Desenvolvimento de um *framework* para aplicações robóticas utilizando o aplicativo Telegram

Isabelle Nedilha Okuma, Dierone César Foltran Júnior, e-mail: dcfoltran@uepg.br.

Universidade Estadual de Ponta Grossa/Departamento de Informática.

Sistemas de Computação, Teleinformática.

Palavras-chave: *Application Programming Interface* (API), *chatbot* e *framework*.

Resumo

O aplicativo Telegram possui recursos interessantes designados para desenvolvedores, como a API (*Application Programming Interface*), essa que permite a criação de aplicativos baseados no código original. O objetivo dessa pesquisa foi estudar conceitos relacionados ao uso da API de aplicativos de mensagens instantâneas e desenvolver um *chatbot* utilizando a API do aplicativo Telegram. O *chatbot* é um programa de computador que tenta simular o comportamento humano com objetivo de responder mensagens de forma semelhante aos mesmos. Existem diversos métodos disponibilizados na API do Telegram que podem ser utilizados para o desenvolvimento do *chatbot*. Para facilitar a comunicação entre o Telegram e estes métodos, foi utilizado o módulo Telepot.

Introdução

Com a chegada do *smartphone* e outros dispositivos móveis, houve um grande crescimento no uso de aplicativos de mensagens instantâneas. Estes aplicativos vêm sendo grandes facilitadores para a comunicação diária e viraram parte da rotina de muitas pessoas. São exemplos de aplicativos de mensagens instantâneas o WhatsApp, Facebook Messenger, Telegram, Slack, entre outros. (KOZHEVNIKOV, SABININ, SHATS, 2017)

O Telegram não é muito utilizado, se comparado ao WhatsApp, mas possui muitos recursos interessantes designados para desenvolvedores. Um desses recursos é a API - *Application Programming Interface* ou Interface de Programação de Aplicativos, essa que é um conjunto de padrões de programação. Além disso, o Telegram também possui uma API específica para *bots*, onde é possível fazer a criação de um *chatbot* de forma facilitada. (Telegram Bot API, 2018)

Podemos descrever *chatbot* como um programa de computador que tenta simular o comportamento humano ao responder mensagens. Além disso, ele também pode ser programado para automatizar processos repetitivos. Unindo os conceitos descritos anteriormente, podemos observar que é possível desenvolver um *chatbot* para o aplicativo Telegram através dos métodos descritos em sua API. Para que o código do *chatbot* possa ser personalizado por qualquer outro desenvolvedor é necessário fazer a utilização do conceito framework.

O *framework* é um conjunto de códigos abstratos desenvolvidos em uma determinada linguagem de programação. Eles podem ser divididos em módulos, com a possibilidade de fazer a escolha entre utilizar todas as funções disponíveis no código ou somente uma parte delas.

Materiais e Métodos

Como parte dos materiais foi utilizado um *notebook* da marca Samsung, que possui a seguinte configuração: Sistema Operacional Linux (Ubuntu versão 18.04), 8GB de memória RAM e Processador Intel Core i7 com 2.40GHz. O restante dos materiais utilizados está relacionado com os conhecimentos adquiridos através de pesquisas, para que fosse possível o desenvolvimento do *framework*, e será descrito a seguir.

Revisão de literatura

Principais métodos da API

A API do aplicativo Telegram fornece diversos métodos que podem ser utilizados para o desenvolvimento do *bot* e podem ser adaptados para a linguagem de programação escolhida. Os principais métodos utilizados para o desenvolvimento foram: *getUpdates, getMe sendMessage*, *sendPhoto, sendAudio, sendVideo e sendDocument*.

Bots

Um *bot* é um *software* feito com objetivo de automatizar tarefas repetitivas para os seres humanos, e pode possuir desde comandos simples até os mais complexos, dependendo do nível de programação. O tipo de *bot* semelhante ao que está sendo desenvolvido é o *chatbot*.

Comparação entre aplicativos de mensagens instantâneas

O desenvolvimento de um *framework* deve permitir que a aplicação possa ser implementada e personalizada por outro desenvolvedor. Ou seja, o *chatbot* desenvolvido para o aplicativo Telegram deve possibilitar a integração com outras plataformas de mensagens como Slack e Messenger.
A figura 1 mostra os principais métodos utilizados no aplicativo Telegram, e os métodos equivalentes do aplicativo Slack. A cor verde representa os métodos disponíveis e a cor vermelha, os indisponíveis.

MÉTODOS:	Telegram	Slack
Envia mensagens	sendMessage	chat.postMessage
Envia foto(s)	sendPhoto	
Envia áudio(s)	sendAudio	
Envia documento(s)	sendDocument	
Envia vídeo(s)	sendVideo	
Envia arquivos animados (gif)	sendAnimation	
Envia várias fotos e/ou vídeos de uma vez	sendMediaGroup	
Envia a localização do usuário	sendLocation	
Envia um contato do celular	sendContact	
Define uma nova foto de perfil do grupo	setChatPhoto	
Cria um canal/chat de conversa	createChat	channels.create
Remove um usuário da conversa	deleteChatUser	channels.kick
Renomeia o nome/título da conversa	setChatTitle	channels.rename
Adiciona uma descrição no grupo	setChatDescription	
Adiciona/convida um usuário na conversa	addChatUser	channels.invite
Exclui uma determinada mensagem	deleteMessages	chat.delete
Retornar atualizações de mensagens	getUpdates	chat.update
Download de arquivos	getFile	
Webhook	setWebhook	Extensão
Teclado personalizado com opções de resposta	ReplyKeyboardMarkup	
Teclado personalizado com opções de resposta	InlineKeyboardButton	
Retorna informações do usuário	getUsers	users.info
Método para testar o token, retorna informações básicas	getMe	auth.test

Figura 1 – Comparação entre os métodos do Telegram e Slack

Métodos

Os métodos utilizados para o desenvolvimento do *framework* para o aplicativo Telegram serão descritos a seguir.

Criação de um *bot*

Para que seja possível utilizar os métodos disponibilizados na API do Telegram, antes é necessário fazer a criação de um bot. Os passos da criação estão descritos na figura 2.



Figura 2 – Criação de um *bot* no aplicativo Telegram

Escolha da linguagem de programação

Para esta etapa, foi necessário fazer a escolha da linguagem de programação para o desenvolvimento do *framework.* A escolha da linguagem Python aconteceu por diversos motivos, como facilidade de aprendizado e versatilidade.

Uso do módulo telepot

Para o desenvolvimento, foi utilizado o *framework* Telepot que permite a utilização dos métodos descritos na API do Telegram de forma facilitada, e também possui vários exemplos de aplicações. É necessário fazer o *download* do módulo Telepot para que seja possível importá-lo no código, e assim fazer o uso de seus recursos. O Telepot funciona para as versões 2.7 e 3 do Python, sendo que para as versões mais novas é necessário fazer o uso de uma outra versão do módulo.

Resultados e Discussão

Com a ajuda de diversos módulos, foi desenvolvido um *framework* que fica em execução, por um período de tempo definido, aguardando o usuário digitar novas mensagens. Quando uma nova mensagem é recebida, o comando '*contente_type*' descobre o tipo de mensagem enviada (texto, áudio, imagem, etc). Se a mensagem não for do tipo texto, uma mensagem é enviada para o usuário informando que ele precisa digitar uma mensagem de texto. E se a mensagem for do tipo texto, ele vai para as próximas verificações.
As próximas verificações são responsáveis por enviar o que o usuário está pedindo, ou seja, se o usuário digitar 'foto' então o método *sendPhoto* deverá ser utilizado

para enviar uma foto. Foram implementadas funções como enviar áudio, documentos, mensagens, fotos e criação de um teclado, esse que pode ser customizado e programado para realizar diferentes ações.
Durante o desenvolvimento foi possível perceber que um *bot* pode ser utilizado para realizar diferentes ações, e pode ser programado para realizar tarefas específicas. Por causa disso, o *framework* desenvolvido permite que outros desenvolvedores de projetos diferentes consigam dar continuidade ao projeto e adequá-lo a sua realidade.

Conclusões

O conceito de *framework* foi muito utilizado, pois através dele é possível disponibilizar uma pesquisa para que a mesma seja utilizada como base para o desenvolvimento de projetos futuros.
Um dos problemas encontrados durante o desenvolvimento ocorreu com a utilização do módulo telepot, pois o mesmo possui algumas restrições. A versão utilizada (2.7) não permite a implementação de módulos, e com isso o *framework* desenvolvido não possui a possibilidade de escolha entre utilizar todas as funções disponíveis no código ou somente uma parte delas.
A execução do *framework* foi feita no terminal do Linux (Ubuntu versão 18.04) e comprova a funcionalidade de todos os métodos implementados.

Agradecimentos

Agradecimentos à UEPG pela bolsa concedida e ao orientador Dierone César Foltran Júnior por toda a dedicação quanto ao projeto.

Referências

Bot Code Examples. Disponível em: <https://core.telegram.org/bots/samples>. Acesso em: 17 mar. 2018.
Enabling Interactions with Bots. Disponível em <https://api.slack.com/bot-users>. Acesso em: 20 jul. 2018.
Telegram Bot API. Disponível em: <https://core.telegram.org/bots/api>. Acesso em: 17 mar. 2018.
Introduction Telepot. Disponível em: <https://telepot.readthedocs.io/en/latest/>. Acesso em: 08 jun. 2018.
Bots: An introduction for developers. Disponível em: <https://core.telegram.org/bots#botfather>. Acesso em: 08 jun. 2018.
KOZHEVNIKOV, Vadim Andreevich; SABININ, Oleg Yurievich; SHATS, Julia Efimovna. **Library Development for creating bots on Slack, Telegram and Facebook Messenger.** Theoretical & Applied Science, n. 6, p. 59-62, 2017.