UniFG App

Beatriz L. Martins, Erik N. O. Batista, Júlio C. de Barros, Marcos F. V. de Souza

[1]Centro Universitário dos Guararapes (UNIFG)

Recife – PE – Brazil

blmartins017@gmail.com, eriknathan.contato@gmail.com,
juliocbarros339@gmail.com, marcosfvs92@gmail.com

Abstract. *This article is a description of an application aimed at UniFG - Faculdade dos Guararapes, presenting the development of a mobile tool for college students who will be able to access various services offered by the institution directly from their cell phone. The prototype developed used Kivy and KivyMD technologies explored in the Python language.*

Resumo. *Este artigo constitui uma descrição de um aplicativo voltado para a UniFG - Faculdade dos Guararapes, apresentando o desenvolvimento de uma ferramenta mobile para estudantes da faculdade que poderão ter acesso a vários serviços ofertados pela instituição direto do seu celular. O Protótipo desenvolvido usou as tecnologias Kivy e KivyMD exploradas na linguagem Python.*

1. Introdução

Um aplicativo móvel ou mobile é um software desenvolvido para ser utilizado em smartphones e tablets. Sua aquisição é através de uma loja de app, como por exemplo: Google Play para Android e App Store para Apple.

A evolução dos sistemas de telefone móvel vem trazendo mudanças no comportamento dos consumidores ao possibilitar a liberdade de se comunicar, acessar serviços entre outros a qualquer momento.

O avanço da tecnologia nos dispositivos celulares, a partir da necessidade de aprimoramento e melhoria nos componentes de hardware tem permitido um melhor desempenho e surgimento de dispositivos com capacidade similares de um microcomputador. É neste cenário que surgem os smartphones.

Esses telefones geralmente englobam a maioria das funções e serviços encontrados em um computador, como, e-mail, sincronização de dados e a capacidade de armazenamento.

Nas instituições educacionais, nota-se que os aplicativos surgem primeiramente como ferramentas, não tão eficientes, visto a limitação da tecnologia móvel. Entretanto, no começo havia ferramentas simples que já ajudavam os seus estudantes, hoje existem mecanismos mais avançados que permitem, por exemplo, que o aluno saiba como entrar em contato com o professor direto pelo smartphone.

Portanto, tendo em vistas estes fatos, o presente trabalho tem como objetivo analisar fatos sobre a faculdade e implementação de um aplicativo para soluções de dúvidas recorrentes sobre a Instituição UniFg.

2. Parte teórica

O nosso aplicativo foi criado usando a linguagem de programação Python e Kivy, linguagens usadas para criação de aplicações que podem ser usadas em back-end e front-end. Usamos todas as funções de código limpo e de fácil leitura, deixando assim o código legível para quem quiser e puder contribuir futuramente incrementando novas funcionalidades.

Python é uma linguagem de programação criada na década de 80 por Guido Van Rossum. A primeira ideia de implementar o Python surgiu em Amsterdã, Holanda, enquanto Guido participava do projeto no CWI (Centrum Wiskunde & Informatica, centro de Matemática e ciências da computação em um time no qual iria desenvolver a linguagem.

Segundo Guido, no ano de (2014), a nova linguagem deveria preencher o vazio entre o C e o shell Script. Esse foi o objetivo principal na criação do Python.

Python é uma linguagem de alto nível, de script, tipagem dinâmica e orientada a objetos. Pode ser aprendida facilmente tanto por programadores experientes em outras linguagens quanto por programadores iniciantes por sua sintaxe limpa.

A figura abaixo demonstra o uso da linguagem:

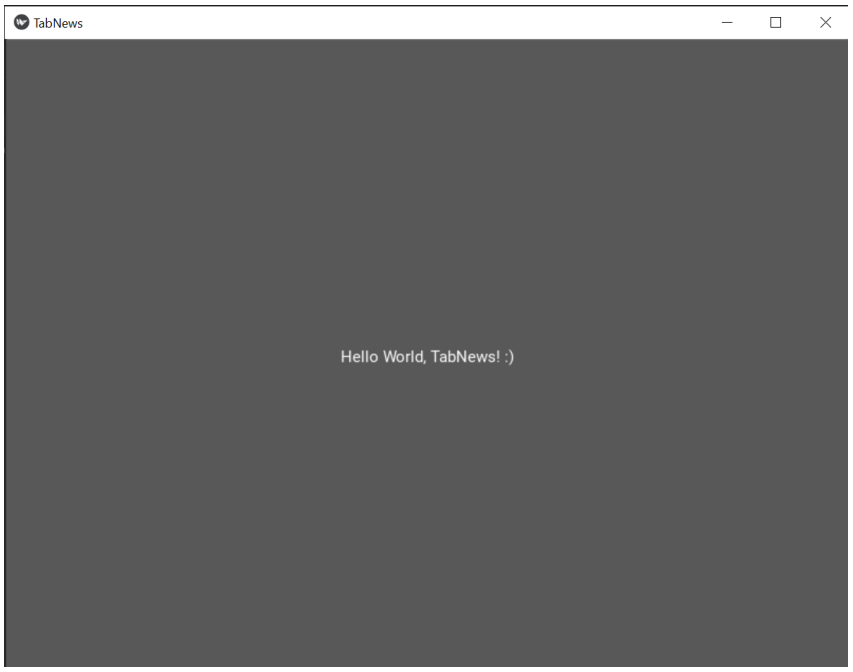
```
1  #coding: utf-8
2
3  valor = input("Digite um número: ")
4  num = int(valor) #convertendo de str p/ int
5
6  print("O número digitado foi: ", str(num))
7
```

A linguagem está presente em grandes empresas. Segundo Cuong(2007), arquiteto de *software* do Youtube, “Python sempre foi uma importante parte do Google e continua até hoje nos sistemas que o envolvem.

Kivy

Kivy é uma biblioteca de código aberto em Python que oferece ferramentas para um rápido desenvolvimento de aplicações que fazem uso de interfaces de usuário. As aplicações desenvolvidas com essa ferramenta podem ser executadas nas plataformas Linux, Windows, OS X, Android e IOS. o motor gráfico é construído sobre OpenGL ES 2 para se obter um rápido processamento

visual, possui mais de 20 Widgets (componentes de interface) para o uso no desenvolvimento e aceita até 5 toques e cliques simultâneos na tela, oferecendo assim diversas formas de como jogos e instrumentos musicais(KIVY, 2016.)



2.1 Subseções

Os títulos da subseção devem estar no negrito, 12pt, alinhado à esquerda.

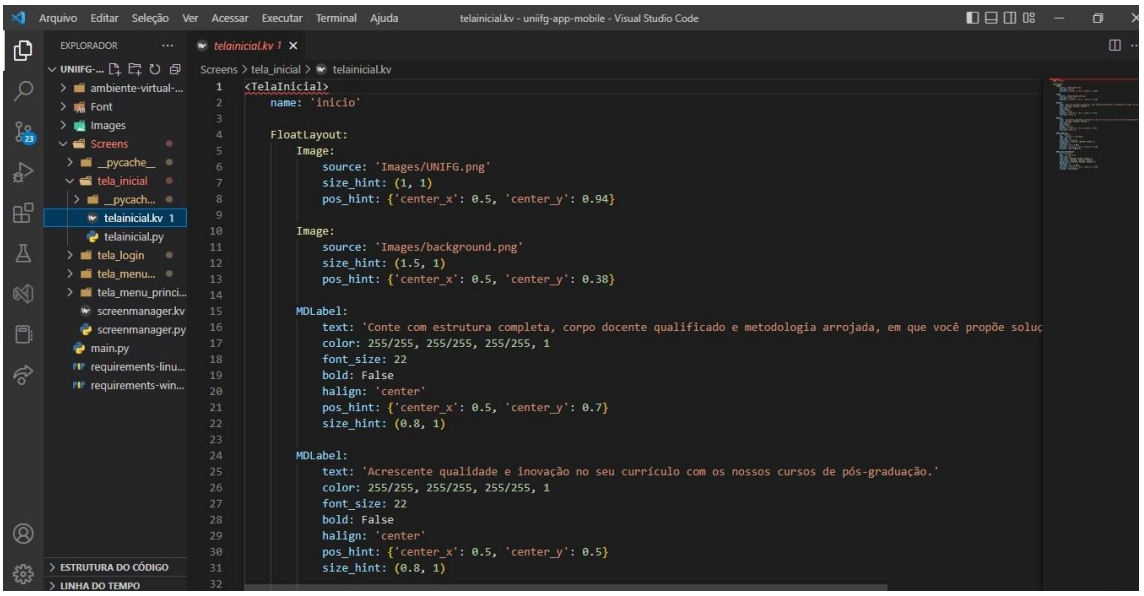
3. Metodologia prática

O projeto foi separado por pastas onde cada pasta foi designada uma função do projeto, deixando assim o projeto mais organizado e legível. O grande desafio de usar a metodologia foi fazer com que cada função estivesse ligada a tela sem impedir que as outras funções fossem executadas. Deixando assim o projeto ao máximo acessível, diferente do chatbot da Unifg que não tem muita funcionalidade/acessibilidade ao usuário final, que termina complicando uma simples tarefa em uma aplicação com erros e pouco aproveitamento. Observa-se que cada etapa do projeto é de extrema importância para o desenvolvimento do mesmo. Segundo Silva e Menezes(2005, p. 30) “A revisão da literatura é fundamental porque fornecerá elementos para evitar a duplicação de pesquisa sobre o mesmo enfoque do tema, e favorece a definição de contornos mais precisos do problema a ser estudado.

3.1 Código do Kivy | Python

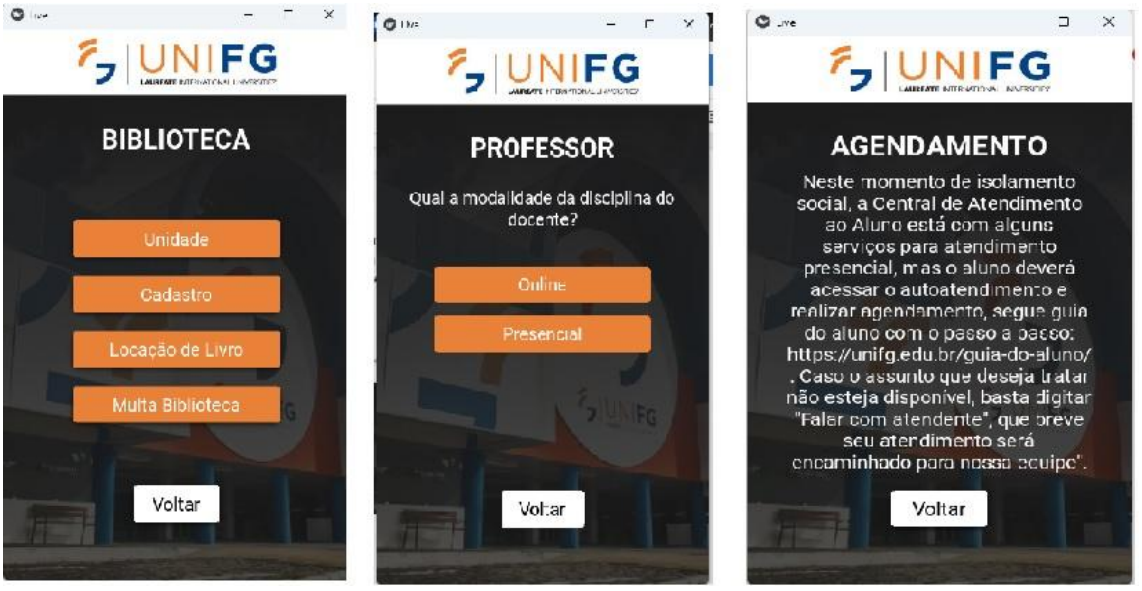
```python
import os
from kivymd.app import MDApp
from kaki.app import App
from kivy.factory import Factory
from kivy.core.window import Window
Window.size = (380,700)

# main app class for kaki app with kivymd modules
class LiveApp(MDApp, App):

    DEBUG = 1 # set this to 0 make live app not working

    # *.kv files to watch
    KV_FILES = {
        os.path.join(os.getcwd(), "Screens/screenmanager.kv"),
        os.path.join(os.getcwd(), "Screens/tela_inicial/telainicial.kv"),
        os.path.join(os.getcwd(), "Screens/tela_login/telalogin.kv"),
        os.path.join(os.getcwd(), "Screens/tela_menu_principal/tela_menu_principal.kv"),

        os.path.join(os.getcwd(), "Screens/tela_menu_atendimento/tela_menu_atendimento.kv"),
        os.path.join(os.getcwd(), "Screens/tela_menu_atendimento/at_caa/at_caa.kv"),

        os.path.join(os.getcwd(), "Screens/tela_menu_atendimento/at_cac/at_cac.kv"),
        os.path.join(os.getcwd(), "Screens/tela_menu_atendimento/at_cac/at_cac_unidade/at_cac_unidade.kv"),
        os.path.join(os.getcwd(), "Screens/tela_menu_atendimento/at_cac/at_cac_contato/at_cac_contato.kv"),
        os.path.join(os.getcwd(), "Screens/tela_menu_atendimento/at_cac/at_cac_ingresso/at_cac_ingresso.kv"),

        os.path.join(os.getcwd(), "Screens/tela_menu_atendimento/at_agendamento/at_agendamento.kv"),

        os.path.join(os.getcwd(), "Screens/tela_menu_atendimento/at_professor/at_professor.kv"),
        os.path.join(os.getcwd(), "Screens/tela_menu_atendimento/at_professor/online/online.kv"),
        os.path.join(os.getcwd(), "Screens/tela_menu_atendimento/at_professor/presencial/presencial.kv"),
```

Parte do arquivo em ***Python*** onde estão as principais funcionalidades, ligando de uma tela a outra do aplicativo, esta imagem está mostrando apenas uma parte do código inteiro do código Main.py



Aqui mostra parte do arquivo em Kivy onde foram definidas as estruturas de posições de cada elemento na tela inicial do aplicativo. Trabalhamos em uma estrutura padrão para não fugir da experiência do usuário, fazendo com que ele consiga facilmente acessar todas as telas sem que haja nenhuma confusão ou falta de entendimento no acesso da do app.

4. Imagens



5. Resultados

Obtemos resultados de facilidade na comunicação aluno professor, faculdade e aluno. Ajudando assim a otimizar o tempo de serviço dos operadores da faculdade, agilizando processos que antes estavam cada vez mais pesado, agora contando com essa nova aplicação, cada processo poderá levar menos tempo do que o que se espera no sistema anterior que variava entre 40 min/1 hora, se não fosse em tempo de matrícula e rematrícula dos alunos no início do período. Todo esse projeto foi idealizado para ajudar na hora de manutenção de qualquer serviço que o aluno ou professor precisasse realizar.

6. Conclusões

Temos conclusões que de acordo com os testes feitos na plataforma, esta implementação além de ter um custo benefício baixo, ajudaria toda parte operacional da faculdade relacionada a operações com comunicação. Observamos que a Unifg tem muito a crescer, o que eles podem fazer é se dedicar um pouco mais ao conforto tanto dos alunos que já estão na faculdade quanto nos novos alunos.

7. Referências

CHAPADEV, My First Experience With Kivy. Disponível no site: http://cheparev.com/myexperience-with-kivy/. Acesso em: 20 out, 2016. CUONG,

Do, Quotes About Python, 2007. EMPORIO, As 10 melhores aplicações multiplataforma. Disponível no site:

http://blog.emporio-web.com/lang/pt-pt/2010/05/as-10-melhores-aplicacoes-multiplataforma/. Acesso em: 19 jun, 2016.

DESENVOLVIMENTO DE APLICAÇÕES MULTIPLATAFORMA COM KIVY no site:

https://m.uniara.com.br/arquivos/file/cca/artigos/2016/ricardo-augusto-silva-mattos.pdf

.