

E-book informativo

MandraKinho *chatbot*

Desenvolvido por:

Bruno Gallo de Almeida Passos

Eduardo de Siqueira Nardoni

Eliel Ribeiro dos Santos de Andrade

José Carlos de Lima Júnior

Nathan Dantas Nascimento

Pedro Henrique O. da Anunciação

Universidade Anhembi Morumbi, São Paulo.

Sumário

1. Projeto ChatBot: Mandrakinho	4
2. Bot Mandrakinho.....	4
3. Complementações no código-fonte	5
4. Observações importantes	6

1. Projeto ChatBot: Mandrakinho

Ao iniciar o chat com Mandrakinho, o usuário recebe algumas informações sobre possíveis interações com o bot e de ajuda, o usuário pode digitar “*interações*” ou “*/ajuda*” e algumas informações extras são compartilhadas, como por exemplo, quais palavras que ao serem digitadas o Mandrakinho reconhece ou adicionar respostas ao algum banco de respostas pré-existente, dependendo do que é solicitado.

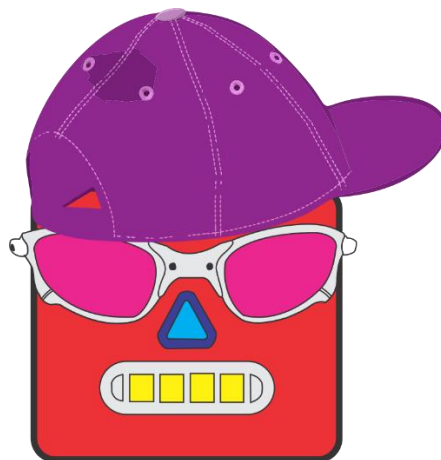
Antes de iniciar a conversa com o bot, a plataforma solicita ao usuário que digite seu nome de usuário, removendo espaços contidos no texto, caso haja algum.

```
-----  
= Iniciando o chatbot com Mandrakinho =  
-----  
# Para saber algumas palavras chaves já conhecidas, digite 'Interações'  
# Para saber como adicionar respostas ao chatbot, digite '/ajuda'  
  
# Para começar, digite seu nome de usuário: usuário 123  
  
Iniciando a conversa.  
  
> @usuário123:
```

1.1. Console: Iniciando uma conversa com Mandrakinho

2. Bot Mandrakinho

Mandrakinho é um verdadeiro cidadão de São Paulo, que vive e conhece sua cidade. A linguagem tipicamente paulista, é reconhecida assim que se inicia uma nova conversa, com muitas gírias e dialetos locais. Mandrakinho é um bot muito engraçado, divertido, educado e prestativo. As vezes meio mal-humorado, por causa da “correria”, mas não é só isso, ele também é muito inteligente, certamente o usuário pode aprender muito com Mandrakinho.



2.1. Foto Bot Mandrakinho

3. Complementações no código-fonte

Para obtermos originalidade no desenvolvimento do ChatBot, adicionamos algumas funcionalidades extras, a fim de manter e desenvolver uma comunicação diferente e satisfatória entre o usuário e o nosso bot Mandrakinho.

A primeira mudança para o projeto inicial é que não é necessário adicionar respostas e palavras chaves ao nosso ChatBot. Ele já vem pré-programado com um banco de respostas pronto para as situações propostas no projeto.

Caso o usuário queira adicionar novas palavras e respostas ao nosso bot, é possível através de comandos já estabelecidos no código.

Aqui, temos alguns exemplos de funções/métodos que implementamos para facilitar e melhorar a experiência do usuário com o ChatBot.

```
public ChatBot() {..}
```

```
public void adiciona(Resposta resposta) {..}
```

```
public void processa(String texto) {..}
```

```
public void adicionarPiada(String piada) {..}
```

```
public void adicionarCuriosidade(String curiosidade) {..}
```

```
public void adicionarRespostaAlternativa(String resposta) {..}
```

```
public void bancoDePiadas() {..}
```

```
public void bancoDeCuriosidades() {..}
```

```
public void bancoDeRespostasSimples() {..}
```

```
public void bancoDeRespostasAlternativas() {..}
```

```
public void bancoDeRespostas() {..}
```

3.1. Construtor e métodos da *Classe ChatBot*

Descrição da imagem:

- Verde – Construtor ChatBot;
- Vermelho – Métodos base do projeto ChatBot;
- Azul – Novos métodos criados e adicionados ao projeto original;

4. Observações importantes

Como nosso objetivo é fornecer uma experiência diferente e mais completa ao usuário, fizemos alguns complementos na *Classe RespostaData*.

Além de responder a data presente, Mandrakinho também informa ao usuário o dia da semana correspondente a data fornecida. Isto é possível, graças aos métodos extras de formatação de data e dia da semana que adicionamos ao código-fonte, onde podemos ver na figura 4.1, destacado em Roxo:

```
public RespostaData(String palavraChave) {  
  
public boolean verifica(String entrada) {  
  
public String produz() {  
  
public String diaSemana() {  
public String data() {
```

4.1. Construtor e métodos da *Classe RespostaData*

Na *Classe Main*, onde rodamos o programa, optamos por manter somente as linhas de comando de impressão no console e de entrada de texto:

```
public static void main(String[] args) throws InterruptedException {  
  
    ChatBot bot = new ChatBot();  
  
    System.out.println("-----");  
    System.out.println("= Iniciando o chatbot com Mandrakinho =");  
    System.out.println("-----");  
    System.out.println(" # Para saber algumas palavras chaves já conhecidas, digite 'Interações'");  
    System.out.println(" # Para saber como adicionar respostas ao chatbot, digite '/ajuda'\n");  
  
    Scanner input = new Scanner(System.in);  
  
    System.out.print("\n # Para começar, digite seu nome de usuário: ");  
    String usuario = input.nextLine();  
    usuario = usuario.replace(" ", "");  
    System.out.println("\nIniciando a conversa...\n");  
  
    String texto = "";  
    while(true){  
        Thread.sleep(1000);  
        System.out.print("> @" + usuario + ": ");  
        texto = input.nextLine();  
        Thread.sleep(1500);  
        bot.processa(texto);  
    }  
}
```

4.2. *Classe Main*

Para passar uma melhor sensação de conversa, adicionamos um tempo de espera entre a mensagem enviada pelo usuário e a resposta do Mandrakinho. Utilizamos *Thread.sleep()*, para adicionar um tempo pré-definido entre as

mensagens do usuário e do Mandrakinho, como podemos observar dentro do laço de repetição *while* na *Classe Main* na figura 4.2.

Para completar o código, pensamos também em uma forma de encerrar a conversa com Mandrakinho e fazer com que o funcionamento do programa seja interrompido.

Ao digitar “tchau” ou “falow”, Mandrakinho se despede do usuário com uma mensagem amigável, informando o fim da conversa e o interrompendo o funcionamento do programa. Caso o usuário queira iniciar uma nova conversa, é só rodar o programa novamente.

Também facilitamos a identificação da mensagem, utilizando o nome do usuário e o nome de usuário do Mandrakinho, ao trocarem mensagens no console.

```
<terminated> Main (6) [Java Application] C:\Users\than_\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_
```

```
-----  
= Iniciando o chatbot com Mandrakinho =  
-----
```

```
# Para saber algumas palavras chaves já conhecidas, digite 'Interações'  
# Para saber como adicionar respostas ao chatbot, digite '/ajuda'
```

```
# Para começar, digite seu nome de usuário: usuário123
```

```
Iniciando a conversa...
```

```
> @usuário123: Oi!  
> @mandrakinho157: Qual a boa?  
> @usuário123: tudo bem?  
> @mandrakinho157: Tudo em cima, e aí?  
> @usuário123: tudo bem também  
> @mandrakinho157: Maravilha meu bom, só progresso  
> @usuário123: tchau!  
> @mandrakinho157: Valeu amigo, foi bom trocar aquela ideinha, até a próxima
```

```
Fim da conversa.
```

4.3. Demonstração de fim de conversa com Mandrakinho