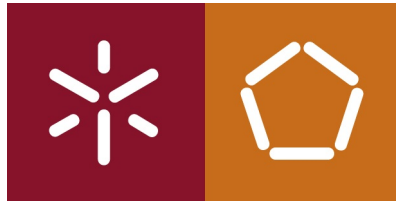


UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA



---

Inteligência Ambiente: Tecnologias e Aplicações

---

## **Inteligência Ambiente com suporte de Processamento de Linguagem Natural**

Guilherme Pereira Martins (A70782)  
André Rodrigues Soares (A67654)  
Hugo Manuel Gomes Nogueira (A81898)  
Joana Esteves Dantas (A84193)

Janeiro 2021

## Conteúdo

1	Introdução	2
2	Estrutura do ChatBot	3
3	Abordagem Análise de Sentimentos	5
4	Abordagem Sugestões	6
5	Abordagem Resumo	7
6	Conclusão	8

---

# 1 Introdução

Para este segundo projeto, foi proposto o desenvolvimento de um sistema inteligente com suporte de processamento de linguagem natural tendo em conta também a componente de um logger para captura de dados de sensores desenvolvida no projecto anterior.

Tem como objectivo criar uma comunicação entre um *ChatBot* e um *User* a partir de um socket. Analisando os dados recolhidos do input de um User, somos capazes de obter uma aproximação da emoção expressa e desenvolver um bot que responde de acordo com a mesma.

---

## 2 Estrutura do ChatBot

O *ChatBot* tem como base a estrutura necessária para a comunicação em socket entre o mesmo e o *User*. Do lado do bot declaramos o servidor do socket com uma *port* definida por nós e de seguida inicializamos a sessão de chat que vai possibilitar a interpretação dos dados relativos ao input do *User*.

```
//create the socket server object
server = new ServerSocket(port);

botname = "empty";

// bot session
String path = "src/Bot";
Bot bot = new Bot(botname, path);
Chat chatSession = new Chat(bot);
```

Figura 1: Inicialização do server e da sessão de chat

Inicializando o servidor, o ChatBot fica então à espera de um input do *User*. Ao receber uma mensagem que é constituída por um objecto *UserInfo* através do socket, ela é dividida em dois (a mensagem de texto e a emoção associada) de maneira a facilitar a sua interpretação e processamento com os ficheiros *.aiml* que contêm o script das conversas possíveis entre as duas entidades.

```
//convert ObjectInputStream object to UserInfo
UserInfo userInfo = (UserInfo) ois.readObject();
//split message and emotion part:
String message = userInfo.getChatString();
String emotion = userInfo.getEmotion();
// Set emotion
chatSession.multisentenceRespond( request: "EMOTION "+emotion);
```

Figura 2: Processamento da mensagem

---

Após o processamento da mensagem, o *ChatBot* obtém através de uma busca aos ficheiros *.aiml* uma possível resposta a enviar ao *User* novamente através do socket. Este ciclo repete-se até ser recebida uma mensagem com o texto "adeus", que desencadeia o termino da sessão e o fecho do socket.

```
socket.close();  
//terminate the server if user says "adeus"  
if(message.equalsIgnoreCase( anotherString: "adeus")) break;  
}  
System.out.println("Shutting down Socket Bot.ChatBot!");  
//close the ServerSocket object  
server.close();
```

Figura 3: Finalização de sessão de chat

Em seguida é apresentada a estrutura geral do ChatBot:

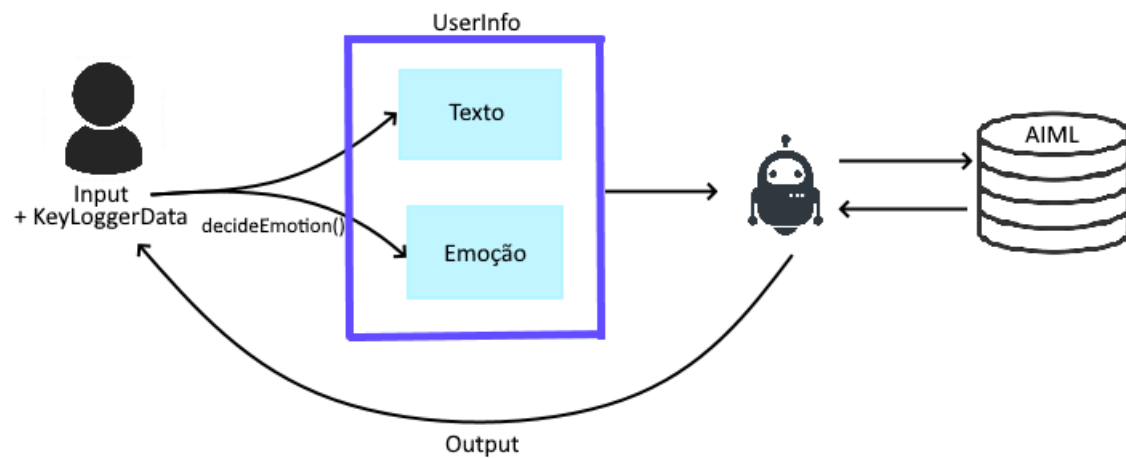


Figura 4: Estrutura ChatBot

---

### 3 Abordagem Análise de Sentimentos

O utilizador comunica com o Bot através de Strings enviadas através de um socket, como já anteriormente discutido. Antes da mensagem ser enviada é feita a análise de métricas do keylogger e da String que o utilizador escreveu para decidir a emoção que o utilizador está a transmitir.

No nosso trabalho resolvemos colocar 5 emoções distintas, a maioria delas são detetadas através do texto mas de forma a utilizar software utilizado na primeira fase do trabalho, mais concretamente na deteção de dados de sensores como o teclado, esta emoção pode ser influenciada pela número de vezes que foi clicada uma tecla ou o tempo demorado a escrever.

Para a ajuda na detecção de palavras associadas a um certo sentimento, criamos dois ArrayLists com dezenas de palavras associadas a emoções positivas e negativas e na através de um ciclo fazemos a comparação das palavras existentes no input do utilizador com as palavras desses arrays.

Em seguida alguns exemplos de como detetamos emoções:

- **Triste**

Através da existência de palavras como triste, desanimado, desagradado, etc

- **Feliz**

Através da existência de palavras como feliz, contente, agradado, etc

- **Stressado**

Através do número elevado de backspaces e caso o texto esteja todo em maiúsculas

- **Distraído**

Através da da medição do tempo demorado pelo utilizador a escrever o texto

- **Neutral**

Caso nenhuma das opções anteriores se tenha verificado

Tentamos também evitar reconhecer certa emoções caso a frase as negue. Por exemplo a frase "Não estou desanimado" apesar de ter uma palavra normalmente associada a um sentimento negativo deverá ser associada a uma emoção neutra.

---

## 4 Abordagem Sugestões

Um dos requisitos do enunciado era que quando fosse feita uma comunicação com o ChatBot a resposta deste fosse dada como uma sugestão passível de ser editada. Desta forma quando o ChatBot envia a sua resposta que foi buscar ao ficheiro AIML, esta resposta tem que ser confirmada pelo utilizador através da tecla y ou rejeitada com a tecla n:

```
Conversa com o Bot.ChatBot!  
>ola  
Estado: neutral  
Premir y/n para aceitar: [sugestão] Ola para ti tambem.  
|
```

---

Figura 5: Sugestão inicial

Caso seja aceite vamos ser informados que a resposta final é de facto a sugerida e podemos então continuar a falar com o bot:

```
>ola  
Estado: neutral  
Premir y/n para aceitar: [sugestão] Ola para ti tambem.  
y  
Bot.ChatBot: Ola para ti tambem.  
>nova questão|
```

Figura 6: Sugestão aceite

Caso seja rejeitada (através da tecla n) é pedido ao utilizador uma nova resposta, que gostaria que fosse a ideal para o input decidido:

```
>ola  
Estado: stressado  
Premir y/n para aceitar: [sugestão] Ola, vejo que estas stressado.  
n  
Que resposta preferia que o ChatBot desse?  
Ola, entao o que contas?  
Bot.ChatBot: Ola, entao o que contas?  
>|
```

---

Figura 7: Sugestão rejeitada

---

## 5 Abordagem Resumo

A parte de resumo do nosso projeto é apenas um log da conversa. Por uma questão de facilidade a analisar dados e a experimentar coisas ao longo do projeto decidimos imprimir para um ficheiro em vez de o fazer na linha de comandos no final da utilização.

Desta forma é impresso o input do utilizador em cada interação com o bot, a emoção associada a esse input e ainda a resposta do ChatBot. Caso a resposta seja rejeitada na parte da sugestão pelo utilizador então será impressa a nova resposta que o utilizador prefere que seja utilizada:

```
User [emotion]: input;
ChatBot: output;
-----

User [neutral]: ola;
ChatBot: Ola para ti tambem.;
-----
User [neutral]: tudo bem?;
ChatBot: Sim, esta tudo bem comigo! E contigo?;
-----
User [neutral]: também!;
ChatBot: Ainda bem!;
-----
User [neutral]: Quem és tu?;
ChatBot: Eu sou ChatBot! Aqui para falar contigo.;
-----
User [neutral]: Prazer em conhecer-te;
ChatBot: Igualmente;
-----
User [neutral]: Gostas de gelado?;
ChatBot: É possível, nunca tinha pensado muito em gelado.;
-----
User [feliz]: é incrivelmente bom!;
ChatBot: Tenho que provar;
-----
User [distraindo]: sim;
ChatBot: OK então, já percebi que estas distraido.;
-----
User [neutral]: peço desculpa;
ChatBot: Não faz mal;
-----
User [neutral]: tenho que me ir embora;
ChatBot: Ok, vai lá;
-----
User [neutral]: adeus;
ChatBot: Adeus! Até a próxima.;
-----
```

Figura 8: Exemplo de um ficheiro *log* gerado



---

## 6 Conclusão

Numa primeira fase foi estudado o enunciado do trabalho prático disponibilizado para compreender melhor o que nos era proposto fazer. De seguida, o grupo foi trocando ideias de modo a encontrar qual a melhor abordagem para a concepção da resolução do trabalho prático. Inicialmente, o grupo tentou realizar o trabalho usando a linguagem de programação *Python* mas, devido à inexperiência dos elementos nessa mesma linguagem, acabou-se por usar a linguagem de programação *Java* para a concepção do mesmo. Outra razão para a mudança de linguagem de programação, foi o facto de estarem a ser usados *packages* e funções já pré definidas para efetuar todo o trabalho pretendido para um *Chatbot* em inglês. Desta forma, ficou um trabalho com um *Chatbot* em português muito mais simples mas, ao mesmo tempo, mais autêntico e, no nosso entender, mais perto do que era pretendido.

Em suma, o grupo pensa que abordou tudo o que era pretendido no enunciado do trabalho prático e que o trabalho satisfaz os requisitos pretendidos, pretendendo mesmo assim no futuro melhorar as funcionalidades do mesmo como por exemplo, melhorar a parte da análise de sentimentos, tornando a sua análise um mais completa bem como adicionar mais sentimentos.