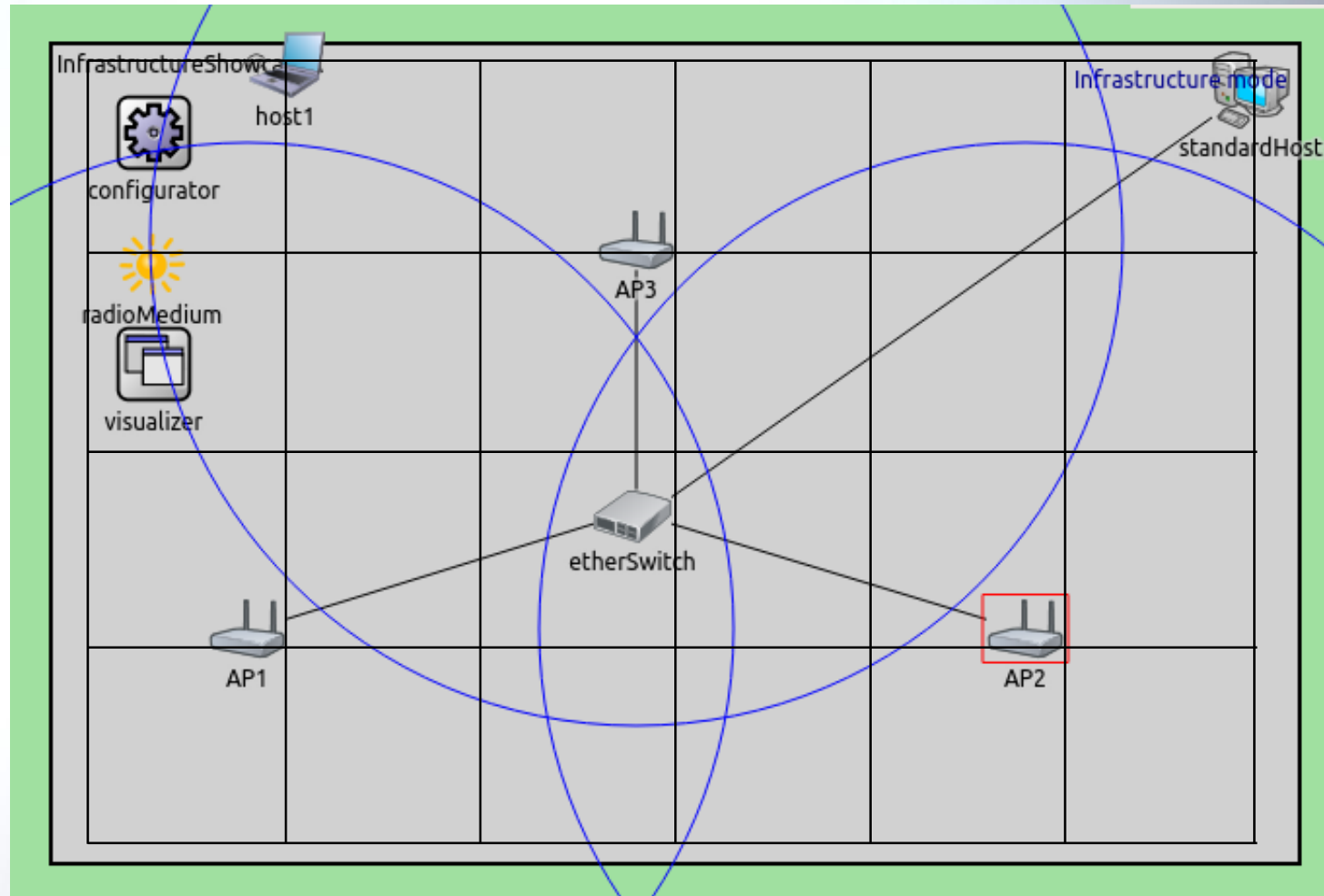
- Extração de dados
- Refinamento dos dados
- Definição de modelos para treino
- Treinamento
- Testes



Ambiente de testes



Ambiente de testes



Em andamento

- Treinamento
- Modelagem dos testes
- Testes



Aprendizado por reforço

- Aprende estratégias ao interagir com o meio (ambiente)
- Aprende por meio de recompensas
- Em fase de treinamento o agente testa diferentes ações para tentar maximizar a recompensa

Aprendizado por reforço

- Política: comportamento do agente no tempo
- Recompensa: informa ao agente o que é bom e o que é ruim
- Função valor: montante de recompensas em um dado período de tempo
- Modelo do ambiente: representação aproximada do ambiente (opcional)

Modelagem – recompensas

- Toda vez que o agente se desconectar de um ponto de acesso ele será punido
- A cada instante que o agente ficar desconectado ele será punido
- A cada instante que o agente ficar conectado em um dado ponto de acesso ele recebe recompensa

Modelagem V2 – matriz valor

	(AP_0, M)	(AP_1, M)	(AP_2, M)	(AP_3, M)
(P_A, P_0, AP_0)				
(P_A, P_0, AP_1)				
(P_A, P_0, AP_2)				
(P_A, P_0, AP_3)				
...				

Legenda:

(AP_i, M) : conjunto de ações disponíveis para cada estado;

(P_A, P_j, AP_i) : conjunto de estados;

Resultados

```
(' (210,210,0)', ' (220,210,0)', 'AP3') : { AP3:1.9999241602040851 AP0:-1.0 AP1:-0.00014759509502946844 }
(' (210,210,0)', ' (220,210,0)', 'AP0') : { AP3:0.0 AP0:0.0 AP1:0.0 }
(' (210,210,0)', ' (220,210,0)', 'AP1') : { AP3:-8.870412759298798e-05 AP0:-1.0 AP1:1.9998147114964362 }
(' (230,210,0)', ' (220,210,0)', 'AP3') : { AP3:1.999934555364137 AP0:-1.0 AP1:-4.293286628379699e-05 }
(' (230,210,0)', ' (220,210,0)', 'AP0') : { AP3:0.0 AP0:0.0 AP1:0.0 }
(' (230,210,0)', ' (220,210,0)', 'AP1') : { AP3:-0.00015172736247468887 AP0:-1.0 AP1:1.9999458557839938 }
(' (220,200,0)', ' (220,210,0)', 'AP3') : { AP3:1.999875047147281 AP0:-1.0 AP1:-1.1395363826906097e-05 }
(' (220,200,0)', ' (220,210,0)', 'AP0') : { AP3:0.0 AP0:0.0 AP1:0.0 }
(' (220,200,0)', ' (220,210,0)', 'AP1') : { AP3:-0.000142666733135934 AP0:-1.0 AP1:1.9998207327957969 }
(' (220,220,0)', ' (220,210,0)', 'AP3') : { AP3:1.9999255079704652 AP0:-1.0 AP1:-7.086185678337965e-05 }
(' (220,220,0)', ' (220,210,0)', 'AP0') : { AP3:0.0 AP0:0.0 AP1:0.0 }
(' (220,220,0)', ' (220,210,0)', 'AP1') : { AP3:-6.093362419660527e-05 AP0:-1.0 AP1:1.999618319206946 }
```

OBSERVAÇÃO

Vetor [a,b,c][d,e] representa:

a: recompensa por ficar conectado em um ponto de acesso

b: punição por trocar de ponto de acesso

c: punição por ficar desconectado

d: valor de alpha

e: valor de gama

Fórmula usada para o treinamento:

$$Q(S_t, A) = Q(S_t, A) + \alpha[R + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A)]$$

Teste 3 APs - $[0, -1, -1][0.1, 0.9]$

Considerando $[0, -1, -1][0.1, 0.9]$:

. Iteração 100

Porcentagem tempos médios: -62.3975980044286%

Porcentagem intervalos: -78.96685854655536%

. Iteração 500

Porcentagem tempos médios: -62.79310890626556%

Porcentagem intervalos: -79.3955263936181%

. Iteração 1000

Porcentagem tempos médios: -61.93361821443415%

Porcentagem intervalos: -79.12204849098323%

Teste 3 APs - $[1, -1, -1][0.1, 0.9]$

Considerando $[1, -1, -1][0.1, 0.9]$:

. Iteração 100

Porcentagem tempos médios: 11.528457316762257%

Porcentagem intervalos: 15.748521239722834%

. Iteração 500

Porcentagem tempos médios: 11.528457316762257%

Porcentagem intervalos: 15.748521239722834%

. Iteração 1000

Porcentagem tempos médios: 11.528457316762257%

Porcentagem intervalos: 15.748521239722834%

Teste 3 APs - $[1, -1, -1][0.1, 0.9]$

Detalhando um pouco mais, tem-se os dados apenas para a iteração 1000:

AP1

- Melhorou 3.0219248539796126% tempo médio conectado
- Melhorou 1.477832512315274% número de trocas

AP2

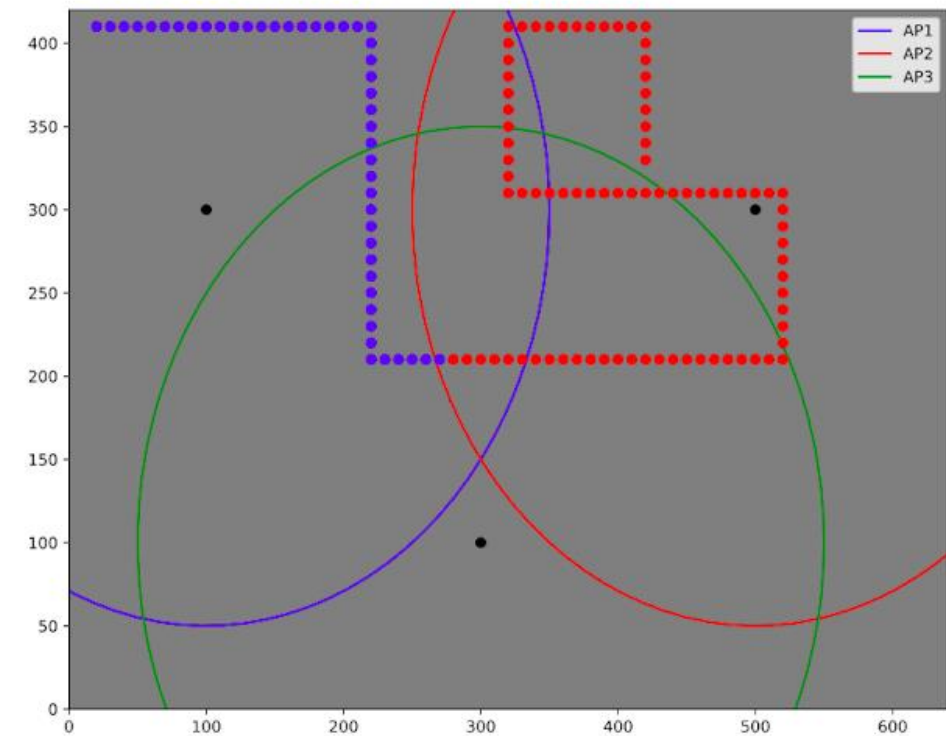
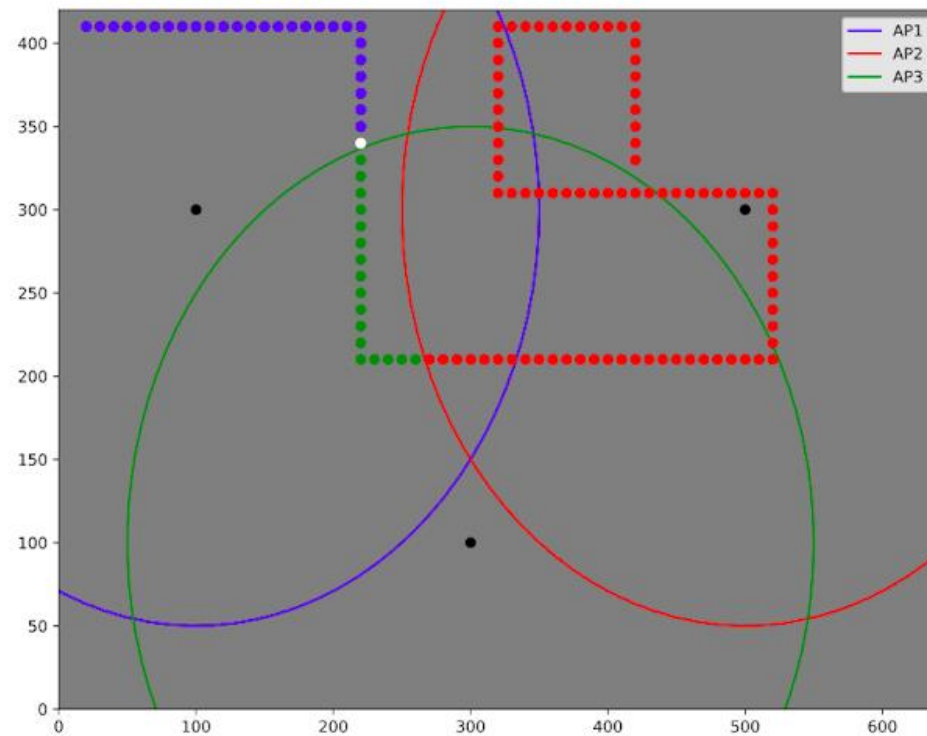
- Melhorou 2.189367266922787% tempo médio conectado
- Piorou 0.8894536213468882% número de trocas

AP3

- Melhorou 6.317165195859857% tempo médio conectado
- Melhorou 15.160142348754448% número de trocas

Teste 3 APs

Simulador X Q-Learning --- Rota 176

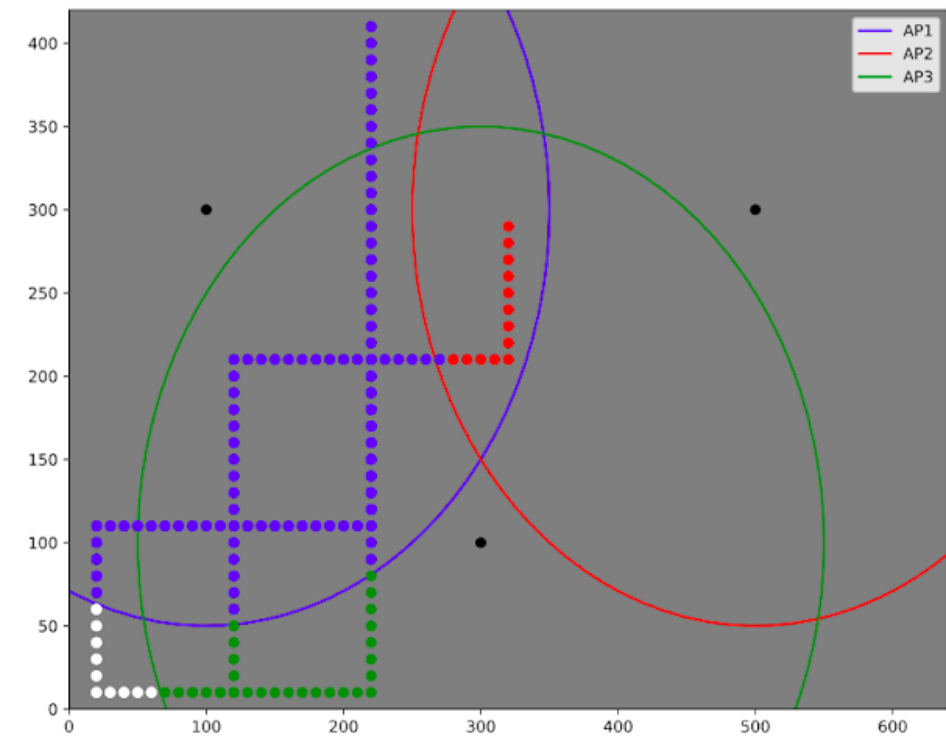
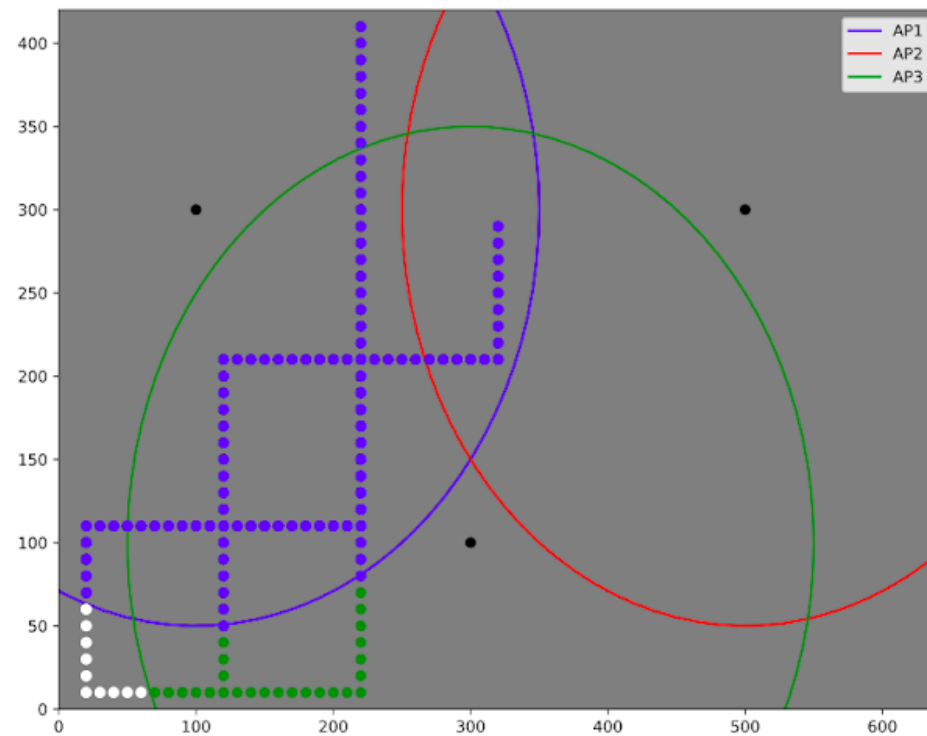


Teste 3 APs

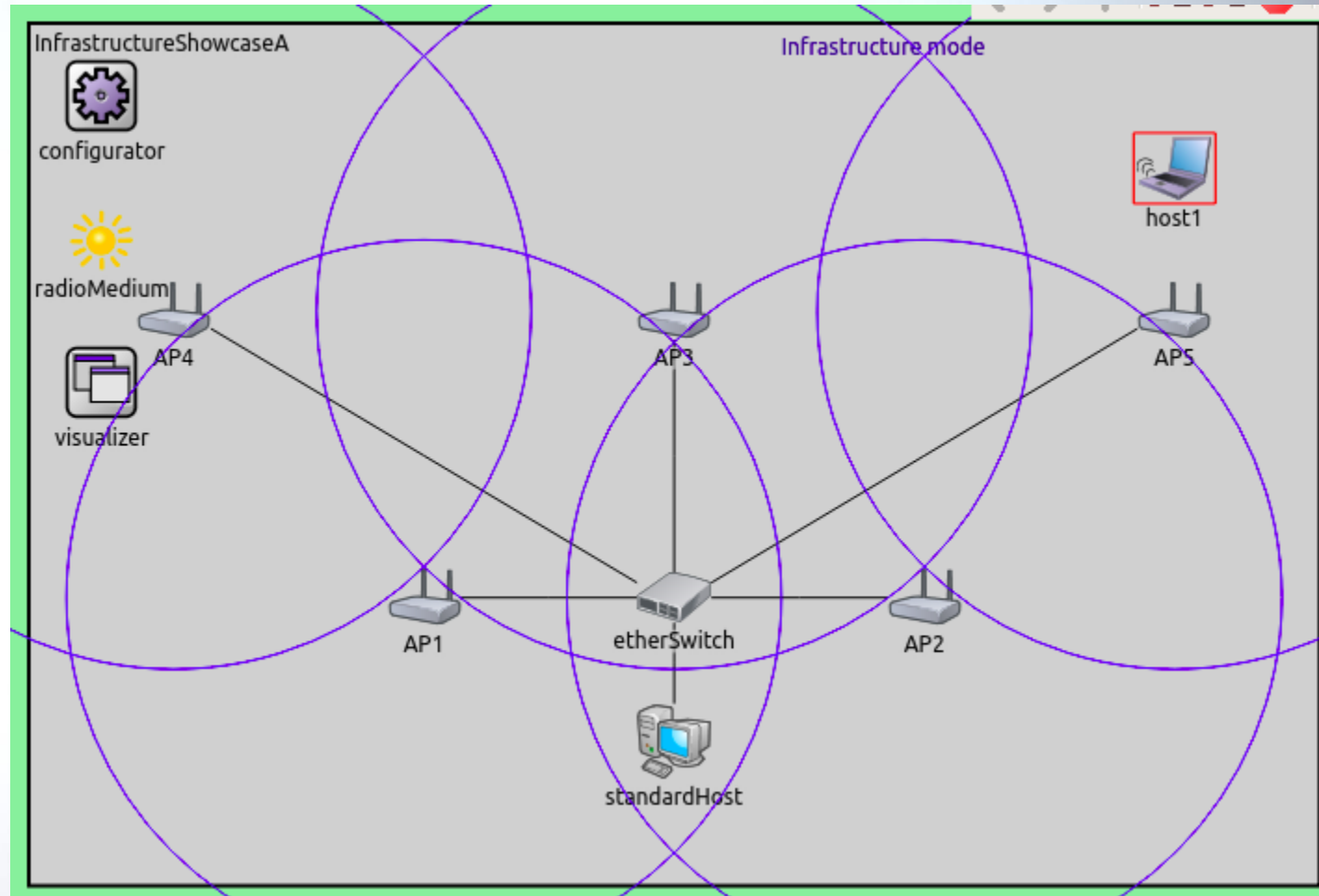
Entretanto essa melhora não abrange todos os casos, há alguns em que houve uma ligeira piora na escolha dos pontos de acesso:

Teste 3 APs

Simulador X Q-Learning --- Rota 146



Teste 5 APs



Teste 5 APs

Considerando $[1, -1, -1][0.1, 0.9]$:

. 100 iterações

Porcentagem tempos médios: 31.943589900503014%

Porcentagem intervalos: -111.64340643030803%

. 500 iterações

Porcentagem tempos médios: 31.900353362720693%

Porcentagem intervalos: -111.80832848769184%

.1000 iterações

Porcentagem tempos médios: 31.900353362720693%

Porcentagem intervalos: -111.80832848769184%

Teste 5 APs

Detalhando um pouco mais, tem-se os dados apenas para a iteração 1000:

AP1

- Melhorou 5.206873978088055% tempo médio conectado
- Melhorou 0.8442776735459745% número de trocas

AP2

- Melhorou 10.967818811939352% tempo médio conectado
- Melhorou 8.16326530612244% número de trocas

AP3

- Melhorou 4.426732839838493% tempo médio conectado
- Piorou 9.2071611253197% número de trocas

Teste 5 APs

Detalhando um pouco mais, tem-se os dados apenas para a iteração 1000: (continuação)

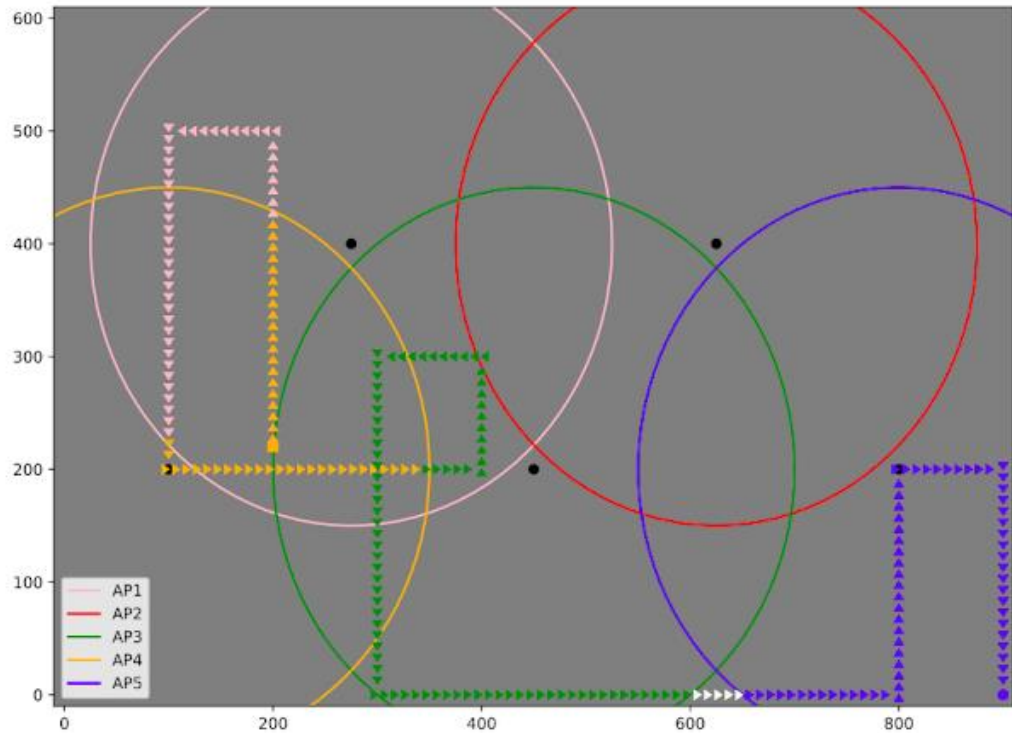
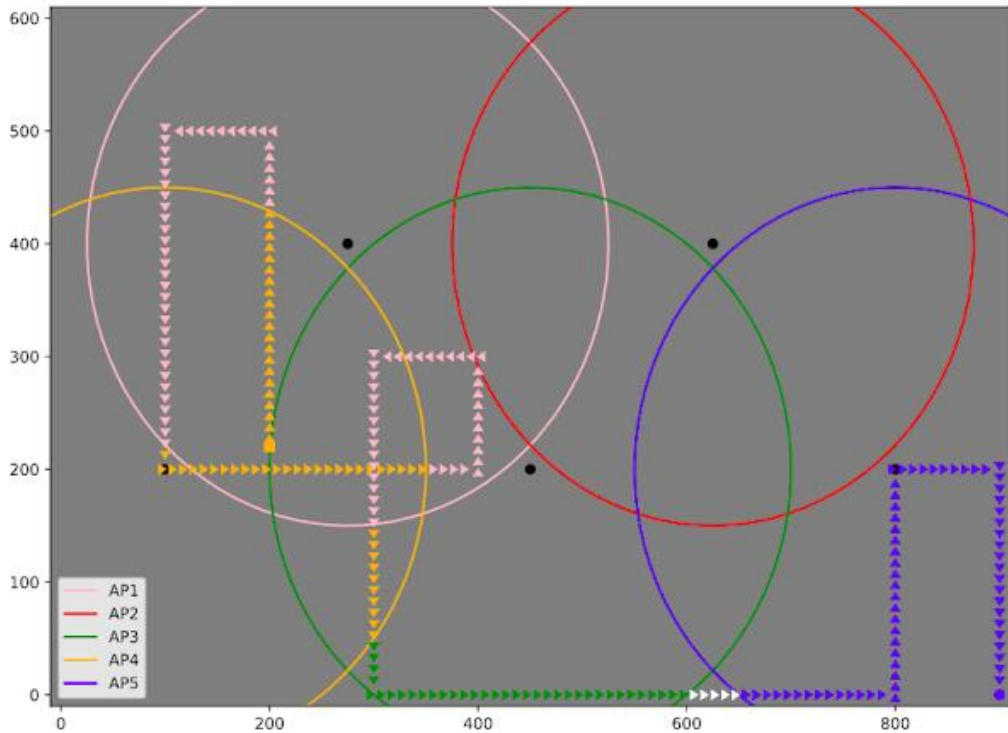
AP4

- Melhorou 3.8857501072208294% tempo médio conectado
- Piorou 0.5385996409335689% número de trocas

AP5

- Melhorou 7.413177625633963% tempo médio conectado
- Piorou 111.07011070110698% número de trocas

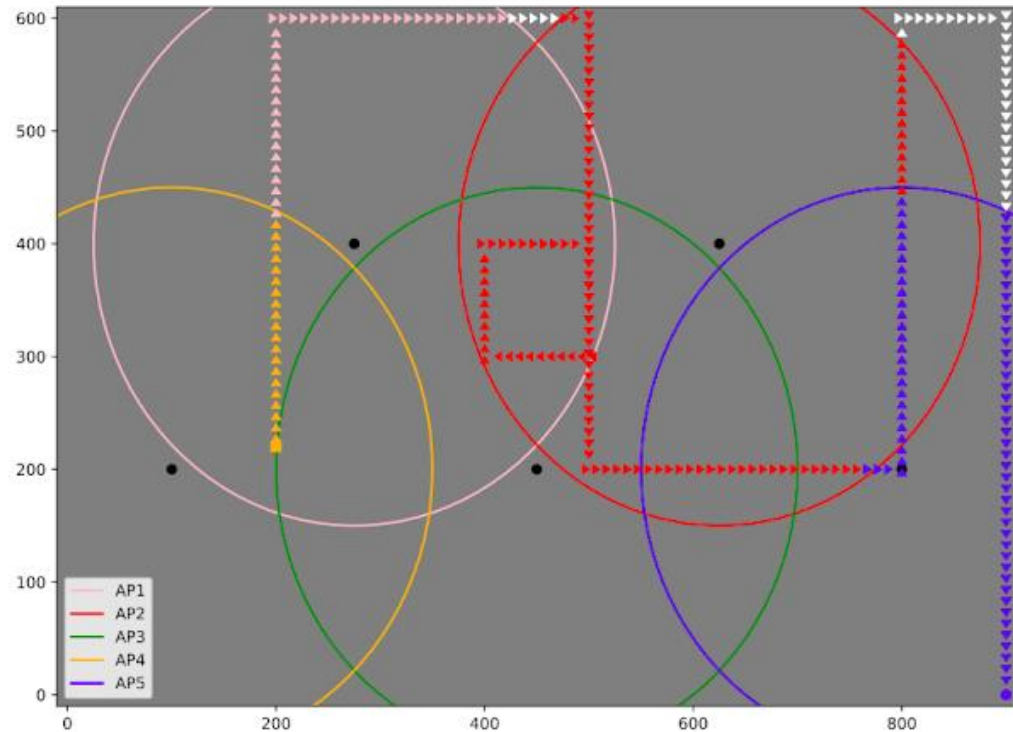
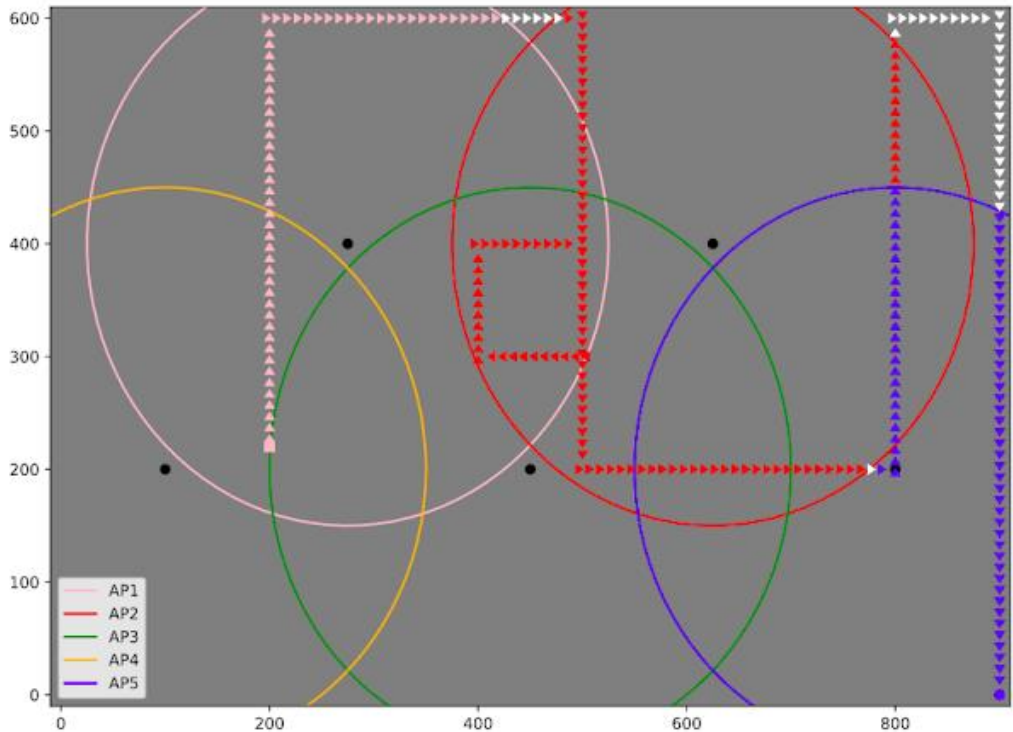
Simulador X Q-Learning --- Rota 10



Teste 5 APs

Entretanto essa melhora não abrange todos os casos, há alguns em que houve uma ligeira piora na escolha dos pontos de acesso:

Simulador X Q-Learning --- Rota 37



Teste 5 APs

Tentando melhorar o algoritmo optou-se por trocar de:

```
APn = random.choice(list(est[P]))
```

Para:

```
APn = random.choice(list(est[P]))
```

```
rand = random.randint(0, 1)
```

```
if rand == 0:
```

```
    listKey = list(est_ac[S].keys())
```

```
    listValue = list(est_ac[S].values())
```

```
    APn = listKey[listValue.index(max(listValue))]
```

Teste 5 APs

Considerando $[1, -1, -1][0.1, 0.9]$:

.100 iterações

Porcentagem tempos médios: 10.849226743471846%

Porcentagem intervalos: -83.2492289290307%

.500 iterações

Porcentagem tempos médios: 32.616029670537415%

Porcentagem intervalos: -61.728501725916615%

.1000 iterações

Porcentagem tempos médios: 32.449907791601845%

Porcentagem intervalos: -61.728501725916615%

Teste 5 APs

Detalhando um pouco mais, tem-se os dados apenas para a iteração 1000:

AP1

- Piorou 4.257220058794914% tempo médio conectado
- Melhorou 11.632270168855527% número de trocas

AP2

- Melhorou 8.372154384301169% tempo médio conectado
- Melhorou 3.265306122448976% número de trocas

AP3

- Melhorou 5.228550273361179% tempo médio conectado
- Piorou 13.17135549872124% número de trocas

Teste 5 APs

Detalhando um pouco mais, tem-se os dados apenas para a iteração 1000: (continuação)

AP4

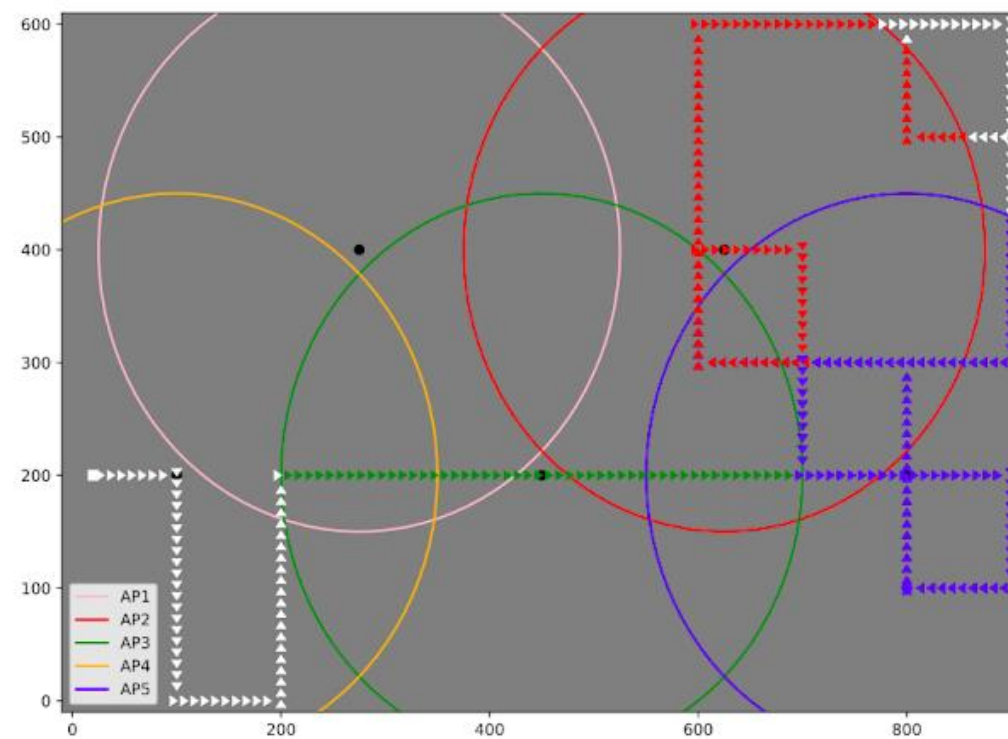
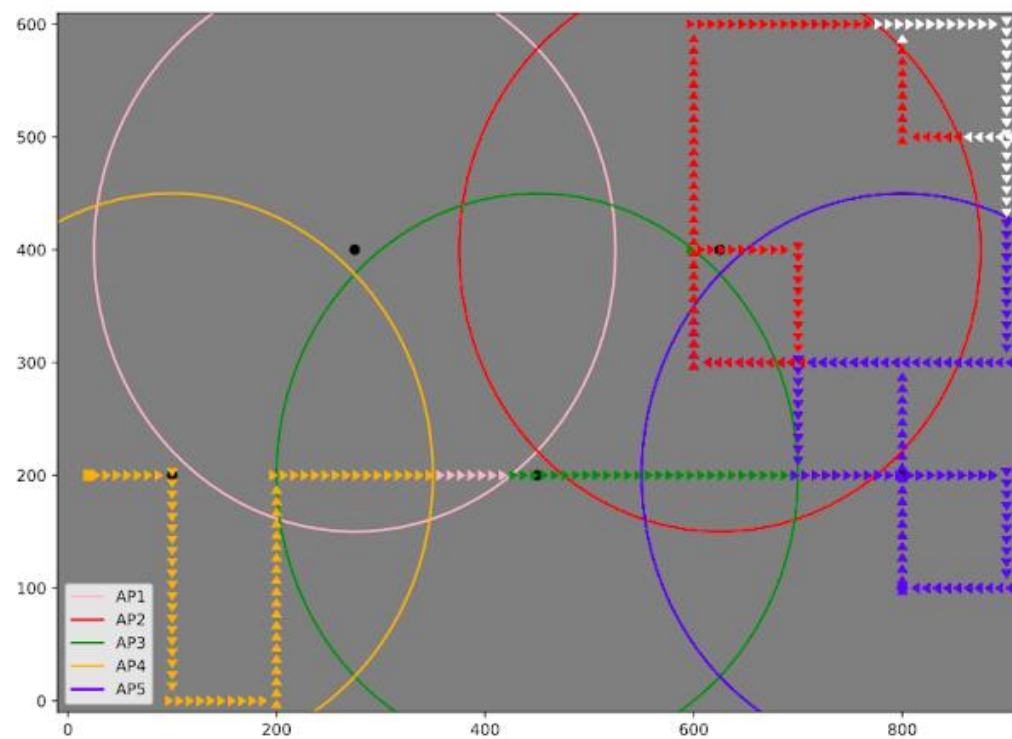
- Melhorou 12.841886087102992% tempo médio conectado
- Melhorou 39.497307001795335% número de trocas

AP5

- Melhorou 10.264537105631405% tempo médio conectado
- Piorou 102.95202952029521% número de trocas

Teste 5 APs

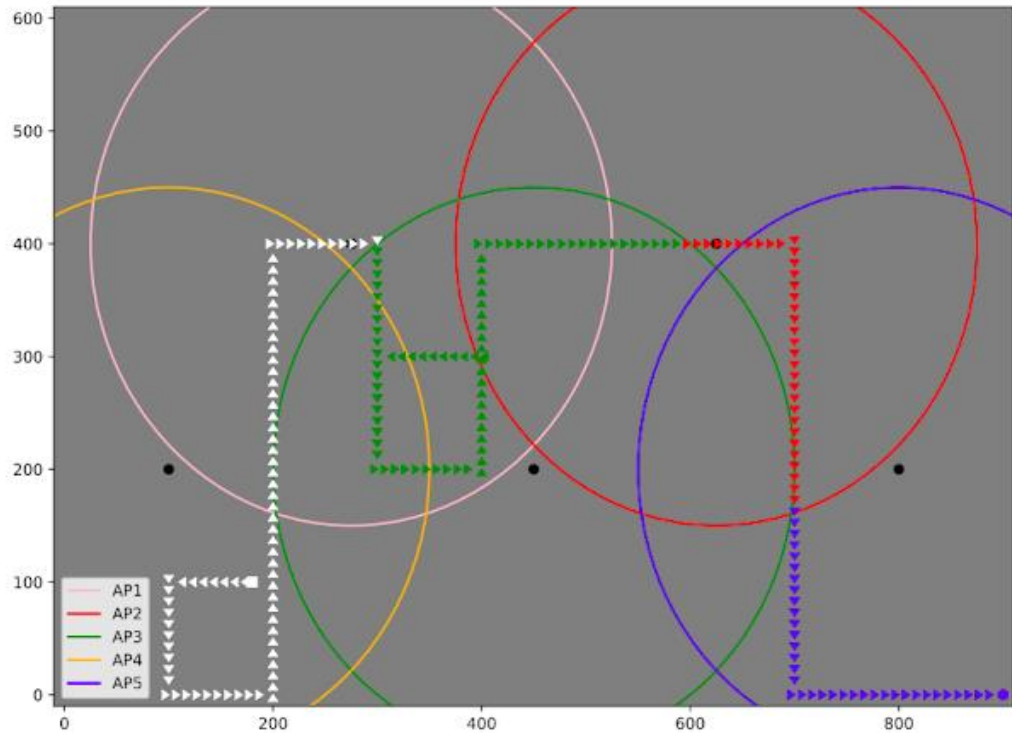
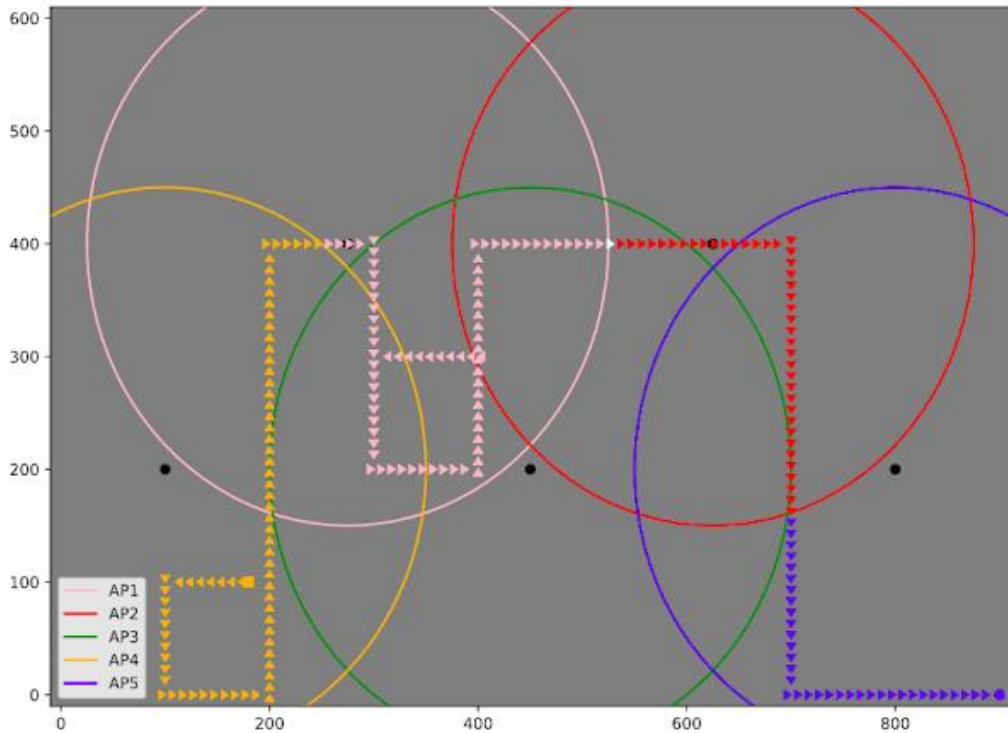
Simulador X Q-Learning --- Rota 22



Teste 5 APs

- Entretanto essa melhora não abrange todos os casos, há alguns em que houve uma ligeira piora na escolha dos pontos de acesso:

Simulador X Q-Learning --- Rota 12



Teste 5 APs

Código 1

Melhora TM %	Melhora NT %
35.76	-101.49
25.78	-42.10
32.84	-107.03
17.96	-107.13
31.90	-111.80

Código 2

Melhora TM %	Melhora NT %
25.69	-40.77
43.62	-85.51
32.44	-61.72
23.25	-106.58
25.61	-31.58

Legenda:

TM: tempo médio conectado

NT: número de trocas

Cronograma 2019-2

	Agosto	Setembro	Outubro	Novembro	Dezembro
1.	X	X			
2.	X	X	X		
3.	X	X	X	X	
4.			X	X	X

Legenda:

- 1. Modelagem
- 2. Implementação
- 3. Testes
- 4. Escrita

Obrigado!

Link do projeto:

<https://github.com/MoisesHenr/INF496>

Referências

- Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (pp. 13-16). ACM.
- Chiang, Mung, et al. "Clarifying fog computing and networking: 10 questions and answers." *IEEE Communications Magazine* 55.4 (2017): 18-20.
- Perato, L., & Al Agha, K. (2002). Handover prediction: user approach versus cell approach. In *4th International Workshop on Mobile and Wireless Communications Network* (pp. 492-496). IEEE.

Referências

- Shin, M., Mishra, A., & Arbaugh, W. A. (2004, June). Improving the latency of 802.11 hand-offs using neighbor graphs. In Proceedings of the 2nd international conference on Mobile systems, applications, and services (pp. 70-83). ACM.
- SUTTON, R. S., & BARTO, A. G. (2018). REINFORCEMENT LEARNING: AN INTRODUCTION. MIT PRESS.
- Imagem slide 3: autoria própria