

BÓSON TREINAMENTOS EM CIÊNCIA E TECNOLOGIA

Aprenda Ciências, Bancos de Dados, Linux, SQL, Excel, Lógica de Programação, Hardware, Eletrônica, Química, Arduino e muito mais!

[HOME](#) | [BANCOS DE DADOS](#) | [ELETRÔNICA](#) | [LINUX](#) | [PROGRAMAÇÃO](#) | [HARDWARE, REDES](#) | [TECNOL. EMERGENTES](#) | [MAKERSPACE](#) | [CARREIRA](#)

Início » Programação em Python » Algoritmo para validação de RG SSP/SP (dígitos verificadores) em Python

Algoritmo para validação de RG SSP/SP (dígitos verificadores) em Python

Postado Em 30/06/2023 [data do post] por Fábio dos Reis [autor do post] em Programação em Python [categoria do post]

Curtir 0

Tweetar

Validação de RG SSP/SP (dígitos verificadores) em Python

Neste tutorial mostro como realizar o cálculo do dígito verificador para validação de RG emitido no estado de SP, usando um script criado em **Python** – e que você pode adaptar para a linguagem de programação que desejar.

Formato do RG

O formato de um RG (Registro Geral) de São Paulo é composto por 9 dígitos no formato **XX.XXX.XXX-Y**, onde os “X” representam os dígitos numéricos e o “Y” representa o dígito verificador (DV).

Esse dígito verificador é o módulo 11 dos 8 dígitos anteriores, calculado usando um algoritmo específico – que vamos criar em Python.

No caso do RG, o DV corresponde ao resto da divisão por 11 do somatório da multiplicação de cada um dos outros oito algarismos respectivamente por 9, 8, 7, 6, 5, 4, 3, 2, a partir do dígito mais à esquerda. O resto 10 é considerado como valor X.

Observação importante: Em outros estados o formato do RG pode ser levemente diferente, e o algoritmo que mostrarei aqui pode não funcionar corretamente para realizar a validação, sendo necessário criar um algoritmo diferente.

Como calcular o dígito verificador do RG: Exemplo

Vejamos como realizar o cálculo do dígito verificador do RG por meio de um exemplo.

Pesquisar no Site

Para pesquisar, digite o que deseja e tecla Enter

Todos os posts da Bóson Treinamentos em ordem cronológica

Curso de MySQL com Certificado e Material para Download



Posts mais recentes

Sensor de Temperatura TMP36 com Arduino e visor LCD 16x2

Como usar o objeto Series da biblioteca Pandas em Python

Como gerar números primos em Python em um intervalo especificado

O que são f-strings em Python

Algoritmo Torre de Hanoi em Python

O que é Recursividade

Algoritmo para verificar se um número pertence à série de Fibonacci

Algoritmo para verificar se número é quadrado perfeito

Algoritmo para Fibonacci Recursivo e Iterativo – Lógica de Programação

Suponhamos o RG de número 29.465.327-2. Como verificar se ele é válido?

Para isso temos de efetuar a validação do dígito verificador – que no caso é o valor 2. Vamos ao processo:

Passo 1

Multiplicar os oito primeiros dígitos do RG pela sequência numérica de 9 a 2 e realizar sua soma:

2	9	4	6	5	3	2	7	x
9	8	7	6	5	4	3	2	=
18	72	28	36	25	12	6	14	

Somatório dos valores obtidos:

$$18 + 72 + 28 + 36 + 25 + 12 + 6 + 14 = 211$$

Passo 2

Calcular o módulo 11 do somatório – ou seja, o resto da divisão por 11 do somatório obtido no passo 1.

O valor resultante desse cálculo é o dígito verificador do RG:

$$211 \% 11 = 2$$

Dígito verificador calculado: 2 → RG validado

Obs.: O símbolo % representa a **operação de módulo**.

Note que existe um caso especial a ser tratado aqui:

1. Caso o resto calculado seja igual a 10, o dígito verificador será considerado como a letra **X** (que representa 10 em algarismos romanos).

Programa em Python para validação de RG SSP/SP

```
# Função de validação do dígito verificador de RG SSP/SP em Python
def validaRG(rg):
    cont = 9
    soma = 0
    # Somatório das multiplicações dos 8 primeiros dígitos do rg
```

Algoritmo para validação de RG SSP/SP (dígitos verificadores) em Python
 Algoritmo para validação de CPF (dígitos verificadores) – Lógica de Programação
 Como criar menus com o comando SELECT em Shell Script
 Questionário de Lógica de Programação com respostas
 Formatar e exibir texto com o comando printf no terminal e em scripts
 Por quê Python pode não ser uma boa linguagem para iniciantes?
 Como acessar bancos de dados SQL Server usando Python
 Ler entrada de dados com comando read em Shell Scripting
 Exemplos do comando curl no Linux – Tutorial completo
 Controle Condicional em Rust – if / else if / else
 Declaração e Atribuição de Variáveis em Rust
 Como instalar o Miniconda no Linux (Python)
 23 – Lógica de Programação – Pesquisa Binária em Vetores (Arrays)
 O que são Funções Lambda em Python ou Funções Anônimas
 O que é Compreensão de Lista em Python
 Tipos de Dados em Rust
 Como usar a função map em Python
 Como usar a função zip em Python
 Como criar seu primeiro programa em Rust e instalação da linguagem
 O que é a linguagem Rust de programação
 Como implementar uma pilha em linguagem C usando array

Tags

Análise de Dados Apple Arduino

Bancos de Dados c#

Carreira Certificação

Ciência de Dados Criptografia

Elettricidade Eletrônica

Eletrônica Digital Física

Hardware Java JavaScript

Linguagem C Linguagem R Linux

Livros LPIC Lógica

Lógica de Programação Mac OS X

Maker Matemática

```
for digito in rg[0:8]:
    soma += int(digito) * cont
    cont -= 1

# Calcula dv fazendo somatório mód 11
dv = soma % 11

# Verifica se dv calculado coincide com dv fornecido no RG (último
dígito)
if ((dv == 10) & (rg[-1] == 'X')):
    return(True)
elif (int(rg[-1]) == dv):
    return(True)
else:
    return(False)

if __name__ == '__main__':
    rg = input("Digite seu RG, sem pontos ou traço: ")
    res = validaRG(rg)
    if (res == True):
        print("RG validado com sucesso.")
    else:
        print("Documento inválido.")
```

Modelagem de Dados

MySQL

Notícias

Oracle

PHP

PostgreSQL

Powershell

Programação

Protocolos

Python

Redes

Segurança

Shell Scripting

SQL

SQL Server

Ubuntu

Virtualização

Windows

Windows 10

Vídeos sobre Carreira e Certificação

É possível aprender Tecno...



Funcionamento do algoritmo

O algoritmo começa definindo uma função de nome **validaRG(rg)** e criando dentro dela as variáveis **cont** e **soma**. A variável **cont** é inicializada com o valor 9 e a variável **soma** é inicializada com o valor 0.

Em seguida, é iniciado um laço *for* que percorre os primeiros 8 dígitos do número de RG fornecido. Para cada dígito, o algoritmo multiplica o valor do dígito pela variável **cont** (que é decrementada em cada iteração) e acumula o resultado na variável **soma**.

Após o término do laço, o algoritmo calcula o dígito verificador (**dv**) fazendo o módulo 11 da soma.

Em seguida, o algoritmo verifica se o dígito verificador calculado coincide com o dígito verificador fornecido no RG (último dígito).

Se o dígito verificador calculado for 10 e o último dígito do RG for 'X', isso é considerado uma coincidência e retorna *True*. Caso contrário, se o último dígito do RG for um número e for igual ao dígito verificador calculado, também retorna *True*.

Caso contrário, retorna *False*.

No corpo principal do programa **if __name__ == '__main__':**, o algoritmo solicita que o usuário digite o número do RG, sem pontos ou traços. Em seguida, chama a função definida **validaRG()**

Aprenda Redes de Computadores com o livro de Gabriel Torres:



passando o número do RG como argumento e armazena o resultado na variável res.

Por fim, o algoritmo imprime uma mensagem indicando se o RG foi validado com sucesso ou se é inválido, com base no resultado retornado pela função validaRG.

Teste

Teste o algoritmo com seu número de RG, e com outros números, como **45.727.503-0** (contém DV zero no final) e **37.606.335-X** (dígito verificador é a letra X), para verificar também os casos especiais.

Tente também melhorar o programa para que aceite a digitação dos documentos com pontos e traços, como exercício.

É isso aí!

Veja também: [Como calcular e validar dígitos verificadores em números de CPF](#)

Algoritmos Programação Python

Curtir 0

Tweetar



Sobre Fábio dos Reis (1175 Artigos)

Fábio dos Reis trabalha com tecnologias variadas há mais de 30 anos, tendo atuado nos campos de Eletrônica, Telecomunicações, Programação de Computadores e Redes de Dados. É um entusiasta de Ciência e Tecnologia em geral, adora Viagens e Música, e estuda idiomas, além de ministrar cursos e palestras sobre diversas tecnologias em São Paulo e outras cidades do Brasil.

Contato: | Website

← Artigo anterior

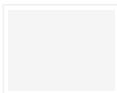
Próximo artigo →

Artigos Relacionados



Como gravar dados em arquivos de texto com Python

O que é Compreensão de Lista em Python



Programação em C# – Estruturas de Decisão Condicional Aninhadas – 09



Introdução à Linguagem de Programação C++



Algoritmo Torre de Hanoi em Python

Escreva um comentário

Seu e-mail não será divulgado

Comentário

Nome*

Email*

Website

Publicar comentário

Siga-nos nas Redes Sociais!



Arquivos

Arquivos

Artigos Recentes

Sensor de Temperatura TMP36 com Arduino e visor LCD 16x2

Como usar o objeto Series da biblioteca Pandas em Python

Como gerar números primos em Python em um intervalo especificado

O que são f-strings em Python

Algoritmo Torre de Hanoi em Python

O que é Recursividade

Buscar no Site