

## Importação das bibliotecas

```
In [7]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from functools import reduce
from sqlalchemy import create_engine
import psycpg2
import io
```

## Função para definir endereço das tabelas

```
In [8]: def path(trimestre, tabela):
    endereco = "C:/Users/grama/Desktop/TCC - Computação/Notebook/Dados/IF.data/"
    arquivo = "/dados ("
    final = ").csv"
    resultado = endereco + trimestre + arquivo + tabela + final
    return resultado
```

## Importação da tabela Resumo

```
In [9]: tabela = str(0)
dados = []

for i in range (25):
    if i < 3:
        dados.append(pd.read_csv(path(str(i), tabela), sep=';', header = 0, index_col
=False, thousands = '.', names = ['IF', 'COD', 'TCB', 'SR', 'TD', 'TC', 'CID', 'UF',
'DATA', 'ATIV_TOT', 'CC_CLASS', 'PASS_CIRC', 'CAPT', 'PATR_LIQ', 'LUCR_LIQ', 'PATR_RE
F', 'IND_BAS', 'IND_IMOB', 'NUM_AGE', 'NUM_PA']))
    else:
        dados.append(pd.read_csv(path(str(i), tabela), sep=';', header = 0, index_col
=False, thousands = '.', names = ['IF', 'COD', 'TCB', 'SR', 'TD', 'TC', 'CID', 'UF',
'DATA', 'ATIV_TOT', 'CC_CLASS', 'PASS_CIRC', 'CAPT', 'PATR_LIQ', 'LUCR_LIQ', 'NUM_AG
E', 'NUM_PA']))

resumo = pd.concat(dados, axis=0, ignore_index=True)
```

## Importação da tabela Ativo

```
In [10]: tabela = str(1)
dados1 = []

for i in range (25):
    dados1.append(pd.read_csv(path(str(i), tabela), sep=';', header = 1, index_col=Fa
lse, thousands='.', names = ['IF', 'COD', 'TCB', 'SR', 'TD', 'TC', 'CID', 'UF', 'DAT
A', 'DISP', 'APL_INTER', 'TVM_INST', 'OPER_CRE', 'OPER_CRE_PROV', 'OPER_CRE_LIQ', 'AR
RE_MERC_REC', 'ARRE_MERC_IMOB', 'ARRE_MERC_CRED', 'ARRE_MERC_PROV', 'ARRE_MERC_LIQ',
'OUT_CRE_LIQ', 'OUT_ATIV_REAL', 'PERM_AJUS', 'ATIV_TOT_AJUST', 'CRED_ATENC', 'ATIV_TO
T']))

ativo = pd.concat(dados1, axis=0, ignore_index=True)
```

## Importação da tabela Passivo

```
In [14]: tabela = str(2)
dados2 = []

for i in range(25):
    if i < 24:
        dados2.append(pd.read_csv(path(str(i), tabela), sep=';', header = 2, index_col=False, thousands='.', names = ['IF', 'COD', 'TCB', 'SR', 'TD', 'TC', 'CID', 'UF', 'DATA', 'DEP_A_VIS', 'DEP_POUP', 'DEP_INT', 'DEP_PRAZ', 'DEP_OUT', 'DEP_TOT', 'OBR_OPER_COMP', 'LET_CRED_IMOB', 'LET_CRED_AGRO', 'LET_FIN', 'OBR_TIT_EXT', 'OUT_REC_ACEI', 'REC_ACEI_EM_TIT', 'OBRIG_EMP_REP', 'CAPT', 'INST_DER', 'OUT_OBRIG', 'PASS_CIRC_EXIG', 'RES_EXEC_FUT', 'PATR_LIQ', 'PASS_TOT']))
    else:
        dados2.append(pd.read_csv(path(str(i), tabela), sep=';', header = 2, index_col=False, thousands='.', names = ['IF', 'COD', 'TCB', 'SR', 'TD', 'TC', 'CID', 'UF', 'DATA', 'DEP_A_VIS', 'DEP_POUP', 'DEP_INT', 'DEP_PRAZ', 'CONT_PAG_PRE_PAG', 'DEP_OUT', 'DEP_TOT', 'OBR_OPER_COMP', 'LET_CRED_IMOB', 'LET_CRED_AGRO', 'LET_FIN', 'OBR_TIT_EXT', 'OUT_REC_ACEI', 'REC_ACEI_EM_TIT', 'OBRIG_EMP_REP', 'CAPT', 'INST_DER', 'OUT_OBRIG', 'PASS_CIRC_EXIG', 'RES_EXEC_FUT', 'PATR_LIQ', 'PASS_TOT']))

passivo = pd.concat(dados2, axis=0, ignore_index=True)
```

## Importação da tabela Demonstração de Resultado

```
In [15]: tabela = str(3)
dados3 = []
nomes3 = ['IF', 'COD', 'TCB', 'SR', 'TD', 'TC', 'CID', 'UF', 'DATA', 'REND_OPER_CRED', 'REND_OPER_ARRE', 'REND_OPER_TVM', 'REND_OPER_INST_FIN', 'REN_OPER_CAMB', 'REND_APLIC_COMP', 'REC_INTER_FIN', 'DESP_CAPT', 'DESP_OBRIG_EMP', 'DESP_OPER_ARRE', 'DESP_OPER_CAMB', 'RES_PROV_CRED', 'DESP_INTER_FIN', 'RES_INTER_FIN', 'REN_PREST_SERV', 'REN_TAR_BANC', 'DESP_PESS', 'DESP_ADM', 'DESP_TRIB', 'RES_PART', 'OUT_REC_OPER', 'OUT_DESP_OPER', 'OUT_REC_DESP_OPER', 'RES_OPER', 'RES_N_OPER', 'RES_ANT_TRIB', 'IMP_REN_CONSOC', 'PART_LUC', 'LUC_LIQ', 'JUR_SOB_CAP']

for i in range(25):
    dados3.append(pd.read_csv(path(str(i), tabela), sep=';', header = 2, index_col=False, thousands='.', names = nomes3))

dem_res = pd.concat(dados3, axis=0, ignore_index=True)
```

## Importação da tabela Carteira de crédito ativa Pessoa Física - modalidade e prazo de vencimento

```
In [16]: tabela = str(4)
dados4 = []
nomes4 = ['IF', 'COD', 'TCB', 'TD', 'TC', 'SR', 'SEGM', 'CID', 'UF', 'DATA', 'TOT_CAR_PF', 'CONSIG_VENC_15', 'CONSIG_VINC_90', 'CONSIG_VINC_360', 'CONSIG_VINC_1080', 'CONSIG_VINC_1800', 'CONSIG_VINC_5400', 'CONSIG_VINC_ACI', 'CONSIG_TOT', 'N_CONSIG_VENC15', 'N_CONSIG_VINC90', 'N_CONSIG_VINC_360', 'N_CONSIG_VINC_1080', 'N_CONSIG_VINC_1800', 'N_CONSIG_VINC_5400', 'N_CONSIG_VINC_ACI', 'N_CONSIG_TOT', 'VEIC_VENC15', 'VEIC_VINC90', 'VEIC_VINC_360', 'VEIC_VINC_1080', 'VEIC_VINC_1800', 'VEIC_VINC_5400', 'VEIC_VINC_ACI', 'VEIC_TOT', 'HAB_VENC15', 'HAB_VINC90', 'HAB_VINC_360', 'HAB_VINC_1080', 'HAB_VINC_1800', 'HAB_VINC_5400', 'HAB_VINC_ACI', 'HAB_TOT', 'CART_CRED_VENC15', 'CART_CRED_VINC90', 'CART_CRED_VINC_360', 'CART_CRED_VINC_1080', 'CART_CRED_VINC_1800', 'CART_CRED_VINC_5400', 'CART_CRED_VINC_ACI', 'CART_CRED_TOT', 'RUR_VENC15', 'RUR_VINC_90', 'RUR_VINC_360', 'RUR_VINC_1080', 'RUR_VINC_1800', 'RUR_VINC_5400', 'RUR_VINC_ACI', 'RUR_TOT', 'OUT_PF_VENC15', 'OUT_PF_VINC90', 'OUT_PF_VINC_360', 'OUT_PF_1080', 'OUT_PF_1800', 'OUT_PF_5400', 'OUT_PF_ACI', 'OUT_PF_TOT', 'TOT_EXT_PF']

for i in range(25):
    dados4.append(pd.read_csv(path(str(i), tabela), sep=';', header = 1, index_col=False, thousands='.', names = nomes4))

cred_pf = pd.concat(dados4, axis=0, ignore_index=True)
```

### Importação da tabela Carteira de crédito ativa Pessoa Jurídica - modalidade e prazo de vencimento

```
In [17]: tabela = str(5)
dados5 = []
nomes5 = ['IF', 'COD', 'TCB', 'TD', 'TC', 'SR', 'SEGM', 'CID', 'UF', 'DATA', 'TOT_CAR_PJ', 'CAP_GIR_VENC_15', 'CAP_GIR_VINC_90', 'CAP_GIR_VINC_360', 'CAP_GIR_VINC_1080', 'CAP_GIR_VINC_1800', 'CAP_GIR_VINC_5400', 'CAP_GIR_VINC_ACI', 'CAP_GIR_TOT', 'INVEST_VENC15', 'INVEST_VINC90', 'INVEST_VINC_360', 'INVEST_VINC_1080', 'INVEST_VINC_1800', 'INVEST_VINC_5400', 'INVEST_VINC_ACI', 'INVEST_TOT', 'GIR_ROT_VENC15', 'GIR_ROT_VINC90', 'GIR_ROT_VINC_360', 'GIR_ROT_VINC_1080', 'GIR_ROT_VINC_1800', 'GIR_ROT_VINC_5400', 'GIR_ROT_VINC_ACI', 'GIR_ROT_TOT', 'REC_VENC15', 'REC_VINC90', 'REC_VINC_360', 'REC_VINC_1080', 'REC_VINC_1800', 'REC_VINC_5400', 'REC_VINC_ACI', 'REC_TOT', 'COM_EX_VENC15', 'COM_EX_VINC90', 'COM_EX_VINC_360', 'COM_EX_VINC_1080', 'COM_EX_VINC_1800', 'COM_EX_VINC_5400', 'COM_EX_VINC_ACI', 'COM_EX_TOT', 'OUT_PJ_VENC15', 'OUT_PJ_VINC90', 'OUT_PJ_VINC_360', 'OUT_PJ_VINC_1080', 'OUT_PJ_VINC_1800', 'OUT_PJ_VINC_5400', 'OUT_PJ_VINC_ACI', 'OUT_PJ_TOT', 'FIN_INF_DES_VENC15', 'FIN_INF_DES_VINC90', 'FIN_INF_DES_VINC_360', 'FIN_INF_DES_1080', 'FIN_INF_DES_1800', 'FIN_INF_DES_5400', 'FIN_INF_DES_ACI', 'FIN_INF_DES_TOT', 'RUR_PJ_VENC_15', 'RUR_PJ_VINC_90', 'RUR_PJ_VINC_360', 'RUR_PJ_VINC_1080', 'RUR_PJ_VINC_1800', 'RUR_PJ_VINC_5400', 'RUR_PJ_VINC_ACI', 'RUR_PJ_TOT', 'HAB_PJ_VENC_15', 'HAB_PJ_VINC_90', 'HAB_PJ_VINC_360', 'HAB_PJ_VINC_1080', 'HAB_PJ_VINC_1800', 'HAB_PJ_VINC_5400', 'HAB_PJ_VINC_ACI', 'HAB_PJ_TOT', 'TOT_EXT_PJ']

for i in range(25):
    dados5.append(pd.read_csv(path(str(i), tabela), sep=';', header = 1, index_col=False, thousands='.', names = nomes5))

cred_pj = pd.concat(dados5, axis=0, ignore_index=True)
```

### Importação da tabela Carteira de crédito ativa Pessoa Jurídica - por atividade econômica (CNAE)

```
In [18]: tabela = str(6)
dados6 = []
nomes6 = ['IF', 'COD', 'TCB', 'TD', 'TC', 'SR', 'SEGM', 'CID', 'UF', 'DATA', 'TOT_CAR_PJ', 'AGR_PEC_VENC_15', 'AGR_PEC_VINC_90', 'AGR_PEC_VINC_360', 'AGR_PEC_VINC_1080', 'AGR_PEC_VINC_1800', 'AGR_PEC_VINC_5400', 'AGR_PEC_VINC_ACI', 'AGR_PEC_TOT', 'IND_TRANS_VENC15', 'IND_TRANS_VINC90', 'IND_TRANS_VINC_360', 'IND_TRANS_VINC_1080', 'IND_TRANS_VINC_1800', 'IND_TRANS_VINC_5400', 'IND_TRANS_VINC_ACI', 'IND_TRANS_TOT', 'CONST_VENC15', 'CONST_VINC90', 'CONST_VINC_360', 'CONST_VINC_1080', 'CONST_VINC_1800', 'CONST_VINC_5400', 'CONST_VINC_ACI', 'CONST_TOT', 'SER_IND_VENC15', 'SER_IND_VINC90', 'SER_IND_VINC_360', 'SER_IND_VINC_1080', 'SER_IND_VINC_1800', 'SER_IND_VINC_5400', 'SER_IND_VINC_ACI', 'SER_IND_TOT', 'IND_EXTRAT_VENC15', 'IND_EXTRAT_VINC90', 'IND_EXTRAT_VINC_360', 'IND_EXTRAT_VINC_1080', 'IND_EXTRAT_VINC_1800', 'IND_EXTRAT_VINC_5400', 'IND_EXTRAT_VINC_ACI', 'IND_EXTRAT_TOT', 'COM_REP_VEIC_VENC15', 'COM_REP_VEIC_VINC90', 'COM_REP_VEIC_VINC_360', 'COM_REP_VEIC_VINC_1080', 'COM_REP_VEIC_VINC_1800', 'COM_REP_VEIC_VINC_5400', 'COM_REP_VEIC_VINC_ACI', 'COM_REP_VEIC_TOT', 'ADM_PUB_VENC15', 'ADM_PUB_VINC90', 'ADM_PUB_VINC_360', 'ADM_PUB_1080', 'ADM_PUB_1800', 'ADM_PUB_5400', 'ADM_PUB_ACI', 'ADM_PUB_TOT', 'TRANS_ARMAZ_VENC_15', 'TRANS_ARMAZ_VINC_90', 'TRANS_ARMAZ_VINC_360', 'TRANS_ARMAZ_VINC_1080', 'TRANS_ARMAZ_VINC_1800', 'TRANS_ARMAZ_VINC_5400', 'TRANS_ARMAZ_VINC_ACI', 'TRANS_ARMAZ_TOT', 'OUT_CNAE_VENC_15', 'OUT_CNAE_VINC_90', 'OUT_CNAE_VINC_360', 'OUT_CNAE_VINC_1080', 'OUT_CNAE_VINC_1800', 'OUT_CNAE_VINC_5400', 'OUT_CNAE_VINC_ACI', 'OUT_CNAE_TOT', 'ATIV_N_INFOR', 'TOT_N_INDIV_PJ', 'TOT_EXT_PJ']

for i in range (25):
    dados6.append(pd.read_csv(path(str(i), tabela), sep=';', header = 1, index_col=False, thousands='.', names = nomes6))

pj_cnae = pd.concat(dados6,axis=0, ignore_index=True)
```

### Importação da tabela Carteira de crédito ativa Pessoa Jurídica - por porte do tomador

```
In [19]: tabela = str(7)
dados7 = []
nomes7 = ['IF', 'COD', 'TCB', 'TD', 'TC', 'SR', 'SEGM', 'CID', 'UF', 'DATA', 'TOT_CAR_PJ', 'MICRO', 'PEQ', 'MEDIA', 'GRANDE', 'INDISP', 'TOT_N_INDIV_PJ', 'TOT_EXT_PJ']

for i in range (25):
    dados7.append(pd.read_csv(path(str(i), tabela), sep=';', header = 0, index_col=False, thousands='.', names = nomes7))

pj_porte = pd.concat(dados7,axis=0, ignore_index=True)
```

### Importação da tabela Carteira de crédito ativa - quantidade de clientes e de operações

```
In [20]: tabela = str(8)
dados8 = []
nomes8 = ['IF', 'COD', 'TCB', 'TD', 'TC', 'SR', 'SEGM', 'CID', 'UF', 'DATA', 'QTD_CLIENTES', 'QTD_OPER']

for i in range (25):
    dados8.append(pd.read_csv(path(str(i), tabela), sep=';', header = 0, index_col=False, thousands='.', names = nomes8))

cred_qtd = pd.concat(dados8,axis=0, ignore_index=True)
```

### Importação da tabela Carteira de crédito ativa - por nível de risco da operação

```
In [21]: tabela = str(9)
dados9 = []
nomes9 = ['IF', 'COD', 'TCB', 'TD', 'TC', 'SR', 'SEGM', 'CID', 'UF', 'DATA', 'TOT_GER', 'RISCO_AA', 'RISCO_A', 'RISCO_B', 'RISCO_C', 'RISCO_D', 'RISCO_E', 'RISCO_F', 'RISCO_G', 'RISCO_H', 'TOT_EXT']

for i in range (25):
    dados9.append(pd.read_csv(path(str(i), tabela), sep=';', header = 0, index_col=False, thousands='.', names = nomes9))

cred_risco = pd.concat(dados9,axis=0, ignore_index=True)
```

### Importação da tabela Carteira de crédito ativa - por indexador

```
In [22]: tabela = str(10)
dados10 = []
nomes10 = ['IF', 'COD', 'TCB', 'TD', 'TC', 'SR', 'SEGM', 'CID', 'UF', 'DATA', 'TOT_GER', 'PREFIX', 'TR_TBF', 'TJLP', 'TLP', 'LIBOR', 'OUTR_TAX_POS_FIX', 'CART_ATIV_CDI', 'SELIC', 'OUT_TAX_FLUT', 'IGPM', 'IPCA', 'IPCC', 'OUT_IND_PREC', 'OUT_INDEX', 'TOT_N_INDIV', 'TOT_EXT']

for i in range (25):
    dados10.append(pd.read_csv(path(str(i), tabela), sep=';', header = 0, index_col=False, thousands='.', names = nomes10))

cred_index = pd.concat(dados10,axis=0, ignore_index=True)
```

### Importação da tabela Carteira de crédito ativa - por região geográfica

```
In [23]: tabela = str(11)
dados11 = []
nomes11 = ['IF', 'COD', 'TCB', 'TD', 'TC', 'SR', 'SEGM', 'CID', 'UF', 'DATA', 'TOT_GER', 'SUDESTE', 'CENTRO_OESTE', 'NORDESTE', 'NORTE', 'SUL', 'REG_N_INFOR', 'TOT_EXT']

for i in range (1,25):
    dados11.append(pd.read_csv(path(str(i), tabela), sep=';', header = 0, index_col=False, thousands='.', names = nomes11))

cred_reg_geog = pd.concat(dados11,axis=0, ignore_index=True)
```

### Remoção de linhas com COD NaN e substituição de NI por NaN

```
In [24]: tabelas = [resumo, ativo, passivo, dem_res, cred_pf, cred_pj, pj_cnae, pj_porte, cred_qtd, cred_risco, cred_index, cred_reg_geog]

for i in range(11):
    tabelas[i] = tabelas[i].dropna(subset=['COD', 'DATA'])
    tabelas[i] = tabelas[i].replace({'NI': np.nan})
```

### Junção das tabelas pela chave 'IF', 'COD', 'TCB', 'SR', 'TD', 'TC', 'CID', 'UF', 'DATA'

```
In [25]: chave = ['IF', 'COD', 'TCB', 'SR', 'TD', 'TC', 'CID', 'UF', 'DATA']

juncao = reduce(lambda left,right: pd.merge(left,right,on=chave, suffixes=('', '_y'
)), tabelas)
juncao.drop(juncao.filter(regex='_y$').columns.tolist(),axis=1, inplace=True)
```

### Criação da coluna INAD\_TOT (soma das inadimplências PF e PJ)

```
In [26]: colunas_inad = ['CONSIG_VENC_15', 'N_CONSIG_VENC15', 'VEIC_VENC15', 'HAB_VENC15', 'CA
RT_CRED_VENC15', 'RUR_VENC15', 'OUT_PF_VENC15', 'CAP_GIR_VENC_15', 'INVEST_VENC15',
'GIR_ROT_VENC15', 'REC_VENC15', 'COM_EX_VENC15', 'OUT_PJ_VENC15', 'FIN_INF_DES_VENC1
5', 'RUR_PJ_VENC_15', 'HAB_PJ_VENC_15']
juncao['INAD_TOT'] = juncao[colunas_inad].sum(axis=1)
```

### Conversão do campo DATA para o formato datetime

```
In [30]: juncao.DATA = pd.to_datetime(juncao.DATA)
```

### Exportação da tabela final em formato csv

```
In [27]: juncao.to_csv('juncao_06.csv')
```

### Carrega tabela em banco de dados PostgreSQL

```
In [31]: engine = create_engine('postgresql+psycopg2://postgres:admin@localhost:5432/base_ifda
ta')

juncao.head(0).to_sql('ifdata_juncao_06', engine, if_exists='replace',index=True)

conn = engine.raw_connection()
cur = conn.cursor()
output = io.StringIO()
juncao.to_csv(output, sep='\t', header=False, index=True)
output.seek(0)
contents = output.getvalue()
cur.copy_from(output, 'ifdata_juncao_06', null='')
conn.commit()
```