

FIAP

Telegram Bot Digital Bank

São Paulo, SP.

Agosto de 2018.
FIAP

Daniel Souza de Lima RM331883
David Santos Vasconcelos RM330191
Jean Pierre de Brito RM331429
Luiz Felipe Moreira do Amaral RM330342

Telegram Bot Digital Bank

São Paulo, SP.
Agosto de 2018.

Sumário

1. INTRODUÇÃO	5
2. COMPONENTES E FRAMEWORKS	6
MAVEN	6
JAVA API FOR TELEGRAM	6
PREVLAYER	6
JAVA LOGGING	6
3. ESTRUTURA DO PROJETO	6
FIAP.TELEGRAM	6
FIAP.TELEGRAM.BOT	6
FIAP.TELEGRAM.COMMAND	6
FIAP.TELEGRAM.COMPARATOR	7
FIAP.TELEGRAM.CONSTANTS	7
FIAP.TELEGRAM.MANAGER	7
FIAP.TELEGRAM.MODEL	7
FIAP.TELEGRAM.PREVAYLER	7
FIAP.TELEGRAM.UTILS	7
4. DIAGRAMAS	8
1. DIAGRAMA DE CLASSE	8
2. DIAGRAMA DE SEQUÊNCIA	8
5. PRÉ-REQUISITOS	9
• TECNOLOGIAS NECESSÁRIAS.	9
• DADOS DO BOT	9
6. UTILIZAÇÃO DO BOT	10
FUNCIONALIDADES	10
• PARA UTILIZAÇÃO DO BOT, ENVIAR O COMANDO /START PARA O BOT COM O NOME @SHELDONFIAP_BOT.	10
• TELA DE BOAS-VINDAS DO BANCO	10
• CRIAÇÃO DE CONTA	10
• MODIFICAÇÃO DE CONTA	10
• INCLUSÃO DE DEPENDENTES (CONTA-CONJUNTA)	10
• EXIBIÇÃO DOS DADOS DO TITULAR E DEPENDENTES	10
• DEPÓSITO	10
• SAQUE (CUSTO DO SERVIÇO R\$ 2,50)	10
• SOLICITAÇÃO DE EXTRATO (CUSTO DO SERVIÇO R\$ 1,00)	10
• SOLICITAÇÃO DE EMPRÉSTIMO, CUJO PRAZO MÁXIMO É DE 3 ANOS E VALOR MÁXIMO É DE 40 VEZES O SALDO DA CONTA (CUSTO DO SERVIÇO R\$ 15,00 ALÉM DE JUROS DE 5% A.M.)	10
• EXIBIÇÃO DE SALDO DEVEDOR DO EMPRÉSTIMO E PRAZO DE PAGAMENTO	10
• EXIBIÇÃO DOS LANÇAMENTOS DETALHADA, COM SOMATÓRIA AO FINAL	10
• EXIBIÇÃO DAS RETIRADAS, COM SOMATÓRIA AO FINAL	10
• EXIBIÇÃO DAS TARIFAS DE SERVIÇO, COM SOMATÓRIA AO FINAL DOS SERVIÇOS QUE JÁ FORAM UTILIZADOS NA CONTA	10
• TELA DE AJUDA	10
LISTA DE COMANDOS CONFORME IMAGEM ABAIXO.	11
• UTILIZANDO UM BOT NOVO(OPCIONAL)	12
7. RESUMO TÉCNICO DE FUNCIONAMENTO	12

8. GITHUB	12
9. REFERÊNCIAS	12

1. Introdução

Com o avanço da tecnologia móvel no mundo inteiro, explorando, dentre outras formas pelos comunicadores instantâneos aliado aos assistentes virtuais que tem se mostrado cada vez mais inteligentes e ajudando as pessoas a cumprirem suas tarefas do cotidiano estabelecendo uma plataforma de serviços de autoatendimento, com possibilidades também abertas as empresas.

Os bancos financeiros, por sua vez, estão nascendo ou migrando para a realidade digital, onde não é obrigatório ter agências físicas para obter seus principais serviços. Com o acesso massificado da internet, é possível realizar desde a abertura da conta e transações de qualquer natureza remotamente.

Por outro lado, os bots, abreviação de robots, tem sido utilizado nos principais comunicadores de mensagens, como o Telegram e Facebook. Sua facilidade e flexibilidade de integração permite utilizá-los como assistentes pessoais para resolução de diversas questões, tais como:

Obter segunda via de contas, solicitação de serviços e solicitação de informações. Os bots substituem o atendimento humano pelo virtual, mais efetivo e, podendo ser explorado para ser adaptado a cada tipo de cliente.

O projeto pode ser acessado pelo github em.

<https://github.com/daniesouza/chatbot-fiap-telegram>.

2. Componentes e Frameworks

Maven

O Maven é uma ferramenta para gerenciamento das bibliotecas e plug-ins de um ou mais repositórios de forma dinâmica utilizando um XML para descrever o projeto de software sendo construído.

Java API for Telegram

O Java API for Telegram possibilita o desenvolvimento de um bot personalizado conforme a necessidade do projeto.

Website da API:

<https://oss.sonatype.org/content/repositories/releases/com/github/pengrad/java-telegram-bot-api/2.1.2/>

Prevlayer

Api de persistência usado para persistir os dados do usuário/cliente em arquivo conforme documento de requisitos.

<http://prevlayer.org/>

<https://mvnrepository.com/artifact/org.prevayler>

Java Logging

Ferramenta padrão de log do java em Console.

<https://docs.oracle.com/javase/7/docs/api/java/util/logging/package-summary.html>

3. Estrutura do Projeto

fiap.telegram

- Pacote principal com os arquivos do projeto.

fiap.telegram.bot

- Pacote com a inteligência do Bot para buscar as mensagens do telegrama.

fiap.telegram.command

- Pacote com todas as classes de comandos a serem executados.

fiap.telegram.comparator

- Pacote com o comparador de cliente.

fiap.telegram.constants

- Pacote de constantes

fiap.telegram.manager

- Pacote com os gerenciados de envio de mensagens ao telegram e gerenciamento da sessão do cliente.

fiap.telegram.model

- Pacote com a camada model

fiap.telegram.prevalayer

- Pacote prevlayer de persistência em arquivo.

fiap.telegram.utils

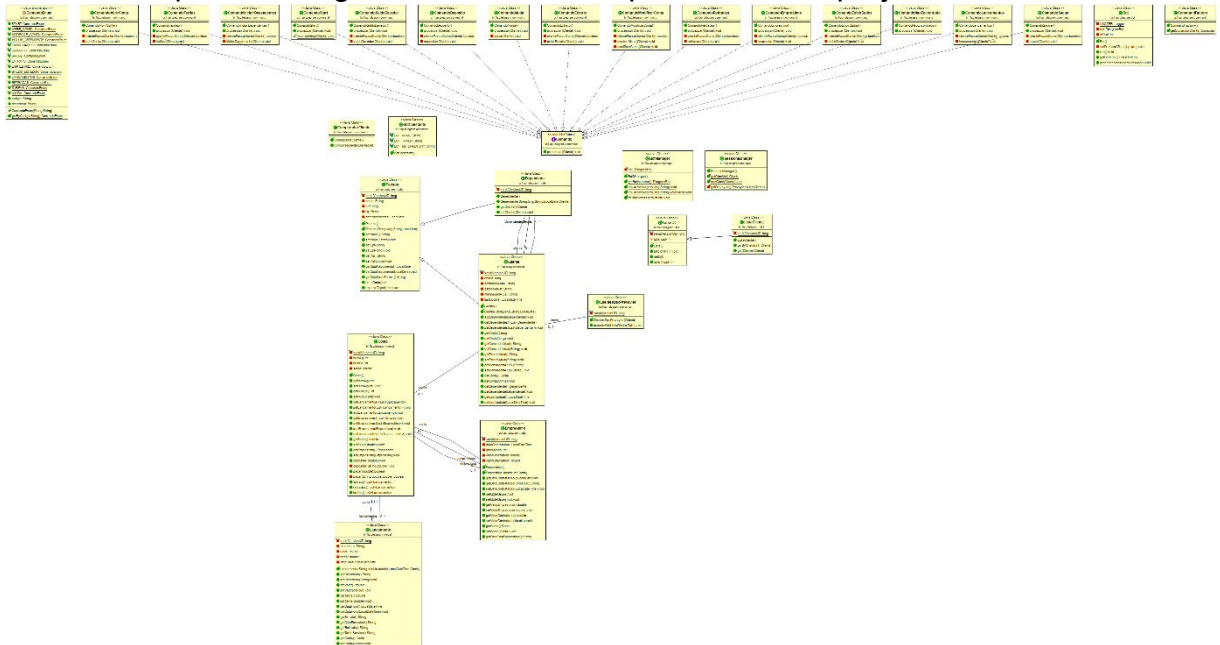
- Pacote de Utilidades.

4. Diagramas

1. Diagrama de Classe

A seguir é representado o diagrama de classe do projeto.

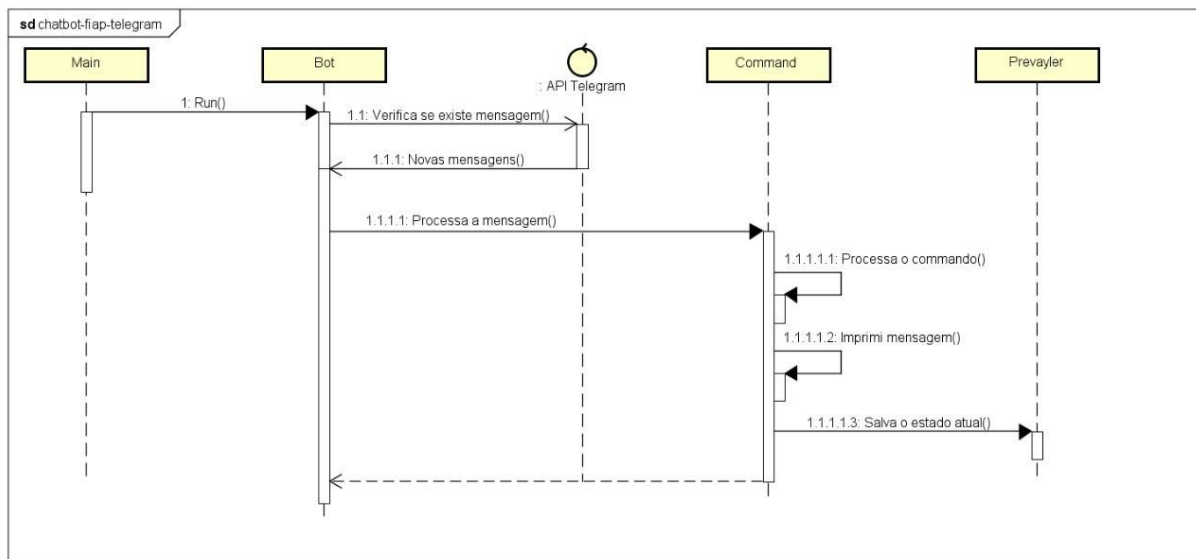
Obs: print do diagrama localizado na pasta docs do projeto.



2. Diagrama de Sequencia

A seguir é representado o diagrama de sequência do projeto.

Obs: print do diagrama localizado na pasta docs do projeto.



5. Pré-requisitos

- Tecnologias necessárias.
 - Para executar o projeto é necessário ter o JRE 8 ou JDK 8 instalado.
 - Utilizar uma ide para execução. (Eclipse é recomendado)
- Dados do Bot

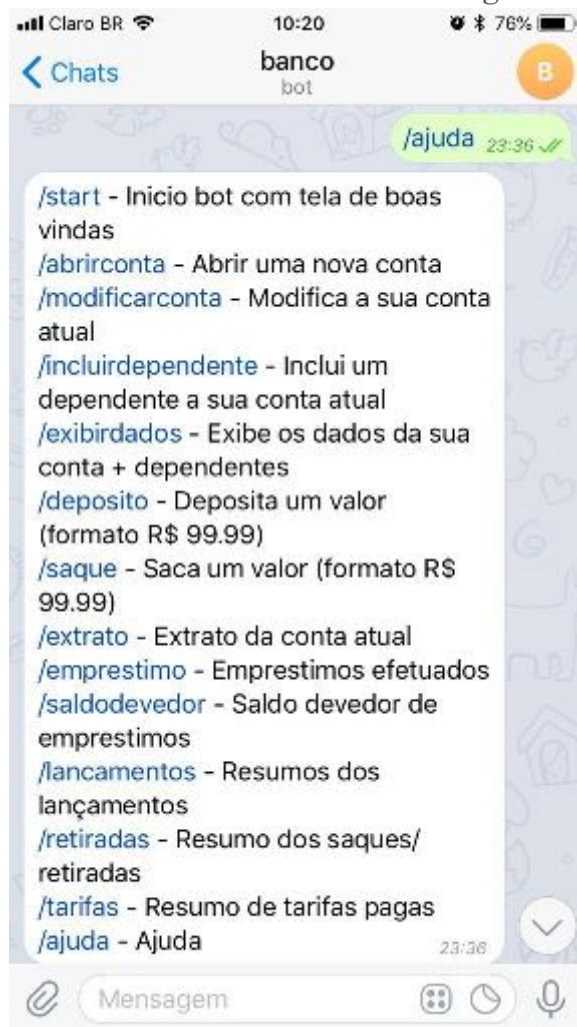


6. Utilização do Bot

Funcionalidades

- Para Utilização do bot, enviar o comando /start para o bot com o nome @sheldonfiap_bot.
- Tela de boas-vindas do banco
- Criação de conta
- Modificação de conta
- Inclusão de dependentes (conta-conjunta)
- Exibição dos dados do titular e dependentes
- Depósito
- Saque (custo do serviço R\$ 2,50)
- Solicitação de extrato (custo do serviço R\$ 1,00)
- Solicitação de empréstimo, cujo prazo máximo é de 3 anos e valor máximo é de 40 vezes o saldo da conta (custo do serviço R\$ 15,00 além de juros de 5% a.m.)
- Exibição de saldo devedor do empréstimo e prazo de pagamento
- Exibição dos lançamentos detalhada, com somatória ao final
- Exibição das retiradas, com somatória ao final
- Exibição das tarifas de serviço, com somatória ao final dos serviços que já foram utilizados na conta
- Tela de ajuda

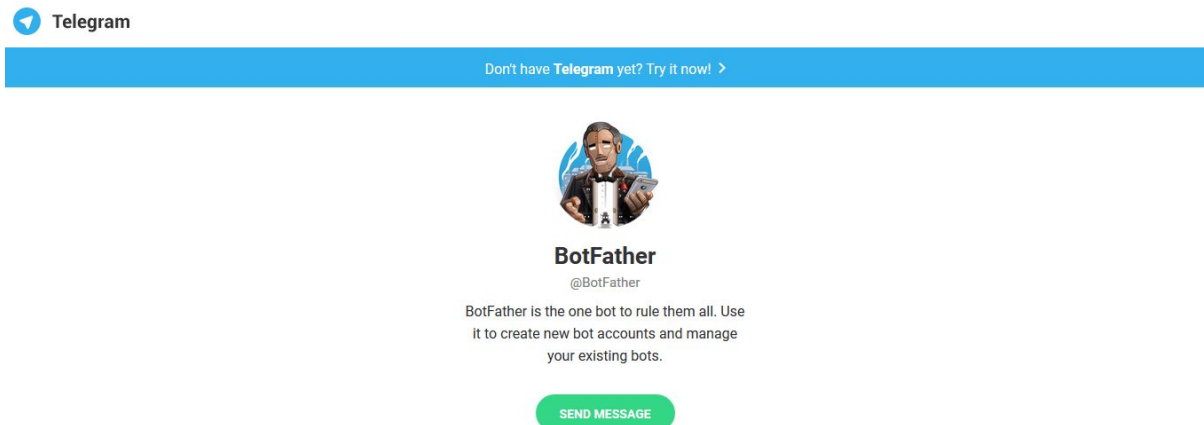
Lista de comandos conforme imagem abaixo.



- Utilizando um bot novo(opcional)

Para o desenvolvimento do Bot, é necessário solicitar um token para o @BotFather, um bot do próprio Telegram para criação e gestão dos Bots.

A solicitação é feita através do endereço <https://telegram.me/BotFather> mostrado a seguir.



7. Resumo técnico de funcionamento

A partir da classe Bot recuperamos as mensagens represadas da API do Telegram, verificamos se o cliente já está salvo através da classe SessionManager que por sua vez, verifica e grava um cliente através da camada de Prevayler.

Na camada de command, instanciamos um comando através da [CommandFactory](#), que verifica se o comando vindo do através da digitação do usuário é valido, ou seja, se está listado no Enum de comandos [ComandoEnum](#) e o instancia para ser executado.

Cada comando corresponde a uma funcionalidade do sistema, verificando em qual etapa o cliente está, devolve uma informação para a Interface [Comando](#) e grava o estado a cada interação com o usuário, no final, retorna as opções de menu principal para a API do Telegram e devolve o resultado para o usuário.

8. Github

O projeto pode ser acessado através do GitHub pelo endereço

- <https://github.com/daniesouza/chatbot-fiap-telegram>

9. Referências

- Blog Medium – Prof - [Michel Fernandes](#)

<https://medium.com/@michelpf/criando-bots-do-telegram-com-api-java-64e921762b5f>

- Telegram Documentação oficial

<https://core.telegram.org/bots>

<https://core.telegram.org/bots/api>