

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO NORTE – IFRN**

DIRETORIA DE GESTÃO E TECNOLOGIA DA INFORMAÇÃO

CURSO SUPERIOR DE TECNOLOGIA EM REDES DE COMPUTADORES

IMPLANTAÇÃO SERVIÇOS DE VOIP

JANSSEN CRISTIAN MARTINS

VoIP: O Uso Desta Ferramenta para Consulta da Taxa de Câmbio

NATAL/RN

2018

JANSSEN CRISTIAN MARTINS

VoIP O Uso Desta Ferramenta para Consulta da Taxa de Câmbio

Projeto apresentado como requisito final para obtenção de nota na segunda etapa da disciplina de Implantação de Serviços de VoIP do curso superior de Tecnologia em Redes de Computadores do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – IFRN.

Orientador: Me. Teobaldo Adelino Dantas de Medeiros

NATAL/RN

2018

VoIP: O Uso Desta Ferramenta para Consulta da Taxa de Câmbio

Aluno: Janssen Cristian Martins

Orientador: Me. Teobaldo Adelino Dantas de Medeiros

Resumo: A crescente expansão da tecnologia que transmite voz através da rede abre um leque de oportunidades no que diz respeito ao desenvolvimento de novas aplicações para auxiliar o mercado. Este projeto propõe implementar um serviço de discagem que consulta, em tempo real, o valor das principais moedas mundiais em comparação ao real brasileiro e informa ao usuário. Além de dispor de uma opção de armazenamento para que o cliente seja capaz de comparar tais valores futuramente.

Palavras-chave: VoIP. Tecnologia. Mercado. Moedas.

Sobre o Projeto

O projeto trata-se de um ramal que ao ligá-lo, exibirá duas opções ao usuário. A primeira, terá a função de informar os valores das principais moedas mundiais em comparação ao real brasileiro e cadastrar tais valores, caso seja requerido. A segunda poderá consultar o banco de dados onde foi cadastrado essas consultas e informar ao cliente tudo que está armazenado, sendo possível assim, ser feita a comparação de acordo com as datas, dos valores dessas moedas.

Funcionamento

Ao ligar para um determinado número a partir de uma central VoIP, o usuário será redirecionado para um script denominado projeto_agi.py escrito em Python e integrado com AGI que apresentará duas opções.

Ao escolher a opção 1, o script executará um outro arquivo .py (Figura 1) que terá a função de fazer uma requisição web via requests.get a API disponível neste link: <<https://api.promasters.net.br/cotacao/>>, tratar essa requisição como um dicionário JSON e armazenar este valor em um arquivo de texto. Após isto, o script principal irá ler o arquivo e

armazenará o valor recebido em uma variável, logo após, excluirá o mesmo arquivo de texto, anteriormente criado (Figura 2).

```
#!/usr/bin/env python
import requests
import json

def requisicao():
    req = requests.get('http://api.promasters.net.br/cotacao/v1/valores')
    dic_cotacao = json.loads(req.text)
    valor_btc = str(dic_cotacao['valores']['BTC']['valor'])
    arq = open('btc.txt', 'w')
    arq.write(valor_btc)
    arq.close()

requisicao()
```

Figura 1 – Arquivo .py que faz a requisição web do BitCoin e escreve em um arquivo de texto.

```
elif op == '3':
    os.system('python3 /var/lib/asterisk/agi-bin/requisicao_btc.py')
    sleep(3)
    arq = open('btc.txt', 'r')
    for linha in arq:
        valor_btc = linha
    btc = float(valor_btc)
    os.system('rm -f btc.txt')
```

Figura 2 – Execução do script secundário que faz a requisição web do valor da moeda, armazenamento deste valor em uma variável e exclusão do arquivo de texto.

Após todas essas tarefas terem sido executadas com êxito, a AGI informará a data da requisição, o valor da moeda a qual foi requerida e perguntará se o usuário deseja armazenar este valor no banco de dados (Figura 3).

```
data = datetime.datetime.now()
dia = data.day
mes = data.month
ano = data.year
agi.stream_file('Hoje_e')
agi.say_number(dia)
agi.stream_file('Do')
agi.say_number(mes)
agi.stream_file('De')
agi.say_number(ano)
agi.stream_file('Valor_btc')
agi.say_number(btc)
while True:
    agi.stream_file('Gostaria_cadastrar')
    cad = agi.wait_for_digit('20000')
```

Figura 3 – AGI fala a data da requisição, bem como valor da moeda e pergunta se o usuário deseja cadastrar esta requisição

Se for informado que sim, será criado um banco de dados SQLite3, caso ainda não exista, para aquela determinada moeda, com as colunas (dia, mês, ano e valor da moeda) e os dados serão cadastrados (Figura 4).

```
banco = "banco_valores_btc.db"
conexao = sqlite3.connect(banco)
c = conexao.cursor()
c.execute('CREATE TABLE IF NOT EXISTS btc (dia integer, mes integer, ano integer, valor real)')
c.execute("insert into btc (dia, mes, ano, valor) values (?, ?, ?, ?)", (dia, mes, ano, btc))
agi.stream_file('Dados_cadastrados')
conexao.commit()
c.close()
```

Figura 4 – Criação do banco e inserção dos dados

Ao escolher a opção 2, o usuário será perguntado sobre qual moeda gostaria de consultar em seu banco de dados. Assim que informado, o script fará uma busca e a AGI informará todas as requisições armazenadas no banco, bem como as datas em que foram adicionadas (Figura 5).

```
banco = "banco_valores_btc.db"
conexao = sqlite3.connect(banco)
c = conexao.cursor()
c.execute('CREATE TABLE IF NOT EXISTS btc (dia integer, mes integer, ano integer, valor real)')
c.execute("select dia, mes, ano, valor from btc")
resultado = c.fetchall()
for linha in resultado:
    dia = linha[0]
    mes = linha[1]
    ano = linha[2]
    valor = linha[3]
    btc = int(valor)
    agi.say_number(dia)
    agi.stream_file('Do')
    agi.say_number(mes)
    agi.stream_file('De')
    agi.say_number(ano)
    agi.stream_file('O_btc_valia')
    agi.say_number(btc)
conexao.commit()
c.close()
```

Figura 5 – Consulta ao banco de dados e reprodução dos dados na AGI

Tecnologias Utilizadas

No decorrer do desenvolvimento do projeto, foram utilizadas diversas bibliotecas e tecnologias que foram estudadas no curso até aqui. As mesmas serão listadas a seguir:

- Linguagem de Programação Python versões 2 e 3 (Utilizada para desenvolver os scripts);
- Asterisk (PBX responsável pela telefonia VoIP);
- AGI (Linguagem que integra o código python ao PBX Asterisk, ferramenta responsável por manipular tudo que diz respeito ao VoIP);

- Web Service (API utilizada para consultar os dados das moedas via web);
- Banco de Dados (Onde foram armazenados os dados e requisições das moedas);
- Biblioteca Requests (Desempenhou a função de fazer as requisições GET a API);
- Biblioteca JSON (Tratou o recebimento dos dados da API como um dicionário);
- Biblioteca Os (Possibilitou a execução de comandos Shell Bash no script Python);
- Biblioteca Datetime (Utilizada para requisitar a hora atual que foi armazenada no banco);
- Biblioteca Time (Serviu para dar um intervalo de 3 segundos após a requisição web);