

# Chatbot Integrado ao Telegram

Java Platform

Turma 36SCJ  
2020

Aruna Fernanda Martins	– RM 338577
Ayrton Henrique Gomes Silva	– RM 337089
Carlos Eduardo Roque da Silva	– RM 338866
Sara Regina Pires	– RM 338142
Willian Yoshiaki Kazahaya	– RM 338950

## 1. Introdução e apresentação

O presente trabalho tem o intuito de apresentar a solução adotada para a criação de um protótipo de chatbot integrado com o aplicativo Telegram. Colaborativamente foi decidido criar um chatbot que simulasse uma Pizzaria Online, a Pizzaria FIAP.

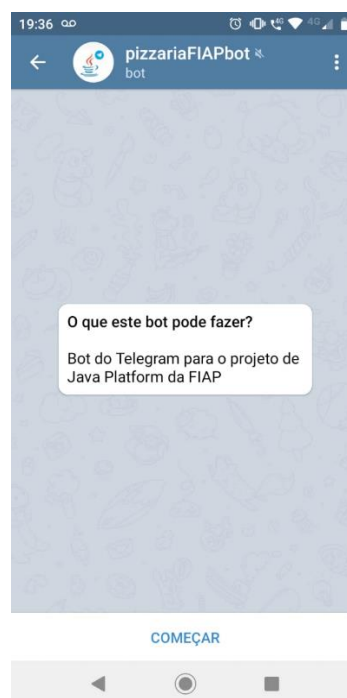
A ideia inicial era de que o chatbot fosse capaz de interagir com um usuário do Telegram que desejasse fazer um pedido de pizza emulando uma Pizzaria online e um atendente online, e que fosse possível controlar a conversa entre os mesmos de maneira automática, onde o chatbot buscava ter uma postura o mais humana possível para ajudar o usuário do Telegram a realizar seu pedido da pizza e concluí-lo sem intercorrências.

## 2. Estrutura básica do chatbot

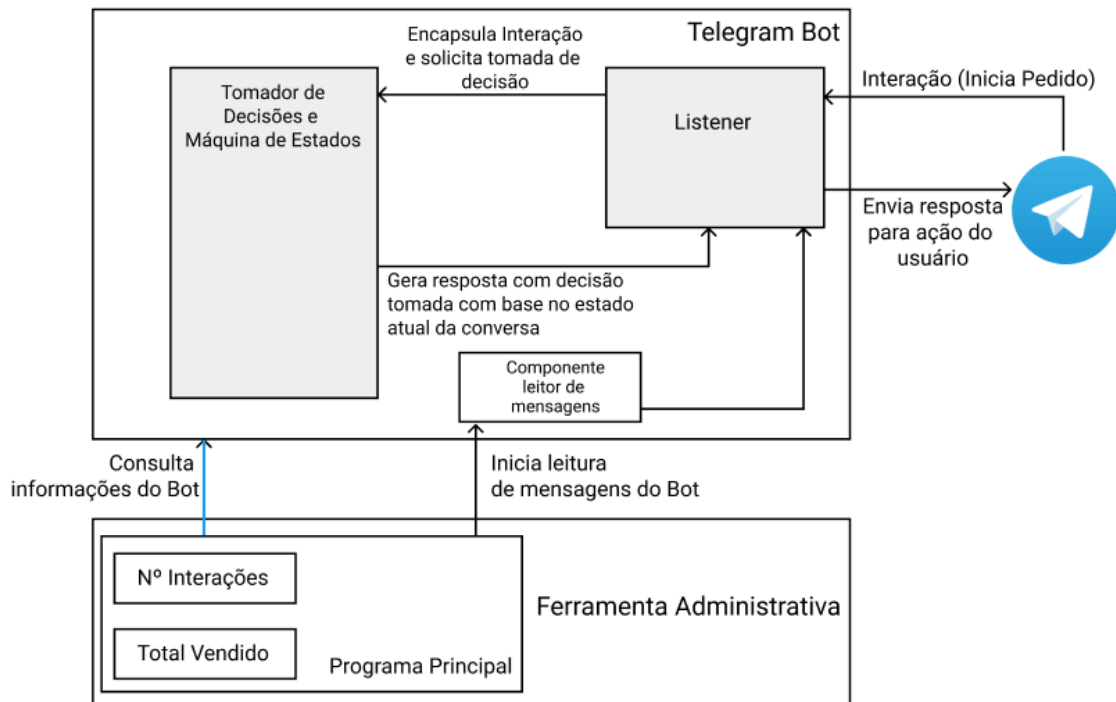
### O usuário-chatbot criado no Telegram

Para a interação com o Telegram foi necessário criar um chatbot padrão usando as próprias ferramentas de chatbots do Telegram (ferramenta BotFather).

Com esta ferramenta, foi criado um Bot (chamado pizzariaFIAPbot) para representar o atendente da pizzaria. Com o bot criado, os usuários poderiam invocar o bot através dos seus comandos padrão e dos comandos customizados na aplicação.



## Estruturação básica da aplicação



## 3. Linguagem, código-fonte e bibliotecas

A solução foi estruturada utilizando a linguagem Java e orientação a objetos. O código fonte foi mantido na ferramenta GitHub. A URL para a solução no GitHub é:



<https://github.com/AyrtonHenrique/bot-fiap-java>

Foi utilizada nesta solução a biblioteca Java Telegram Bot API, disponibilizada também no GitHub na URL abaixo:



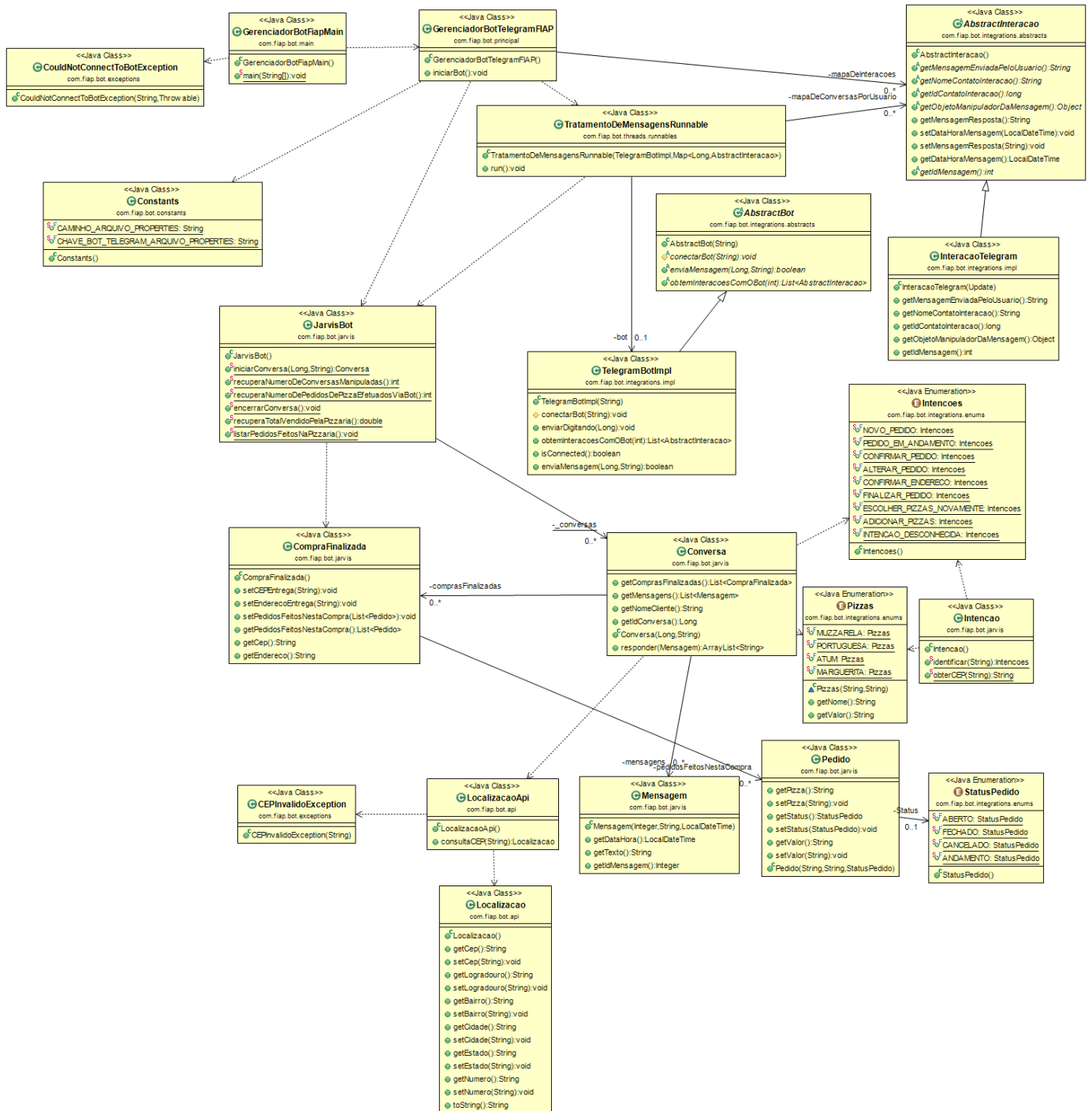
<https://github.com/pengrad/java-telegram-bot-api>

O projeto utiliza o Maven para obtenção das dependências a partir do pom.xml. Ele também possui um arquivo de propriedades no diretório **properties** chamado **config.properties**. Neste arquivo está a chave de conexão provida pelo Telegram Botfather.

Por fim, ele utiliza a biblioteca **gson-2.7.jar** para invocar e tratar o JSON enviado pela API de endereço.

O programa principal com o método `main()` está representado pela classe **com.fiap.bot.main.GerenciadorBotFiapMain**.

## 4. Modelo de classes



## 5. Mensagens tratadas pelo Bot

Este Bot está preparado para reconhecer algumas mensagens e comandos, dentre eles:

/start → Determina o início da interação com o Bot. Ele faz a saudação e inicia a conversa com o interlocutor

/mussarela, /portuguesa, /atum, /marguerita → São mensagens de escolha dos produtos por parte do interlocutor

/confirmarPedido → Determina a intenção do interlocutor de confirmar e proceder com a finalização do pedido

/alterarPedido → Determina a intenção do interlocutor de alterar um pedido em andamento

/finalizarPedido → Determina a confirmação do usuário do pedido feito e inicia o status de verificação do endereço do interlocutor

/escolherNovamente → Determina a intenção do interlocutor de escolher novamente os produtos a partir do zero

Caso alguma outra mensagem seja direcionada a ele, ele tentará direcionar o interlocutor para o processo normal do pedido.

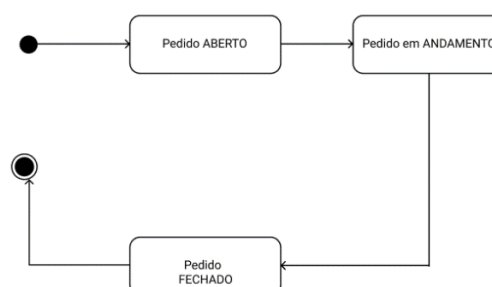
## 6. Máquina de estados dos pedidos

Quando o contato do Telegram inicia uma conversa, o Bot o leva para o início da máquina de estados. Esta máquina de estados permite ao contato caminhar pelos status possíveis do pedido, podendo em qualquer momento cancelar o pedido, exceto quando o pedido está finalizado.

Os estados manipulados pelo Bot são:

- ABERTO → Definido quando a conversa é iniciada e o Bot inicia a interação com o usuário para que ele comece a escolher os produtos
- ANDAMENTO → Definido quando o usuário escolhe o primeiro produto. Aqui ele pode alterar um pedido, confirmá-lo, finalizá-lo ou recomeçá-lo.
- FECHADO → Definido quando o usuário escolhe finalizar o pedido e proceder com o pagamento

Máquina de Estados dos Pedidos



## 7. Regex para reconhecimento do CEP e integração com consulta de endereços na internet a partir do CEP

Ao final do processo de escolha dos produtos, o interlocutor confirma o pedido e a ele é solicitado pelo Bot o CEP para o envio dos produtos. Neste ponto o Bot consulta uma API na internet e recupera o endereço do interlocutor a partir do CEP enviado. O Bot também valida por Regex se o CEP digitado é válido, e interage com o interlocutor para que ele digite um CEP válido.

## 8. Aplicação administrativa e Javadocs

Foi criada uma aplicação administrativa simples que, além de iniciar o processo do chatbot, também permite exibir informações relativas às interações com o Bot através de um menu administrativo.

O objetivo foi demonstrar que é possível saber em tempo real informações sobre o Bot enquanto elas estão ocorrendo e expô-las em um sistema paralelo, em uma API na internet ou intranet, ou mesmo numa aplicação web efetiva.

```
*****
Gerenciador do Bot do Telegram da Pizzaria FIAP
*****
Iniciando conexão com o Bot do Telegram...
Recuperando arquivos de configurações...
Arquivo de configuração recuperado.
Recuperando chave de conexão com o Bot do Telegram do arquivo de configuração:
Chave recuperada.
Conectando ao Bot do Telegram. Por favor aguarde...
Conexão com o Bot realizada com sucesso. Bot Conectado: true
Iniciando Thread de tratamento de mensagens...
Thread de tratamento de mensagens iniciado.
Todos os componentes carregados...
```

```
*****
Gerenciador do Bot do Telegram da Pizzaria FIAP
*****
Escolha uma das opções para iniciar:
```

1. Consultar número de interlocutores com conversas tratadas pelo Bot
2. Consultar número de pizzas vendidas na pizzaria pelo Bot
3. Consultar total vendido pela pizzaria hoje pelo Bot
4. Listar os pedidos feitos na pizzaria atendidos pelo Bot
5. Sair

1

O número de conversas com interlocutores tratados pelo Bot é de: 1

2

O número de pizzas vendidos pelo Bot é de: 2

3

O valor total dos pedidos feitos por usuários tratados pelo Bot é de: R\$ 66,20

4

-----  
 Cliente: Carlos Eduardo  
 Código da conversa: 1145422759  
 Pedido realizado:

-----  
 Compra realizada para o endereço: Rua Doutor Sérgio Meira - Barra Funda São Paulo/SP  
 CEP.:01153010

PORTUGUESA - 35.60

Compra realizada para o endereço: Rua Melo Nunes - Americanópolis São Paulo/SP  
 CEP.:04337120

MARGUERITA - 30.60  
 -----

O Javadoc do projeto pode ser encontrado junto ao código-fonte, dentro do diretório **doc**. Também há um diretório chamado **diagrams** onde é possível encontra o diagrama de classes do projeto.

## 9. Conclusão

O uso de robôs inteligentes tem se espalhado no mundo atual. Com o uso de IA, em breve será possível que não identifiquemos mais se que em interage conosco é uma pessoa ou um robô através da internet.

Isso possibilita que informações sejam captadas, armazenadas, manipuladas e consolidadas totalmente por robôs permitindo a tomada de decisões rapidamente e a redução de custos para trabalhos repetitivos, possibilitando a alocação de pessoas em tarefas mais críticas e decisórias.

Este trabalho demonstra que rapidamente hoje é possível construir robôs que conversam com usuários reais na internet e podem gerar muito valor para pessoas e corporações.