

# ELEIÇÕES

**Agentes e Inteligência Artificial Distribuída**  
**Mestrado Integrado em Engenharia Informática e Computação**

## **Grupo 24**

Bárbara Sofia Silva (201505628)

Julieta Frade (201506530)

Ventura Pereira (201404690)

# APRESENTAÇÃO

Primeira Parte



## DESCRIÇÃO DO PROBLEMA

Tanto os **candidatos** como os **eleitores** possuem **crenças políticas**. No caso dos candidatos este conjunto representa a sua **estratégia** que poderá mudar ao longo do processo eleitoral, contudo se a alterarem em demasia, vão sofrer perdas no que toca ao fator de **credibilidade**.

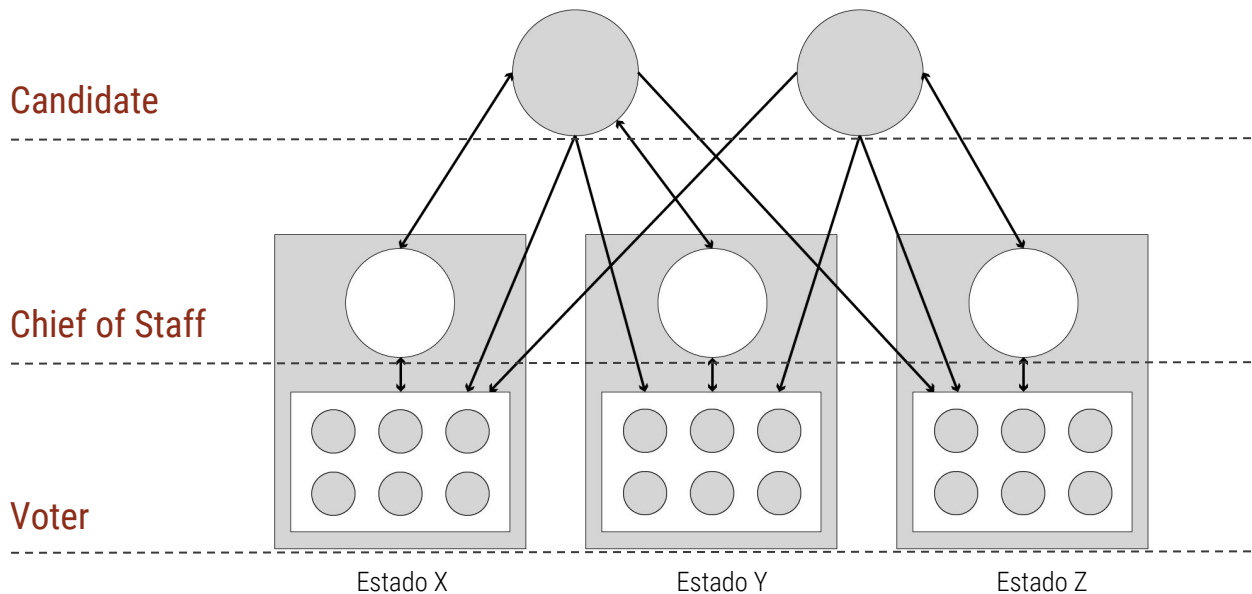
Cada candidato tem uma **equipa de campanha** composta por **chefes de estado**. Cada elemento desta equipa é responsável por fazer **sondagens** à população do estado que lhe foi atribuído e seguidamente **aconselhar** o seu candidato em relação à próxima mudança de estratégia.

Face ao fator de credibilidade de cada candidato, o eleitor exige um valor mínimo de credibilidade por parte destes, caso nenhum tenha atingido este nível a pessoa abster-se-á. Se nenhum candidato tiver em pelo menos metade das crenças o nível mínimo requerido pelo eleitor, este também poderá exercer o seu direito de **abstinência**.



## ESQUEMA GLOBAL

Foram consideramos 3 tipos de agentes: **Voter**, **Chief of Staff** and **Candidate**. No esquema seguinte estão visíveis as interações entre agentes.





# INTERAÇÃO E PROTOCOLOS

No que toca às seguintes interações entre agentes, são utilizadas **ACL Messages** sem protocolos de JADE. As mensagens são enviadas e recebidas com **simple behaviours**.

## Interação Candidate - Voter

- Candidate envia as suas crenças e a sua credibilidade
- Voter recebe as crenças do Candidate e a sua credibilidade

## Interação Voter - Chief of Staff

- Chief Of Staff envia ao Voter o nome do Candidate que ele representa e pergunta em quem vai votar.
- Voter recebe o nome do candidato que Chief Of Staff representa e a pergunta.
- Voter envia ao Chief Of Staff o nome do Candidate no qual vai votar. Caso este não seja o candidato que o Chief Of Staff representa diz qual a crença que esse Candidate deve mudar para o Voter considerar votar nele.

O protocolo **Contract Net** é usado na seguinte interação

## Interação Chief of Staff - Candidate

- Candidate pergunta aos seus Chiefs Of Staff o que deve mudar na sua estratégia.
- Cada Chief Of Staff envia uma mensagem que diz se este Candidate está a ganhar ou a perder no seu estado. Caso esteja a perder manda-lhe a crença que deve mudar e o valor para o qual deve mudar.
- Candidate aceita a proposta que vier do Chief Of Staff que representa o estado com mais população.



# ARQUITETURAS E ESTRATÉGIAS

## VOTER

**Escolha do Candidato:** Primeiramente, o eleitor calcula o máximo de crenças que um candidato pode ter fora dos intervalos definidos por si (*maxWrongBeliefs*), este valor é  $1/2$  do número total de crenças. O eleitor escolhe o candidato que tiver credibilidade maior do que a mínima que ele definiu anteriormente e aquele que tem o número de crenças fora de intervalos menor que o *maxWrongBeliefs*. Se nenhum candidato cumprir estas condições o eleitor votará em nulo (abstinência).

**Escolha da crença que o candidato representado pelo chefe de estado deve mudar:** Caso o eleitor tenha escolhido o candidato do chefe de estado, esta crença vai ser nula. Caso contrário, o eleitor calcula a mediana de todos os seus intervalos relativos às crenças e seguidamente calcula que crença desse candidato é que está mais longe da respetiva mediana.

## CHIEF OF STAFF

**Escolher a informação que passa ao candidato que representa:** Se a maioria dos eleitores do seu estado disser que vai votar no seu candidato, o chefe de estado diz ao mesmo que está a ganhar. Caso esteja a perder, ele vê qual é a crença a mudar (*beliefToChange*) que os eleitores do seu estado escolheram e faz a média dos valores ditos pelas pessoas que escolheram essa crença.

## CANDIDATE

**Escolha da crença a mudar:** Depois de receber em que candidato é que um estado vai votar, que crença é que ele próprio deve mudar e qual o valor, o candidato vai associar esta informação ao número de pessoas de cada estado e vai escolher mudar a crença escolhida pelo maior número de pessoas.

**Valor a subtrair na credibilidade:** O candidato subtrai de credibilidade  $1/4$  do valor associado à mudança da crença.



## OUTROS MECANISMOS E SOFTWARE UTILIZADO

O projeto foi desenvolvido no **IDE Eclipse**, com acesso à framework **JADE**.

Quanto a outros mecanismos, foi implementado uma estrutura de descoberta de agentes Directory Facilitator (DF), onde os agentes do tipo **Voter** estão divididos por estados.





## EXPERIÊNCIAS REALIZADAS

Durante este projeto foram realizadas várias experiências que consistiram em alterar os parâmetros que pensamos que podem influenciar o resultado.

O **valor mínimo e máximo** para a **população** de cada estado, sendo que, se dermos o mesmo valor será esse o número definido. Para além disso temos o **número de candidatos**. Estas três variáveis são enviadas como **argumentos do programa**.

Por último, temos **os estados e as crenças**, que estão codificados e **podem ser adicionados descomentando no código**, de forma a acrescentar à lista de crenças e estados.

No final de uma simulação estamos interessados em dois valores: quem é o **vencedor** das eleições e a quantidade de votos de **abstinência**.

```
public Elections(int minPopulation, int maxPopulation, int nrCandidates) throws StaleProxyException {  
  
    this.minPopulation = minPopulation;  
    this.maxPopulation = maxPopulation;  
    this.nrCandidates = nrCandidates;  
  
    this.states.add("Alaska");  
    this.states.add("California");  
    this.states.add("Florida");  
    this.states.add("Hawaii");  
    this.states.add("Kansas");  
    // this.states.add("Montana");  
    // this.states.add("New Jersey");  
    // this.states.add("New York");  
    // this.states.add("Washington");  
    // this.states.add("Texas");  
  
    this.beliefs.add("Liberalism");  
    this.beliefs.add("Conservatism");  
    this.beliefs.add("Communism");  
    this.beliefs.add("Socialism");  
    this.beliefs.add("Anarchism");  
    // this.beliefs.add("Nationalism");  
    // this.beliefs.add("Fascism");  
    // this.beliefs.add("Monarchism");  
    // this.beliefs.add("Republicanism");  
    // this.beliefs.add("Environmentalism");  
}
```





## ANÁLISE DOS RESULTADOS

Os resultados finais do processo eleitoral demonstram que **a abstinência é mínima** e que, quanto mais parâmetros usarmos, **maior será o tempo de execução**.



## CONCLUSÕES

O primeiro projeto da unidade curricular de Agentes e Inteligência Artificial Distribuída teve como objetivo implementar um sistema composto por múltiplos agentes (**SMA**).

No início do projeto o grupo tinha pensado num mecanismo de influência entre agentes do tipo **Voter**. Os eleitores seriam caracterizados por dois fatores relativos à influência: **assertividade e passividade**. Assertividade corresponde à capacidade de persuasão da restante população do seu estado e passividade o quão influenciável é por estas mesmas pessoas. Infelizmente, devido a falta de tempo não nos foi possível implementar esta interação e é algo que pretendemos fazer no próximo trabalho, assim como fazer uma melhor análise das experiências realizadas.

Em suma, apesar de não terem sido desenvolvidas todas as funcionalidades previstas, o projeto foi finalizado com sucesso.

# INFORMAÇÃO ADICIONAL

Segunda Parte



## EXEMPLOS DETALHADOS DE EXECUÇÃO

Indicação das variáveis iniciais:

```
> STATE: Alaska POPULATION: 3  
> STATE: California POPULATION: 3  
> TOTAL POPULATION: 6  
> NR CANDIDATES: 2
```

O candidato envia as crenças para a população dos diferentes estados.

```
> INFO: ID: candidate_1 CREDIBILITY: 100 BELIEFS: {Conservatism=46, Socialism=36, Communism=31, Anarchism=6, Liberalism=71}  
SENT: [{Conservatism=46, Socialism=36, Communism=31, Anarchism=6, Liberalism=71}, {Credibility=100}] TO: voter_Alaska_2  
SENT: [{Conservatism=46, Socialism=36, Communism=31, Anarchism=6, Liberalism=71}, {Credibility=100}] TO: voter_Alaska_0  
SENT: [{Conservatism=46, Socialism=36, Communism=31, Anarchism=6, Liberalism=71}, {Credibility=100}] TO: voter_Alaska_1  
SENT: [{Conservatism=46, Socialism=36, Communism=31, Anarchism=6, Liberalism=71}, {Credibility=100}] TO: voter_California_2  
SENT: [{Conservatism=46, Socialism=36, Communism=31, Anarchism=6, Liberalism=71}, {Credibility=100}] TO: voter_California_0  
SENT: [{Conservatism=46, Socialism=36, Communism=31, Anarchism=6, Liberalism=71}, {Credibility=100}] TO: voter_California_1
```

## EXEMPLOS DETALHADOS DE EXECUÇÃO

O chefe de estado envia mensagens para a população desse mesmo estado, indicando que candidato representa e sondando os resultados.

```
> INFO:      ID: chiefstaff_0 STATE: Alaska BOSS: candidate_1
SENT:        [My boss is , candidate_1, Who's getting your vote?] TO: voter_Alaska_2
SENT:        [My boss is , candidate_1, Who's getting your vote?] TO: voter_Alaska_0
SENT:        [My boss is , candidate_1, Who's getting your vote?] TO: voter_Alaska_1
```

O eleitor recebe a informação dos candidatos e do respetivo chefe de estado, sendo que ele próprio tem intervalos que considera aceitáveis para as diferentes crenças.

```
> INFO:      ID: voter_Alaska_1 BELIEFS: {Conservatism=[88, 100], Socialism=[74, 100], Communism=[8, 58], Anarchism=[33, 83], Liberalism=[21, 71]} MIN CREDIBILITY: 31
RECEIVED:    [My boss is , candidate_1, Who's getting your vote?] FROM: chiefstaff_0
RECEIVED:    [{Conservatism=46, Socialism=36, Communism=31, Anarchism=6, Liberalism=71}, {Credibility=100}] FROM: candidate_1
RECEIVED:    [{Conservatism=34, Socialism=95, Communism=5, Anarchism=28, Liberalism=97}, {Credibility=100}] FROM: candidate_0
```

## EXEMPLOS DETALHADOS DE EXECUÇÃO

O eleitor escolhe um candidato e responde ao chefe de estado.

```
> INFO: CHOSEN CANDIDATE: candidate_1
SENT: [I'll vote in candidate: , candidate_1, I'll consider changing my vote if your boss candidate_1 changes the belief: , null, to , null] TO: chiefofstaff_0
```

Os chefes de estado recebem as respostas da população.

```
RECEIVED: [I'll vote in candidate: , null, I'll consider changing my vote if your boss candidate_1 changes the belief: , Socialism, to , 100] FROM: voter_Alaska_0
RECEIVED: [I'll vote in candidate: , candidate_1, I'll consider changing my vote if your boss candidate_1 changes the belief: , null, to , null] FROM: voter_Alaska_1
RECEIVED: [I'll vote in candidate: , candidate_1, I'll consider changing my vote if your boss candidate_1 changes the belief: , null, to , null] FROM: voter_Alaska_2
```

O candidato recebe a informação de que os chefes de estado **terminam as sondagens** e pergunta o que deve mudar para angariar mais votos.

```
RECEIVED: Finished FROM: chiefofstaff_0
RECEIVED: Finished FROM: chiefofstaff_1
SENT: What should i change? TO: chiefofstaff_0
SENT: What should i change? TO: chiefofstaff_1
```

## EXEMPLOS DETALHADOS DE EXECUÇÃO

O chefe de estado ouve a pergunta do candidato e envia a informação que este pediu.

```
> INFO:    CANDIDATE candidate_1 IS WINNING IN Alaska  
SENT:     [Winning in , Alaska] TO: candidate_1
```

No fim do processo eleitoral, os votos são contados e é eleito um candidato vencedor.

```
> VOTES: {voter_California_0=candidate_1, voter_California_1=candidate_0, voter_Alaska_2=candidate_1, voter_California_2=candidate_0, voter_Alaska_1=candidate_1, voter_Alaska_0=null}  
> WINNER: candidate_1 VOTES: 3
```



## CLASSES IMPLEMENTADAS

Neste projeto foram implementadas 4 classes.

- **Elections:** classe principal que está encarregue da inicialização, gestão e conclusão de todo o processo eleitoral, assim como da criação dos agentes JADE.
- **Candidate:** agente que representa o candidato.
- **ChiefOfStaff:** agente que representa o chefe de estado.
- **Voter:** agente que representa o eleitor.





## CLASSES IMPLEMENTADAS: CANDIDATE

A classe **Candidate** deriva da classe **Agent** e representa um candidato às eleições.

### ATRIBUTOS

- **credibility**: valor inteiro de credibilidade do candidato [0, 100].
- **beliefs**: hashmap de crença-valor, onde o valor é um inteiro no intervalo [0, 100] e traduz o quanto o candidato acredita na respectiva crença.
- **chiefsOfStaff**: lista de representantes do candidato por estado.

### BEHAVIOURS

- **CandidateListenChiefsFinished**: candidato sabe se o representante de um dado estado acabou de analisar a sua população.
- **CandidateListenChiefStatus**: candidato sabe as estatísticas de cada estado.
- **CandidateSendBeliefs**: candidato envia as suas crenças e credibilidade à população de cada estado, os eleitores.



## CLASSES IMPLEMENTADAS: CHIEF OF STAFF

A classe **ChiefOfStaff** deriva da classe **Agent** e representa um chefe de estado associado a um candidato.

### ATRIBUTOS

- **boss:** variável do tipo **Candidate** que contém o candidato ao qual o chefe de estado está associado.
- **state:** nome do estado que o chefe representa.

### BEHAVIOURS

- **ChiefSendVoterQuestion:** o chefe pergunta à população do seu estado em que candidato é que vão votar.
- **ChiefListenVoterChoices:** o chefe de estado recebe a resposta à pergunta acima.
- **ChiefSendCandidateStatus:** o chefe envia ao candidato que representa quem é que está a ganhar no seu estado.



## CLASSES IMPLEMENTADAS: VOTER

A classe **Voter** deriva da classe **Agent** e representa um eleitor.

### ATRIBUTOS

- **state**: nome do estado ao qual pertence o eleitor.
- **minCredibility**: valor inteiro com a credibilidade mínima permitida, ou seja, o candidato só está apto a conseguir o voto do eleitor se tiver uma credibilidade maior.
- **beliefs**: hashmap de crença-intervalo (range de valores inteiros). O eleitor procura um candidato que tenha cada crença dentro de um certo intervalo.
- **chosenCandidate**: candidato no qual o eleitor vota.

### BEHAVIOURS

- **VoterListenCandidateAndChief**: eleitor recebe a informação dos candidatos e perguntas do chefe do seu estado.
- **VoterSendChiefChoices**: eleitor envia ao chefe do seu estado as suas crenças e o candidato no qual vai votar.