

Documentação 6: Rastreamento de dados pelo CPF

Thiago Fernandes Canonici

1. JUSTIFICATIVA:

Esta entrega foi um aprimoramento da entrega 4 (parcelas atrasadas e dias de atraso) e correção da entrega 5 (validação de CPF's). Primeiro verificou-se a veracidade dos CPF's para a decisão de alocação de recursos para a previsão de comportamento das pessoas físicas. CPF's falsos foram descartados, pois se mostraram ineficazes, fugindo de uma entrega de valor.

2. OBJETIVO:

Objetivou-se aplicar um sistema de filtragem de CPF's válidos, rastreamento das faturas, parcelas atrasadas, dias atrasados e relatório de erros caso houver, usando os mesmos CPF's. Esses dados são importantes para a previsão de comportamento das pessoas físicas. O levantamento de dados de CPF's inválidos é ineficaz.

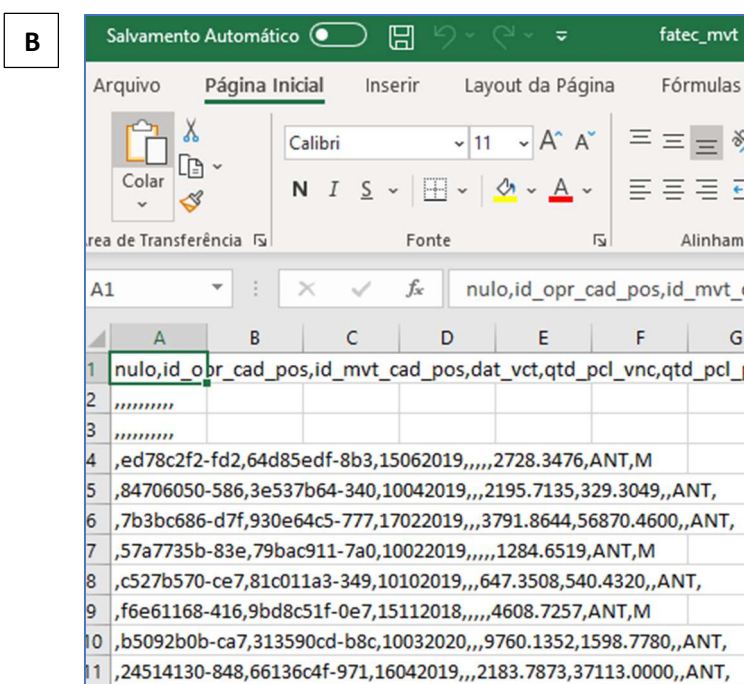
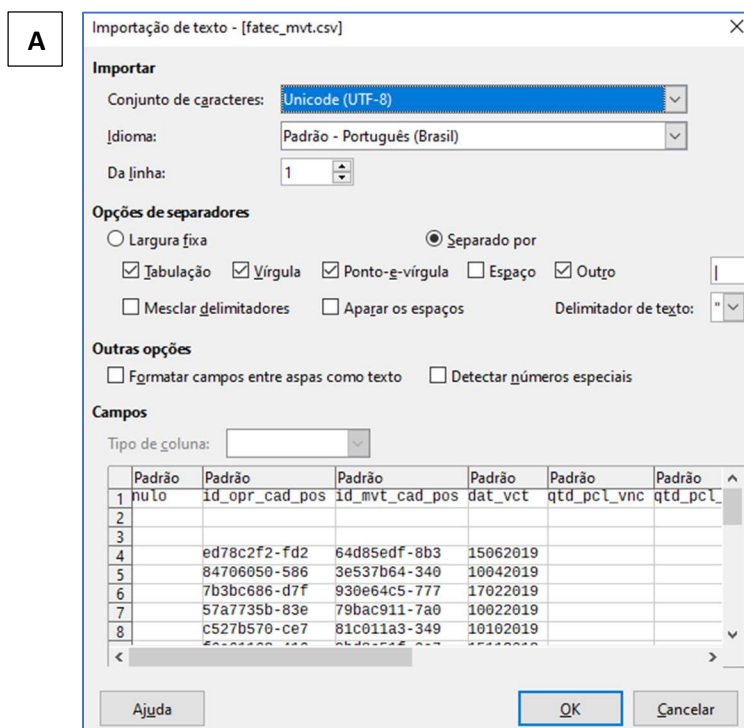
3. MATERIAL E MÉTODO:

As ferramentas utilizadas foram o idle do python para a programação usando essa linguagem, o *jupyter notebook*, o Libre Office Calc, o Excel, o bloco de notas extensão “txt” e as tabelas fornecidas pelo SPC.

3.1.Libre Office Calc e Excel:

Esta ferramenta foi utilizada para o preparo dos dados para sua manipulação no *jupyter notebook*. Como as tabelas estavam em extensão “.csv” o “Calc” foi o melhor software para manipular diretamente as colunas das tabelas para sua formatação e preparo para que sua análise fosse possível (Fig.1).

Figura 1: Página inicial da abertura das tabelas em “Calc”, A, e Excel, B. Nota-se em A que as colunas estão separadas (guia “Campos”); Em B as colunas estão todas juntas como mostrado na área de trabalho.



3.2. BLOCO DE NOTAS:

O bloco de notas foi usado para salvar as colunas de datas das tabelas para a aplicação de códigos ¹ criados de formatação de datas e adição de zero a valores nulos. O código “nulo” foi adaptado para inserir zero no início das datas com 7 algarismos para, em seguida, rodar o código “data”. Essa ordem é importante para a separação de dia, mês e ano corretamente, pois a posição dois da data “9122019” é o número 2, resultando em “91/22...”; em oposição a “09122019”, resultando em “09/12...”. Substituiu-se os valores resultantes nas colunas das tabelas do SPC (foram feitos backups das tabelas originais caso houvesse algum arquivo corrompido ou erro humano).

Os arquivos foram salvos e abertos no Excel para finalizar a formatação por algumas incompatibilidades entre o “Calc” e o Excel, para serem manipulados no *jupyter notebook*.

3.3. JUPYTER NOTEBOOK:

O *jupyter notebook* é uma ferramenta de análise de dados, em que sua instalação no computador é obrigatória e seu acesso é feito por linha de comando digitando “jupyter notebook <ENTER>”. Uma página da web em seu navegador principal abrirá. Todas as bibliotecas a serem utilizadas devem estar instaladas no computador; a biblioteca “pandas” e “matplotlib.pyplot” foram as requisitadas.

¹ Estes códigos foram documentados na *Sprint 1*. Sua descrição seria prolixa.

4. RESULTADO E DISCUSSÃO:

Nesta entrega final do projeto, os dados passaram por vários testes e filtros, identificando CPF's que passaram na lógica aplicada pelo Ministério da Fazenda.

Serão exibidos nas telas do aplicativo os gráficos, as tabelas de valor e tabelas mostrando os erros de 13 CPF's ou o que estiver documentado nas tabelas do SPC. No caso da tabela de endereço, só há 12 CPF's registrados.

4.1. PREPARO DOS CPF'S PARA A VALIDAÇÃO:

Os CPF's contêm exatos 11 algarismos. Ao inserir um valor que comece com zero no Excel ou Calc, o software deleta esse zero inicial. Pressupõe-se² que os CPF's de dez algarismos começassem por zero.

O código “nulo” foi adaptado para a inserção de zero em CPF's com 10 algarismos. O programa pega cada número, passa por um código condicional avaliando o comprimento e insere o zero na primeira posição caso o comprimento seja igual a dez, retornando ao final todos os CPF's no tratamento inicial (Fig. 2). Nesta etapa não se tem os valores válidos dos CPF's.

Figura 2: Função em python para a inserção de zero na primeira posição nos CPF's com 10 algarismos.

```
In [334]: def cpf(a):  
           a = str(a)  
           a = list(a)  
  
           if len(a) == 10:  
               a.insert(0, '0')  
  
           a = ''.join(a)  
           return a
```

² **Erro possível:** Necessita-se da confirmação do SPC para saber se a pressuposição está correta.

4.2. VALIDAÇÃO DE CPF'S:

Elaborou-se um novo código baseado em uma outra referência bibliográfica que utiliza o raciocínio lógico do Ministério da Fazenda (ALGORITMO PARA VALIDAR CPF, 2015?), pois a da quinta entrega reprovou nos testes. O grupo controle dos testes embora pequeno, cinco CPF's, surtiu efeito na validação do código; ele está no GitHub na pasta "Semana 6/Análise_backEnd" arquivo "validade_cpf.py" para testes futuros.

Os testes foram conduzidos escrevendo o número do CPF em aspas simples ou duplas dentro da função "validade()" os resultados possíveis foram 'válido' ou 'inválido'. Alterou-se a ordem dos números dos CPF's mantendo os 11 algarismos e o resultado foi 'inválido'. O algoritmo passou nos cinco testes.

O uso de aspas é obrigatório para que o interpretador do python entenda o valor digitado como número decimal, ou ele entenderá como número octal caso o CPF comece com zero.

O código consiste na separação do número de CPF do dígito. Ex.: XXX.XXX.XXX-XX; número do CPF: XXX.XXX.XXX, dígito -XX. Faz-se algumas multiplicações com cada algarismo e aplica-se uma somatória, multiplica-se por dez e divide-se por 11. Compara-se o resto da divisão com o primeiro dígito do CPF; pega-se uma nova sequência do CPF, agora incluindo o primeiro dígito, e compara-se o resto da divisão com o último dígito. Se ambos os restos forem iguais aos respectivos dígitos, então o CPF é verdadeiro (ALGORITMO PARA VALIDAR CPF, 2015?). Esta lógica ³ teve de ser adaptada, porque os primeiros testes deram errado, CPF's com todos os números repetidos podem dar válido mesmo não o sendo, exemplo: 111.111.111-11 seria válido, além de outras complicações (Fig. 3).

De 3012 linhas na coluna "cpf" do arquivo "fatec_pessoa_fisica", 13 CPF's foram retornados com a aplicação da função da figura 3 e eliminação dos campos nulos (Tab. 1). Foram esses 13 CPF's (tabela 1) usados para o rastreamento das parcelas, dias do pagamento e faturas. O restante, 2989, foi exibido como erro, e não como indicador.

³ O código está disponível no arquivo validade_cpf para o teste de mesa para o melhor entendimento de seu funcionamento.

Como os testes foram poucos, pegou-se cada CPF resultante e pesquisou-se no website da Receita Federal. A verificação da resposta do website foi feita com um CPF inválido digitado de forma aleatória (Fig. 4A). A resposta foi “CPF incorreto. (...)” (Fig. 4B). Sabendo do tipo de resposta da interface gráfica, pesquisou-se os CPF’s achados na tabela do SPC. A figura 4C exibe o resultado desta pesquisa com a resposta “Data de nascimento (...) está divergente (...)”, sem apontar erro ao CPF.

Figura 3: Código de validação usado no *jupyter notebook* (A a D).

A

```
def valido(g):  
    b = str(g)  
    a = 0  
    d = 0
```

B

```
if len(b) == 11:  
    c = list(b)  
  
    if b[0]:  
        c[0] = str(int(b[0])*10)  
    if b[1]:  
        c[1] = str(int(b[1])*9)  
    if b[2]:  
        c[2] = str(int(b[2])*8)  
    if b[3]:  
        c[3] = str(int(b[3])*7)  
    if b[4]:  
        c[4] = str(int(b[4])*6)  
    if b[5]:  
        c[5] = str(int(b[5])*5)  
    if b[6]:  
        c[6] = str(int(b[6])*4)  
    if b[7]:  
        c[7] = str(int(b[7])*3)  
    if b[8]:  
        c[8] = str(int(b[8])*2)  
  
    cont = 0  
    for j in range(9):  
        cont += int(c[j])
```

```

C  if str((cont*10)%11) == b[-2]:
    d += 1

    if b[0]:
        c[0] = str(int(b[0])*11)
    if b[1]:
        c[1] = str(int(b[1])*10)
    if b[2]:
        c[2] = str(int(b[2])*9)
    if b[3]:
        c[3] = str(int(b[3])*8)
    if b[4]:
        c[4] = str(int(b[4])*7)
    if b[5]:
        c[5] = str(int(b[5])*6)
    if b[6]:
        c[6] = str(int(b[6])*5)
    if b[7]:
        c[7] = str(int(b[7])*4)
    if b[8]:
        c[8] = str(int(b[8])*3)
    if b[9]:
        c[9] = str(int(b[9])*2)

    cont = 0
    for j in range(10):
        cont += int(c[j])

```

```

D  e = str((cont*10)%11)
    if e[-1] == b[-1]:
        d += 1

    for i in range(len(b)-1):
        if b[i] == b[i+1]:
            a += 1

    if d == 2 and a != 10:
        return g

```


Tabela 1: Após a aplicação do algoritmo de validação de CPF's, obtiveram-se 13 linhas diferentes de nulo.

```
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  -
0    nulo    0 non-null      object
1    cpf      13 non-null     object
dtypes: object(2)
memory usage: 312.0+ bytes
```

| | nulo | cpf |
|------|------|-------------|
| 66 | NaN | 64122018315 |
| 258 | NaN | 05227401381 |
| 260 | NaN | 06090237051 |
| 1252 | NaN | 05309680888 |
| 1515 | NaN | 53829829299 |
| 1805 | NaN | 11435403525 |
| 1811 | NaN | 09639103101 |
| 1888 | NaN | 24048813021 |
| 1893 | NaN | 28537130320 |
| 2044 | NaN | 38553034011 |
| 2316 | NaN | 12731112573 |
| 2854 | NaN | 18106900800 |
| 2925 | NaN | 01367130760 |

Figura 4: Teste de verificação da veracidade dos CPF's descobertos (A-C). **A:** CPF digitado aleatoriamente para descobrir qual é a resposta que o website dá. **B:** Resposta do website, note que a resposta é "CPF incorreto". **C:** Resposta do website com um CPF resgatado com o algoritmo de validação de CPF's. Note que a resposta foi que a data de nascimento estava incorreta.

Receita Federal
MINISTÉRIO DA ECONOMIA

Perguntas Frequentes | Contato | Serviços | Dados Abertos | Área de Imprensa | Onde Encontrar

A

■ Comprovante de Situação Cadastral no CPF

Preencha os campos abaixo com os dados solicitados.

CPF: 783.928.389-32

Data de Nascimento: 01/01/1985

Realizar Consulta com Captcha Sonoro

✓ Não sou um robô

reCAPTCHA
Privacidade - Termos

O comprovante gerado não fornece informações sobre a situação econômica, financeira ou fiscal do titular do CPF, limita-se tão somente a comprovar a situação cadastral no CPF.

Consultar Limpar

Receita Federal
MINISTÉRIO DA ECONOMIA

Perguntas Frequentes | Contato | Serviços | Dados Abertos | Área de Imprensa | Onde Encontrar

B

Comprovante de Situação Cadastral no CPF

CPF incorreto. Retorne à [página anterior](#) e informe-o novamente!
Informe o CPF completo sem separadores.
Exemplo: 999999999999

Receita Federal

MINISTÉRIO DA ECONOMIA

Buscar no portal

Perguntas Frequentes

Contato

Serviços

Dados Abertos

Área de Imprensa

Onde Encontro

Avisos

English

Español

C

Comprovante de Situação Cadastral no CPF

Data de nascimento informada 10/01/1987 está divergente da constante na base de dados da Secretaria da Receita Federal do Brasil. Retorne a página anterior e informe-a novamente. Persistindo esta mensagem, dirija-se a uma unidade dos Correios, Banco do Brasil ou Caixa Econômica Federal. Caso esteja no exterior, dirija-se a uma representação diplomática brasileira. Retorne à [página anterior](#) e informe-o novamente!

4.3. CPF'S INVÁLIDOS:

Há 2.989 campos inválidos de CPF. A tabela 2 exibe 10 deles. Note que existem CPF's com 11 algarismos inválidos. Eles foram excluídos pela validação de CPF's e, portanto, considerados erros.

Tabela 2: Dez CPF's inválidos resgatados como erro. O total de erros verificado são 2989 campos da coluna "cpf". Note a ocorrência de CPF's com 11 algarismos que também são considerados inválidos.

| | nulo | cpf |
|----|------|-------------|
| 2 | NaN | 37311167211 |
| 3 | NaN | 34115266395 |
| 4 | NaN | 35622655491 |
| 5 | NaN | 33222324480 |
| 6 | NaN | 1505546115 |
| 7 | NaN | 33934682013 |
| 8 | NaN | 1230119467 |
| 9 | NaN | 42785953989 |
| 10 | NaN | 35515466259 |
| 11 | NaN | 33160354240 |

4.4. RESGATE DOS ENDEREÇOS DOS CPF'S VÁLIDOS:

Utilizaram-se os CPF's válidos como chave primária para o resgate das linhas do arquivo "fatec_endereco_pessoa_fisica". Dos 13 CPF's 12 estão registrados com o 01367130760 ausente. Para a identificação dos estados na coluna "estado de origem do cpf", verificou-se o último algarismo do CPF antes do dígito. Exemplo: XXX.XXX.XX(NUM)-XX (CPF-Dígito), em NUM lê-se o algarismo de 0 a 9 que dirá qual é o estado. Desses 12, um não se deslocou da região de origem do CPF. Note que alguns algarismos de estado do CPF pertencem a mais de um estado (tabela 3).

Tabela 3: Tabela de endereço. As Colunas “id_pessoa_fisica”, “cidade atual” e “estado atual” são dados do SPC; A coluna “estado de origem do cpf” foi filtrada usando os CPF’s válidos. Note que há registro de 12 CPF’s com o 01367130760 faltando e que o rastreamento do estado de origem do CPF pode referir-se a mais de uma localidade.

| id_pessoa_fisica | cidade atual | estado atual | estado de origem do cpf |
|------------------|--------------|--------------|---|
| 687332675 | AQUIRAZ | CEARA | Bahia–Sergipe |
| 123419344 | GUARULHOS | SAOPAULO | Rio Grande do Sul |
| 73248792 | CURITIBA | PARANA | Pará–Amazonas–Acre–Amapá–Rondônia–Roraima |
| 78942456 | ELIASFAUSTO | SAOPAULO | Ceará–Maranhão–Piauí |
| 339197808 | FORTALEZA | CEARA | São Paulo |
| 54871997 | SAOPAULO | SAOPAULO | Rio Grande do Sul |
| 662924296 | ANGRADOSREIS | RIODEJANEIRO | Rio Grande do Sul |
| 38399368 | ANANINDEUA | PARA | Bahia–Sergipe |
| 378644008 | ASSIS | SAOPAULO | São Paulo |
| 283458923 | ARACATUBA | SAOPAULO | Ceará–Maranhão–Piauí |
| 334268000 | SIMOESEFILHO | BAHIA | Ceará–Maranhão–Piauí |
| 25177053 | IJACI | MINASGERAIS | Distrito Federal–Goiás–Mato Grosso–Mato Grosso... |

4.5. VERIFICAÇÃO DAS PARCELAS ATRASADAS E DIAS ATRASADOS:

Dos 13 CPF’s o cpf 11435403525 não tinha nenhum valor no arquivo “fatec_pgt” que fala sobre o período de pagamento das parcelas. As colunas “qtd_pcl_vnc” e “qtd_pcl_pgr” do mesmo arquivo de todos os CPF’s devem ser revistas, pois a maioria estão inteiras nulas.

Para saber os dias atrasados, pegou-se a data de pagamento e subtraiu-se a data de vencimento e sobrescreveu-se na coluna “nulo”. Valores menores ou iguais a zero são pagamentos feitos no prazo; maiores que zero são pagamentos atrasados. Na tabela 4A, observa-se o atraso de um pagamento em 29 dias pago em 11/12/2019 do cpf 05309680888. Na tabela 3B, mostra-se que no dia 12/12/2019 não havia parcelas atrasas, o que é válido se compararmos com a tabela 3A que diz que após a data 11/12/2019 não há atrasos. Com exceção da data 12/12/2019 todos os outros campos das colunas “qtd_pcl_vnc” e “qtd_pcl_pgr” estão nulos, o que configura um erro (tabela 4B).

Tabela 4: dados de dias atrasados (**A**) e dados das parcelas a vencer e a pagar (**B**) do 05309680888. **A:** Na coluna “nulo” estão os dias de pagamento em relação aos de vencimento (menor ou igual a zero, pagamento no prazo; maiores que zero, pagamento atrasado). Há um pagamento atrasado efetuado na data “11/12/2019”; **B:** repare que na data “12122019”, na coluna “dat_vct”, não existem parcelas em atraso ($63 - 63 = 0$). Se correlacionarmos as duas tabelas, não haverá discordância entre elas nos campos não nulos (63; 63) - **B** - e a data de atraso “2019-12-11” - **A**.

| A | nulo | id_opr_cad_pos | id_mvt_cad_pos | id_pgt_cad_pos | dat_pgt | dat_vct_tfm |
|----------|-----------|----------------|----------------|----------------|------------|-------------|
| | -123 days | 17317f69-f32 | fbdbb90b-3b7 | dcd9cf62-b76 | 2019-05-09 | 2019-09-09 |
| | -32 days | 17317f69-f32 | db9358de-89f | 2bff3301-c2b | 2019-09-10 | 2019-10-12 |
| | 29 days | 17317f69-f32 | 638c671d-8e3 | 73cdcf28-8fa | 2019-12-11 | 2019-11-12 |

| B | id_opr_cad_pos | id_mvt_cad_pos | dat_vct | qtd_pcl_vnc | qtd_pcl_pgr |
|----------|----------------|----------------|----------|-------------|-------------|
| | 17317f69-f32 | 5f33df5d-010 | 12122019 | 63 | 63 |
| | 17317f69-f32 | fbdbb90b-3b7 | 9092019 | NaN | NaN |
| | 17317f69-f32 | 638c671d-8e3 | 12112019 | NaN | NaN |
| | 17317f69-f32 | db9358de-89f | 12102019 | NaN | NaN |

Na avaliação do cpf 38553034011, não se tem a data das parcelas atrasadas no campo identificado como “NaN”, o que é um erro, porque não é possível cruzar as informações de dias atrasados com parcelas atrasadas (tabela 5).

Tabela 5: Erro na coluna “dat_vct” no campo “NaN” do 38553034011. Não é possível correlacionar essa informação com os dias atrasados para a validação. Além de haver valores nulos nas colunas “qtd_pcl_vnc” e “qtd_pcl_pgr”.

| dat_vct | qtd_pcl_vnc | qtd_pcl_pgr |
|----------|-------------|-------------|
| 25062019 | NaN | NaN |
| 25012019 | NaN | NaN |
| 25122018 | NaN | NaN |
| 25112018 | NaN | NaN |
| 25082019 | NaN | NaN |
| 25072019 | NaN | NaN |
| 25022019 | NaN | NaN |
| NaN | 0 | 0 |
| 25052019 | NaN | NaN |
| 25032019 | NaN | NaN |
| 25092019 | NaN | NaN |
| 25042019 | NaN | NaN |

O cpf 09639103101 diz que há uma parcela atrasada na data 10/12/2019, mas os dias atrasados resgatados deste CPF vão até o dia 08/12/2019. Há dados faltantes (tabela 6).

Tabela 6: Dados do 09639103101 registra os pagamentos até o dia “08/12/2019” - coluna “dat_pgt” – (A), mas há parcelas atrasadas no dia “10/12/2019” (B). Faltam dados na tabela 5A.

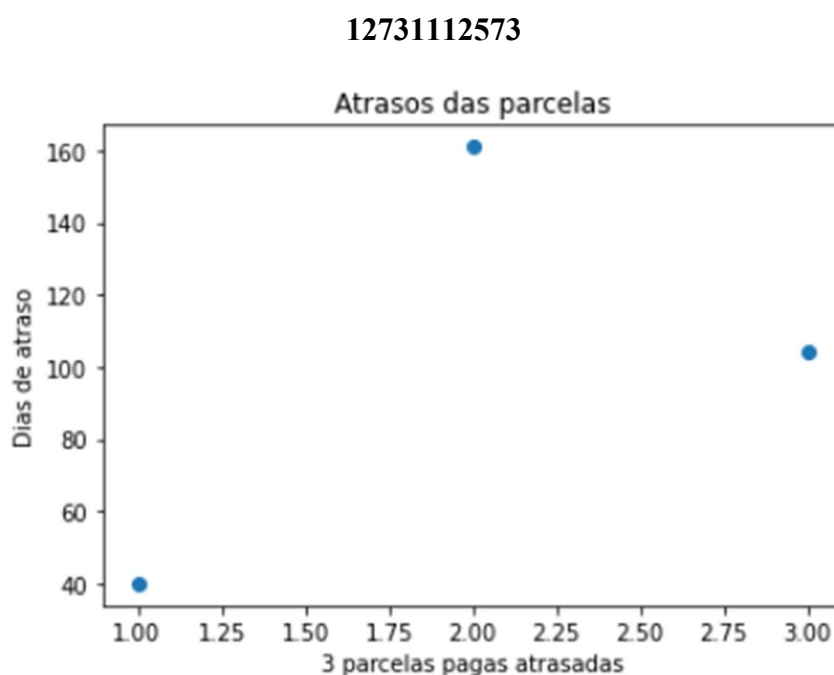
| A | nulo | id_opr_cad_pos | id_mvt_cad_pos | id_pgt_cad_pos | dat_pgt |
|-----------|--------------|----------------|----------------|----------------|---------|
| -183 days | 2dd72848-0cc | 75b762fc-465 | 831d635c-57d | 2018-05-11 | |
| -59 days | 2dd72848-0cc | 64ef1873-ac6 | 31d96803-336 | 2018-10-12 | |
| 7 days | 2dd72848-0cc | dfe32d8e-83b | cb3e481e-5aa | 2019-01-17 | |
| 5 days | 2dd72848-0cc | 92060acc-512 | b60487c0-bfb | 2019-02-15 | |
| 5 days | 2dd72848-0cc | 116c9441-70b | cbacf533-665 | 2019-04-15 | |
| 8 days | 2dd72848-0cc | ae8afb88-b27 | e8e6a570-12b | 2019-06-18 | |
| 3 days | 2dd72848-0cc | 3289e976-fc0 | 4bb701ee-635 | 2019-09-13 | |
| 148 days | 2dd72848-0cc | 66e165fb-aab | a8856bed-e84 | 2019-10-05 | |
| 4 days | 2dd72848-0cc | 0918777a-01b | 76a84975-a80 | 2019-10-14 | |
| 238 days | 2dd72848-0cc | b2bc90f5-f82 | 4409b9aa-1ea | 2019-11-03 | |
| 150 days | 2dd72848-0cc | fad28e64-b02 | 9d6df0b0-c40 | 2019-12-07 | |
| 120 days | 2dd72848-0cc | 18d4e571-e22 | 81ad4c7d-89a | 2019-12-08 | |

| B | dat_vct | qtd_pcl_vnc | qtd_pcl_pgr |
|---|----------|-------------|-------------|
| | 10042019 | NaN | NaN |
| | 10012019 | NaN | NaN |
| | 10032019 | NaN | NaN |
| | 10092019 | NaN | NaN |
| | 10082019 | NaN | NaN |
| | 10122019 | 8 | 9 |

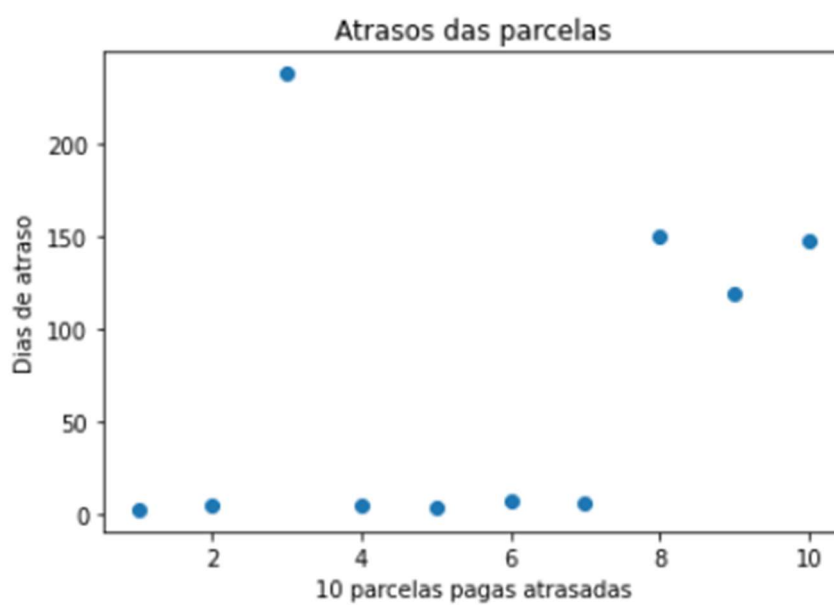
4.6. VISUALIZAÇÃO DO PERÍODO DE PAGAMENTO EM GRÁFICOS:

Construíram-se gráficos das tabelas do período de pagamento. Não há gráficos para o cpf 11435403525, por ausência de dados. Para cada CPF foi construído um gráfico mostrando os atrasos e outro mostrando os pagamentos feitos dentro do prazo.

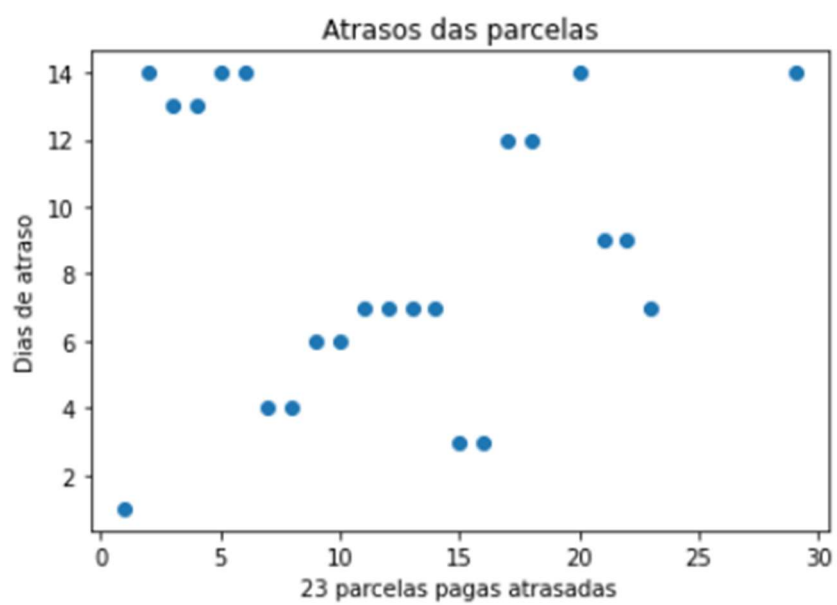
O eixo vertical mostra o total de dias. Se negativo ou zero, pagamentos feitos dentro do prazo. Se maior que zero, pagamentos feitos atrasados. O eixo horizontal mostra a quantidade de pagamentos, cada ponto plotado no gráfico refere-se a um pagamento. Agrupou-se os CPF's em relação aos atrasos: Sem atrasos: 64122018315, 53829829299, 28537130320, 38553034011; um atraso: 05227401381, 01367130760; dois atrasos: 18106900800; três atrasos: 12731112573, 24048813021; oito atrasos: 05309680888; dez atrasos: 09639103101; 23 atrasos: 06090237051. Lembrando que o 11435403525 não tem dados (p. 16-22).



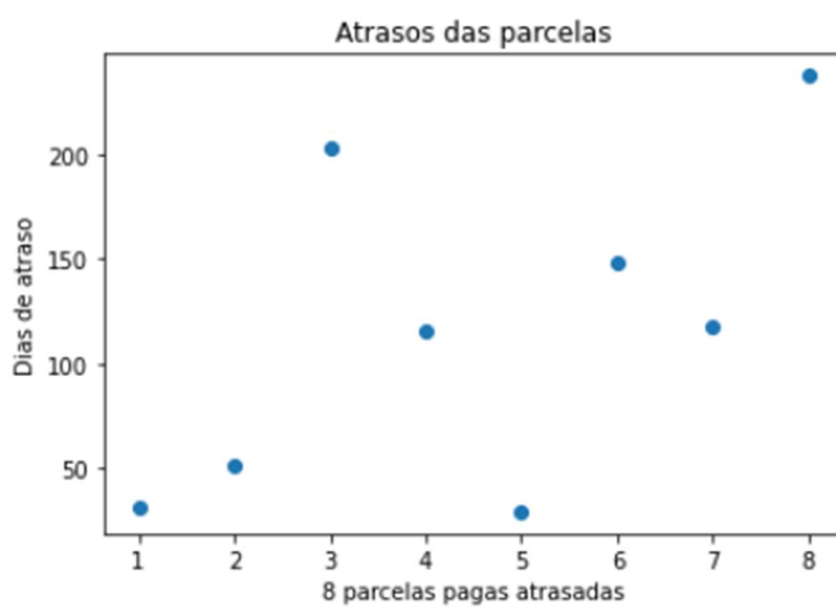
09639103101



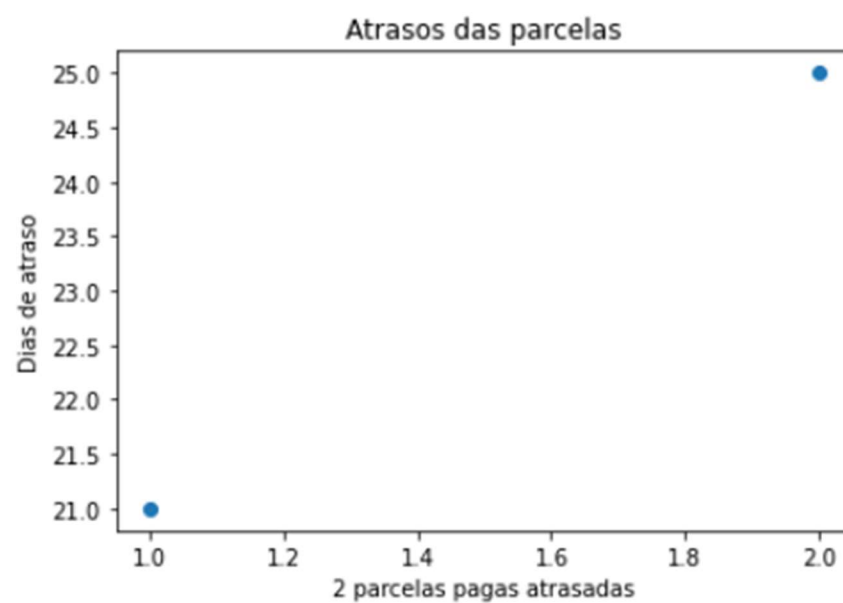
06090237051



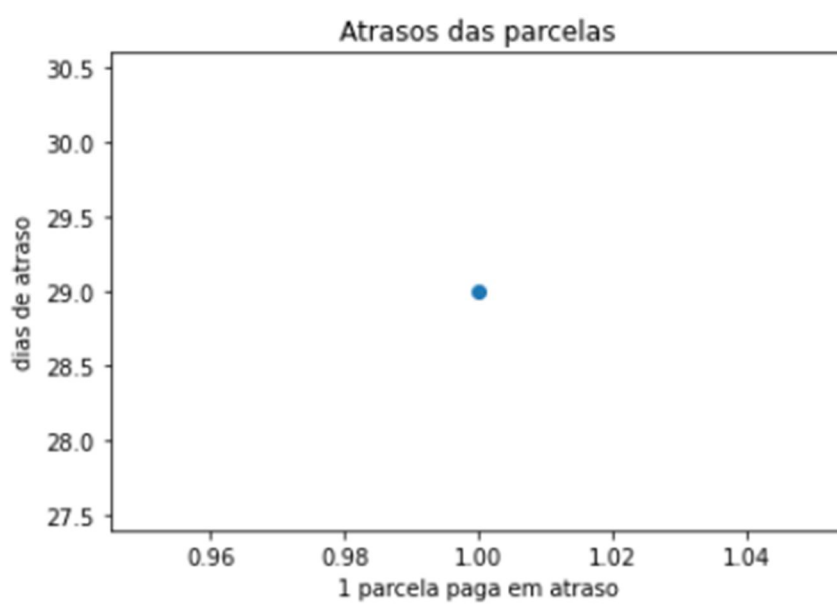
05309680888



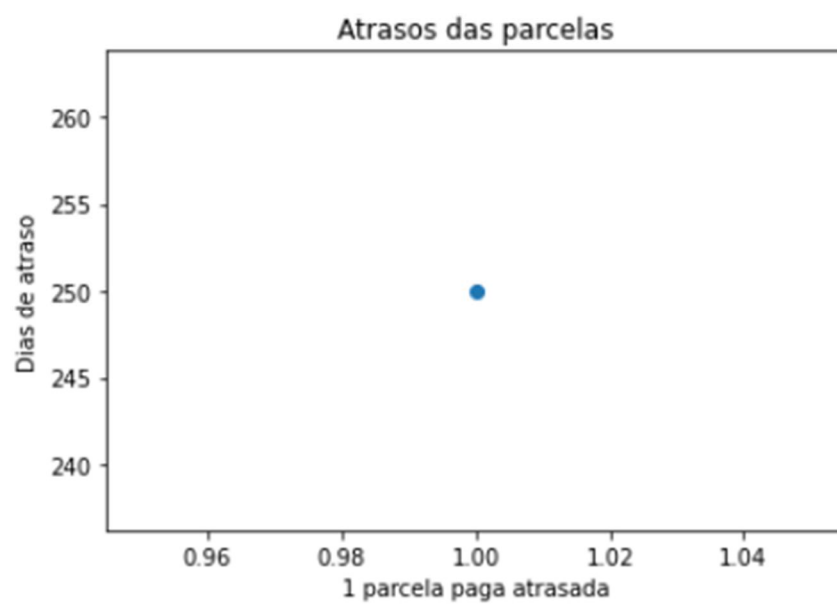
18106900800



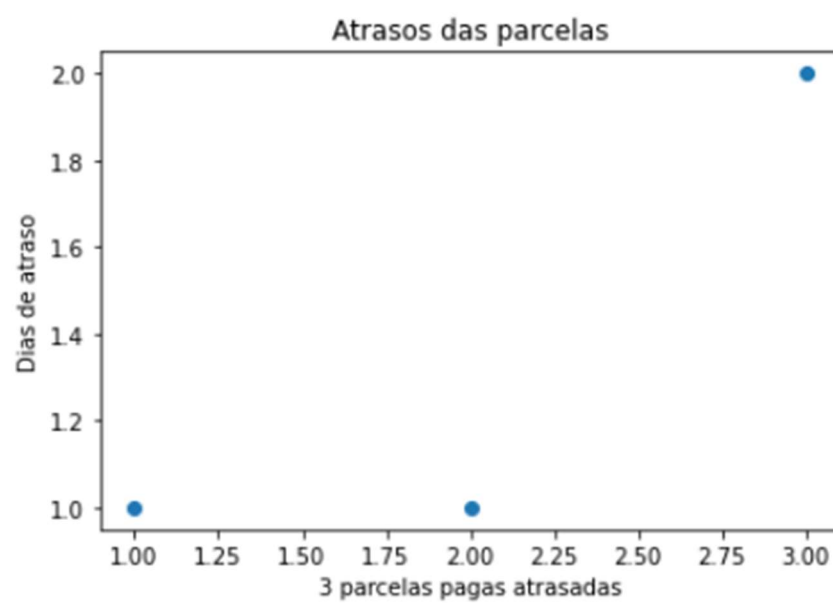
05227401381



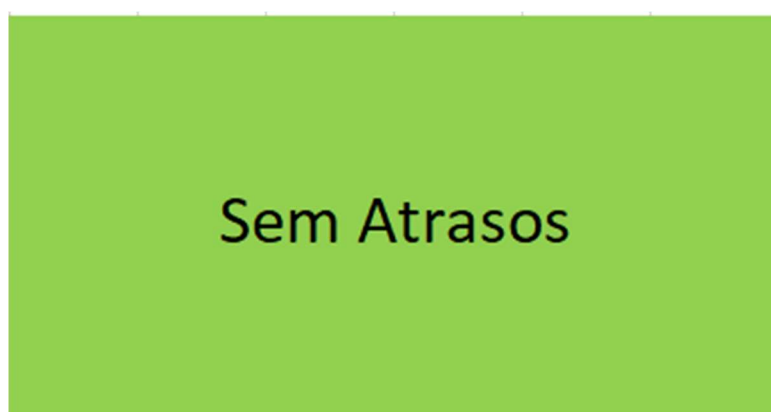
01367130760



24048813021



64122018315



53829829299

Sem Atrasos

28537130320

Sem Atrasos

38553034011

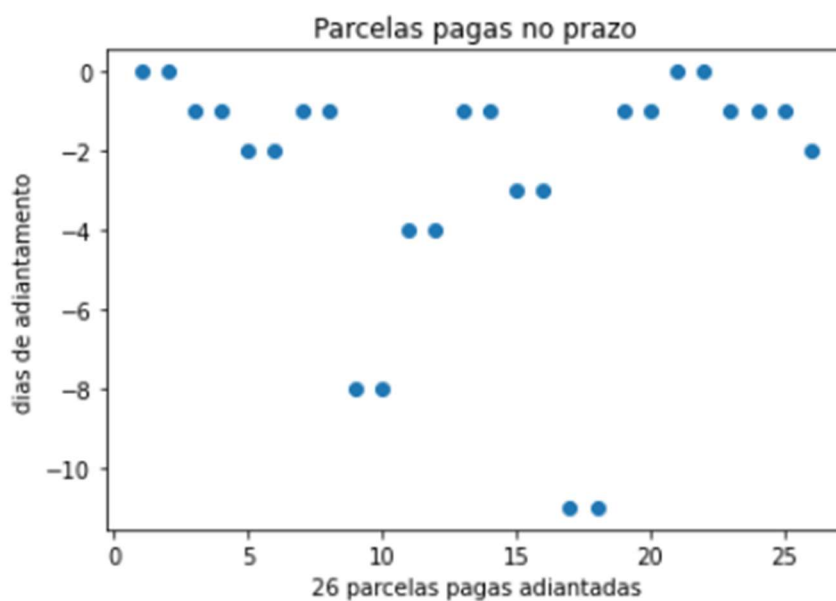
Sem Atrasos

11435403525

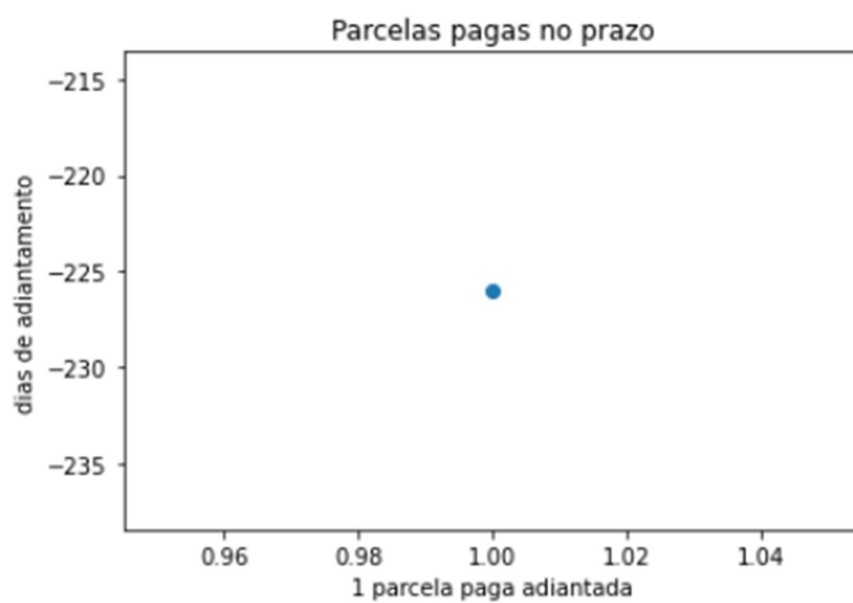
valores inesistentes

Os gráficos das páginas 22 até 29 mostram os pagamentos de cada CPF efetuados no prazo.

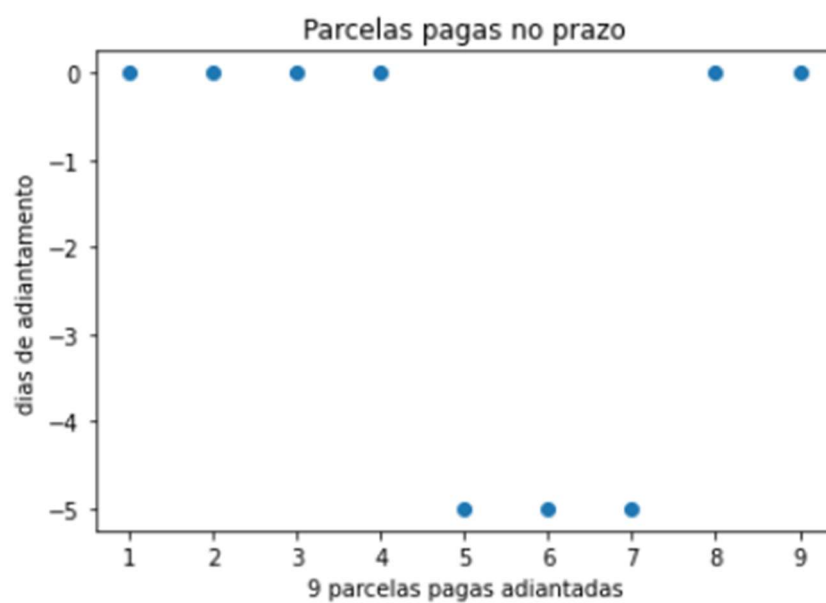
53829829299



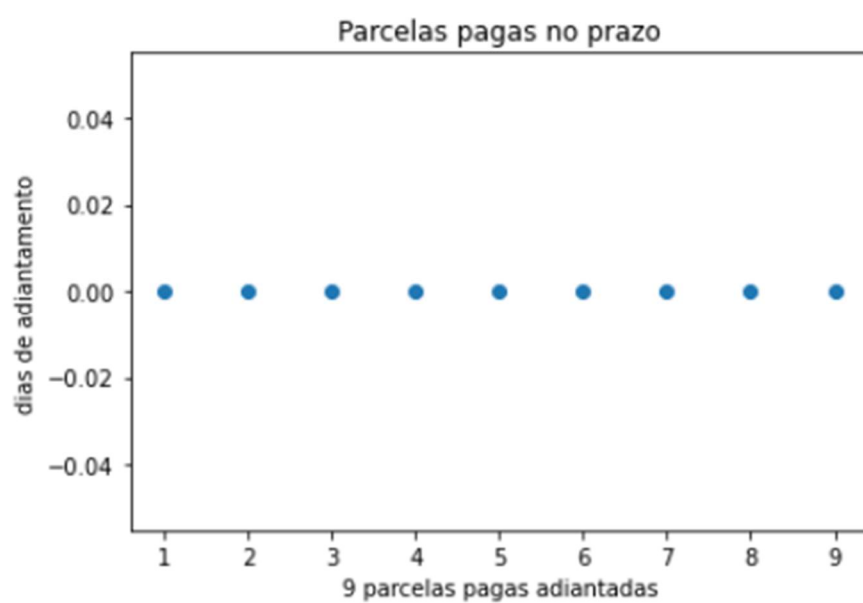
38553034011



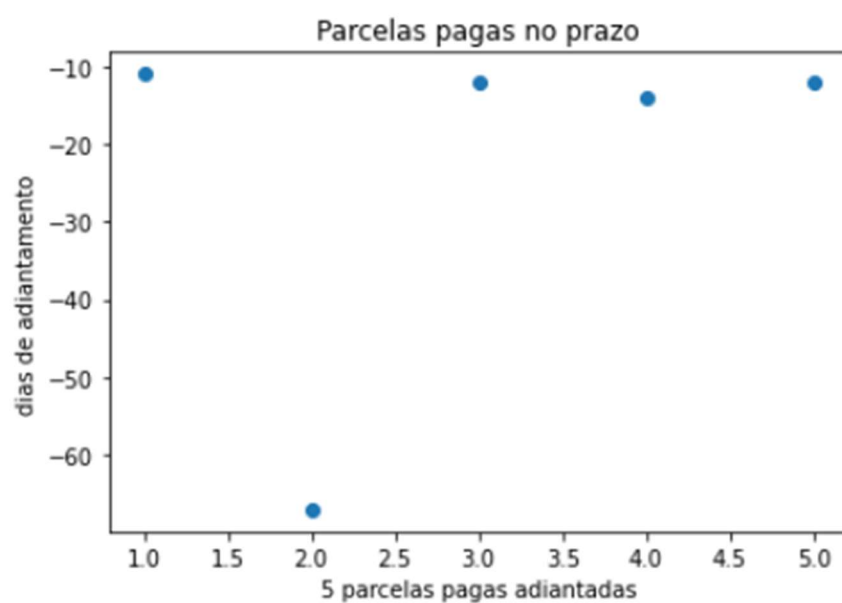
28537130320



24048813021



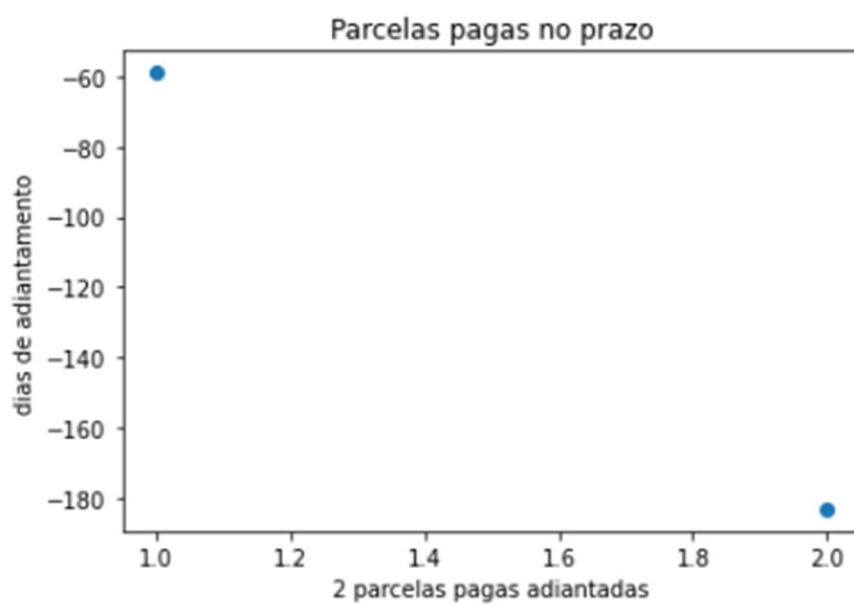
18106900800



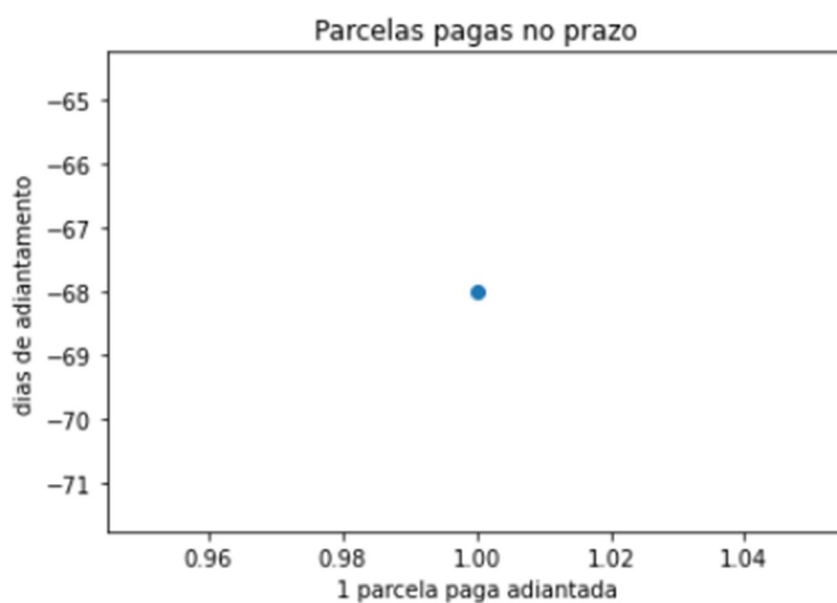
12731112573



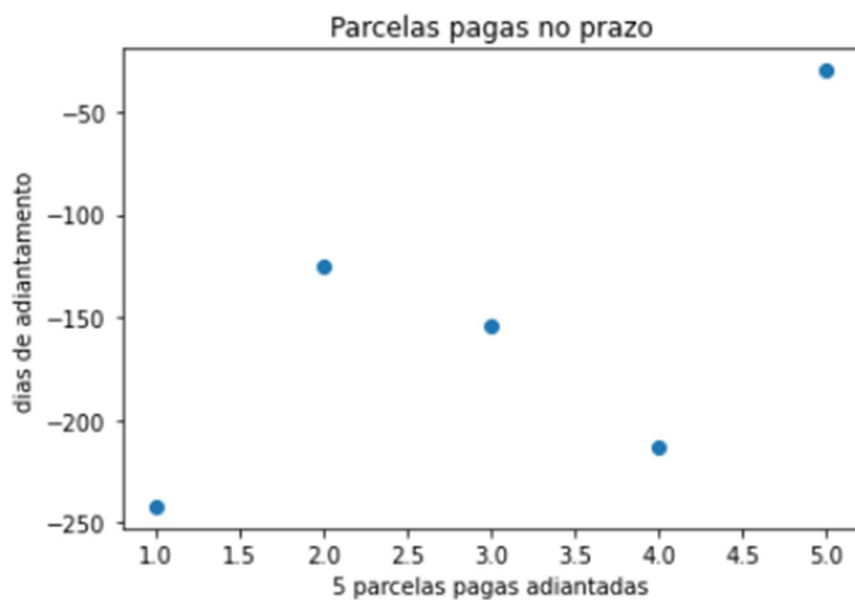
09639103101



06090237051



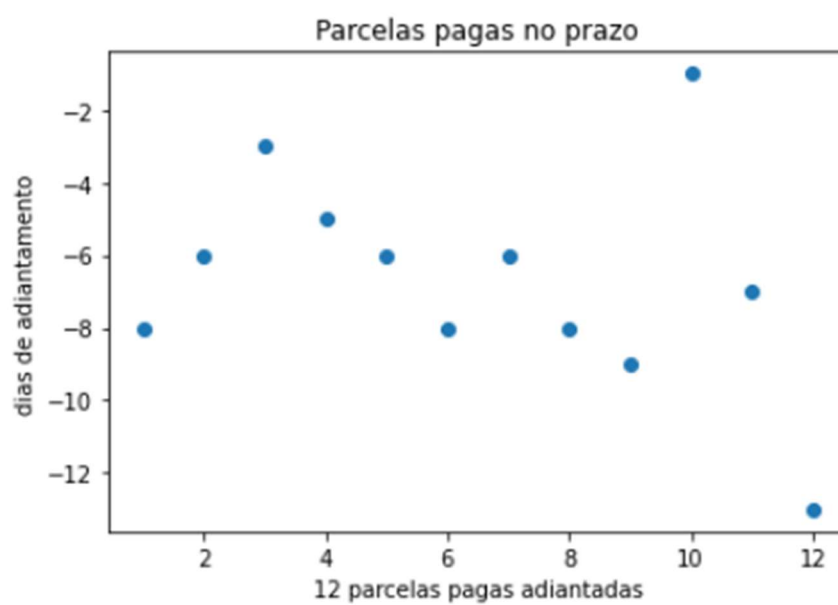
05309680888



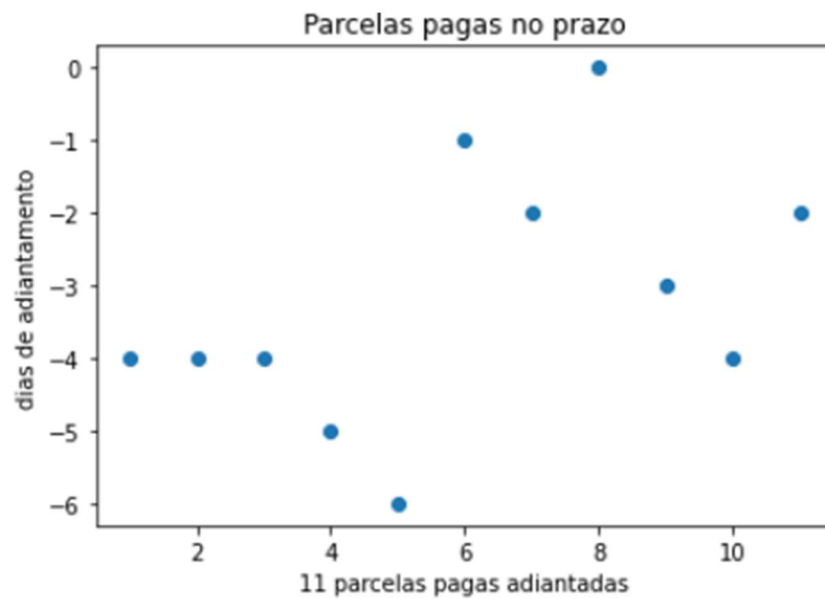
05227401381



01367130760



64122018315



11435403525

valores inesistentes

4.7.GRÁFICOS DOS ESTADOS:

O cpf 11435403525 apareceu no gráfico de estados com a localização de “estado atual” no Ceará e “estado de origem do CPF” na Bahia ou Sergipe; e o cpf 01367130760 não se tem documentado (Fig. 5 e 6).

Figura 5: Exibição gráfica dos estados de residência atual de cada pessoa física em um total de 12 CPF's. O 01367130760 não tem informação.

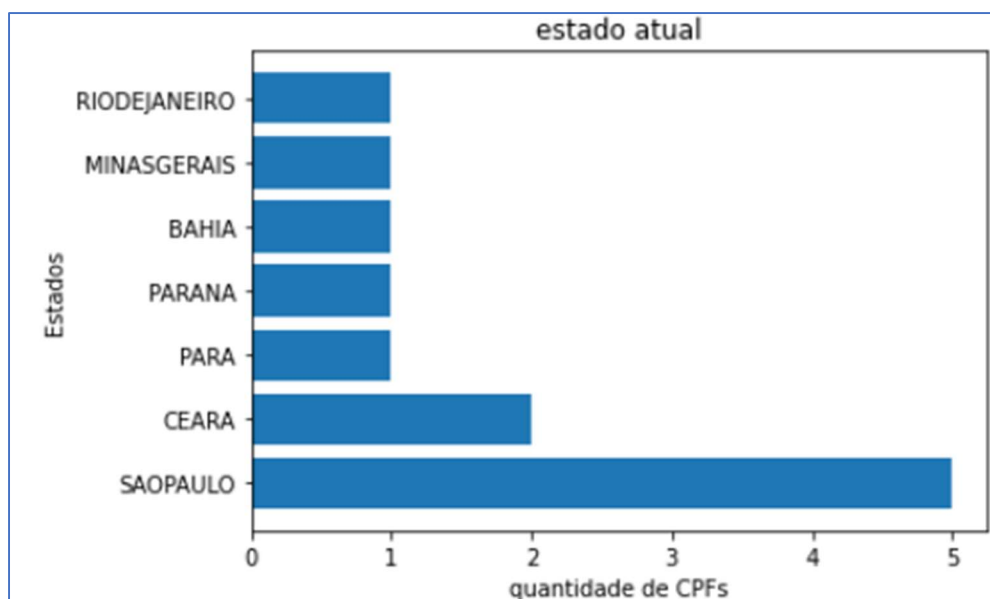
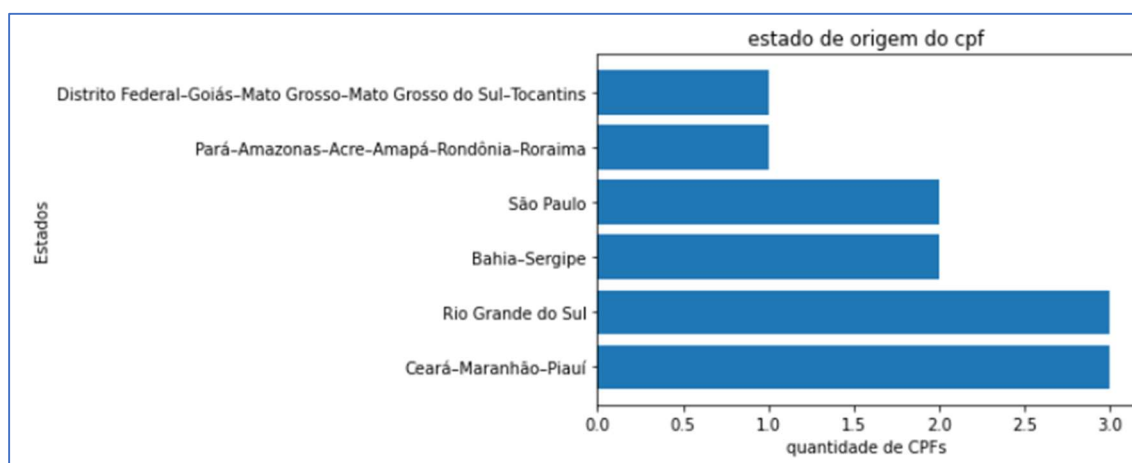


Figura 6: Exibição gráfica dos estados de origem de 12 dos CPF's. O 01367130760 não consta.



5. CONCLUSÃO:

Os únicos dados que se pode formular um algoritmo de validação foi o CPF e, a partir deles, rastreamos as operações de cada um. Houve muitos dados nulos que precisam ser corrigidos. A construção de gráficos do período de pagamento das parcelas deu um realismo às tabelas, reduzindo a característica caótica delas. As ferramentas utilizadas se mostraram eficiente para a conclusão deste trabalho.

Poucos CPF's foram considerados válidos, 13 CPF's; as outras 2989 linhas foram erros que precisam ser corrigidos. Todos os campos nulos são erros.

6. REFERÊNCIA BIBLIOGRÁFICA:

ALGORITMO PARA VALIDAR CPF. **Dicas de Programação**, 2015?. Disponível em: <[https://dicasdeprogramacao.com.br/algoritmo-para-validar-cpf/#:~:text=Regra%20para%20validar%20CPF&text=O%20CPF%20C3%A9%20formatado%20por,do%20sinal%20%22%2D%22\).](https://dicasdeprogramacao.com.br/algoritmo-para-validar-cpf/#:~:text=Regra%20para%20validar%20CPF&text=O%20CPF%20C3%A9%20formatado%20por,do%20sinal%20%22%2D%22).>)> Acesso em: 07/07/2020.