EDA_Buencafe-6 (2) (1)

July 6, 2022

1 Lyophilized Buencafé

The following is a first approach to the databases obtained from Buencafé.

1.1 Concentrator 1 (C1)

Analyze the process of the concentrator 1 to determine the best real drain (efficiency for them) considering the other variables that take part of the process (temperature, current, pressure, time, conductivity, these variables are in different equipment that make up the process such as the TAP, receiver, crystallizer, recrystallizer and the columns).

2 Concentrator 1 process

The graphical representation of Concentrator 1 will be shown below. This machine performs a cryoconcentration process, i.e., it generates water crystals causing the coffee extract to have a higher concentration percentage.

Initially the data and its corresponding type will be read

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api         as sm
import statsmodels.formula.api as smf
```

Visualization and loading of the data delivered for concentrator 1

The three datasets that were delivered by the BuenCafe team are loaded and after this the exploratory analysis begins.

- To know the relationship between dataset 1 and dataset 2 and compare their initial data since they have the same start date.
- A concentration of null data is evidenced in the first samples of the datasets.
- Complex names are identified in each of the columns.

[2]:
```
recristalizador = pd.read_csv('data/TableFloatConc1-1.csv', delimiter=',')
recristalizador.head(5)
```

[2]:

	Time	[CONCENTRACION1]_11VM1Recristalizador1Corriente_AI.Valor	[CONCENTRACION1]_21VM1Recristalizador2Corriente_AI.Valor	[CONCENTRACION1]_31VM1Recristalizador3Corriente_AI.Valor	[CONCENTRACION1]_11V1Recrist1Temp_TT1231_AI.Valor	[CONCENTRACION1]_21V1Recrist2Temp_TT2231_AI.Valor	[CONCENTRACION1]_31V1Recrist3Temp_TT3231_AI.Valor
0	24/02/2022 12:00:27,945 a.m.	NaN	NaN	NaN	NaN	NaN	NaN
1	24/02/2022 12:01:47,954 a.m.	22,70	NaN	NaN	NaN	NaN	NaN
2	24/02/2022 12:04:47,914 a.m.	22,70	NaN	36,45	NaN	NaN	NaN
3	24/02/2022 12:05:17,937 a.m.	22,70	NaN	36,45	NaN	NaN	NaN
4	24/02/2022 12:05:47,918 a.m.	22,70	NaN	36,45	NaN	NaN	NaN

```
4                                                 NaN

  [CONCENTRACION1]_16V01Recibidor1Temp_TT1120_AI.Valor  \
0                                                  NaN
1                                                  NaN
2                                                  NaN
3                                                  NaN
4                                                  NaN

  [CONCENTRACION1]_26V01Recibidor2Temp_TT2120_AI.Valor  \
0                                                  NaN
1                                                  NaN
2                                                  NaN
3                                                  NaN
4                                               -33,23

  [CONCENTRACION1]_36V01Recibidor3Temp_TT3120_AI.Valor  \
0                                                  NaN
1                                                  NaN
2                                                  NaN
3                                               -39,51
4                                               -39,51

  [CONCENTRACION1]_44V01TAPTemp_TT4401_AI.Valor  \
0                                            NaN
1                                            NaN
2                                            NaN
3                                            NaN
4                                            NaN

  [CONCENTRACION1]_11V1Recrist1Presion_PT1105_AI.Valor  \
0                                                  NaN
1                                                  NaN
2                                                  NaN
3                                                  NaN
4                                                  NaN

  [CONCENTRACION1]_21V1Recrist2Presion_PT2105_AI.Valor  \
0                                                  NaN
1                                                  NaN
2                                                  NaN
3                                                  NaN
4                                                  NaN

  [CONCENTRACION1]_31V1Recrist3Presion_PT3105_AI.Valor
0                                               2,37
1                                               2,37
```

```
2                                                 2,37
3                                                 2,37
4                                                 2,37
```

[3]:
```
cristalizador = pd.read_csv('data/TableFloatConc1-2.csv', delimiter=',')
cristalizador.head(5)
```

[3]:
```
                          Time  \
0  24/02/2022 12:00:17,908 a.m.
1  24/02/2022 12:02:27,929 a.m.
2  24/02/2022 12:02:47,958 a.m.
3  24/02/2022 12:21:27,911 a.m.
4  24/02/2022 12:26:27,927 a.m.

  [CONCENTRACION1]_16V01Recibidor1Presion_PT1108_AI.Valor  \
0                                                NaN
1                                                NaN
2                                                NaN
3                                                NaN
4                                                NaN

  [CONCENTRACION1]_26V01Recibidor2Presion_PT2108_AI.Valor  \
0                                                NaN
1                                                NaN
2                                                NaN
3                                                NaN
4                                                NaN

  [CONCENTRACION1]_36V01Recibidor3Presion_PT3108_AI.Valor  \
0                                                NaN
1                                                NaN
2                                                NaN
3                                                NaN
4                                                NaN

  [CONCENTRACION1]_12EM1Cristalizador1Corriente_AI.Valor  \
0                                                NaN
1                                               7,36
2                                               7,36
3                                               7,36
4                                               7,36

  [CONCENTRACION1]_12EM2Cristalizador2Corriente_AI.Valor  \
0                                                NaN
1                                                NaN
2                                               6,01
3                                               6,01
```

```
4                                                6,01

   [CONCENTRACION1]_12EM3Cristalizador3Corriente_AI.Valor  \
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4                                                 NaN

   [CONCENTRACION1]_22EM1Cristalizador4Corriente_AI.Valor  \
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4                                                 NaN

   [CONCENTRACION1]_22EM2Cristalizador5Corriente_AI.Valor  \
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4                                                 NaN

   [CONCENTRACION1]_22EM3Cristalizador6Corriente_AI.Valor  \
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4                                                 NaN

   [CONCENTRACION1]_32EM1Cristalizador7Corriente_AI.Valor  \
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4                                                 NaN

   [CONCENTRACION1]_32EM2Cristalizador8Corriente_AI.Valor  \
0                                                 NaN
1                                                 NaN
2                                                7,74
3                                                7,74
4                                                7,74

   [CONCENTRACION1]_13V01Colum1Conduct_CT1319_AI.Valor  \
0                                                 NaN
1                                                 NaN
```

```
2                                                   NaN
3                                                   NaN
4                                                   NaN

  [CONCENTRACION1]_23V01Colum2Conduct_CT2319_AI.Valor  \
0                                             61,81
1                                             61,81
2                                             61,81
3                                             61,81
4                                            308,21

  [CONCENTRACION1]_33V01Colum3Conduct_CT3319_AI.Valor
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                              515,45
4                                              515,45
```

[4]:
```
eficiencia = pd.read_csv('data/Eficiencia.csv', delimiter=';')
eficiencia.head(5)
```

[4]:

	Dia	Hora	porc_hielo1	porc_hielo2	porc_hielo3	tiempo_ciclo	Desague
0	1/03/2022	22:00	34,9	30,7	37,3	42	1.018
1	1/03/2022	23:00	NaN	NaN	NaN	NaN	1.019
2	1/03/2022	0:00	34,6	30,6	37,1	42	982
3	1/03/2022	1:00	NaN	NaN	NaN	NaN	989
4	1/03/2022	2:00	33,3	31,2	36,6	40	1.188

2.0.1 Convert data types to the corresponding type

Change column names Due to the long and confusing column names in the databases that were identified in the first analysis, the names are changed to more precise identifiers to facilitate their handling.

[5]:
```
recristalizador.columns =['Time', 'Corriente_recristalizador1',␣
 ↪'Corriente_recristalizador2', 'Corriente_recristalizador3',␣
 ↪'Temperatura_recristalizador1', 'Temperatura_recristalizador2',␣
 ↪'Temperatura_recristalizador3','Temperatura_recibidor1',␣
 ↪'Temperatura_recibidor2', 'Temperatura_recibidor3',␣
 ↪'Temperatura_Tap','Presión_recristalizador1', 'Presión_recristalizador2',␣
 ↪'Presión_recristalizador3']
recristalizador.columns
```

[5]:
```
Index(['Time', 'Corriente_recristalizador1', 'Corriente_recristalizador2',
       'Corriente_recristalizador3', 'Temperatura_recristalizador1',
       'Temperatura_recristalizador2', 'Temperatura_recristalizador3',
       'Temperatura_recibidor1', 'Temperatura_recibidor2',
```

```
       'Temperatura_recibidor3', 'Temperatura_Tap', 'Presión_recristalizador1',
       'Presión_recristalizador2', 'Presión_recristalizador3'],
      dtype='object')
```

[6]:
```
cristalizador.columns = ['Time', 'Presión_recibidor1', 'Presión_recibidor2',
                'Presión_recibidor3', 'Corriente_cristalizador1',
                'Corriente_cristalizador2', 'Corriente_cristalizador3',
                'Corriente_cristalizador4', 'Corriente_cristalizador5',
                'Corriente_cristalizador6', 'Corriente_cristalizador7',
                'Corriente_cristalizador8', 'Conductividad_columna1',
                'Conductividad_columna2', 'Conductividad_columna3']
cristalizador.columns
```

[6]:
```
Index(['Time', 'Presión_recibidor1', 'Presión_recibidor2',
       'Presión_recibidor3', 'Corriente_cristalizador1',
       'Corriente_cristalizador2', 'Corriente_cristalizador3',
       'Corriente_cristalizador4', 'Corriente_cristalizador5',
       'Corriente_cristalizador6', 'Corriente_cristalizador7',
       'Corriente_cristalizador8', 'Conductividad_columna1',
       'Conductividad_columna2', 'Conductividad_columna3'],
      dtype='object')
```

Change the variable type After adjusting the datasets db1 and db2 we proceed to identify the variables and their data type, from this it is evident that they are stored as objects which are composed of numerical and dates.

All objects are cast to be handled as number and date type in order to clean the data and obtain a better analysis of the information.

[7]:
```
print(recristalizador.info())
print(cristalizador.info())
print(eficiencia.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41187 entries, 0 to 41186
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Time                          41187 non-null  object
 1   Corriente_recristalizador1    41186 non-null  object
 2   Corriente_recristalizador2    41182 non-null  object
 3   Corriente_recristalizador3    41185 non-null  object
 4   Temperatura_recristalizador1  41173 non-null  object
 5   Temperatura_recristalizador2  41179 non-null  object
 6   Temperatura_recristalizador3  41178 non-null  object
 7   Temperatura_recibidor1        41172 non-null  object
 8   Temperatura_recibidor2        41183 non-null  object
 9   Temperatura_recibidor3        41184 non-null  object
```

```
 10  Temperatura_Tap               41175 non-null  object
 11  Presión_recristalizador1      41176 non-null  object
 12  Presión_recristalizador2      41169 non-null  object
 13  Presión_recristalizador3      41187 non-null  object
dtypes: object(14)
memory usage: 4.4+ MB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43395 entries, 0 to 43394
Data columns (total 15 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Time                       43395 non-null  object
 1   Presión_recibidor1         43372 non-null  object
 2   Presión_recibidor2         43370 non-null  object
 3   Presión_recibidor3         43390 non-null  object
 4   Corriente_cristalizador1   43394 non-null  object
 5   Corriente_cristalizador2   43393 non-null  object
 6   Corriente_cristalizador3   43374 non-null  object
 7   Corriente_cristalizador4   43380 non-null  object
 8   Corriente_cristalizador5   43378 non-null  object
 9   Corriente_cristalizador6   43382 non-null  object
 10  Corriente_cristalizador7   43388 non-null  object
 11  Corriente_cristalizador8   43393 non-null  object
 12  Conductividad_columna1     43385 non-null  object
 13  Conductividad_columna2     43395 non-null  object
 14  Conductividad_columna3     43392 non-null  object
dtypes: object(15)
memory usage: 5.0+ MB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2640 entries, 0 to 2639
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Dia            2640 non-null   object
 1   Hora           2640 non-null   object
 2   porc_hielo1    1066 non-null   object
 3   porc_hielo2    1152 non-null   object
 4   porc_hielo3    1141 non-null   object
 5   tiempo_ciclo   1058 non-null   object
 6   Desague        2130 non-null   object
dtypes: object(7)
memory usage: 144.5+ KB
None
```

Variables Date / doubles In the records of the time column of recristalizador, there is an additional hour after the initial date. Therefore, it was decided to separate this information into two variables, in order to identify with the BuenCafe team if the second date has any relevance.

[8]:
```
recristalizador.columns[1:14]
for i in recristalizador.columns[1:14]:
  recristalizador[i]=recristalizador[i].astype("string").str.replace(",", ".").
 ↪astype(float)

recristalizador['Time2'] = recristalizador['Time'].astype("string").str.
 ↪split(',',expand=True)[0]
recristalizador['Time2'] = pd.to_datetime(recristalizador['Time2'])

recristalizador.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41187 entries, 0 to 41186
Data columns (total 15 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Time                           41187 non-null  object
 1   Corriente_recristalizador1     41186 non-null  float64
 2   Corriente_recristalizador2     41182 non-null  float64
 3   Corriente_recristalizador3     41185 non-null  float64
 4   Temperatura_recristalizador1   41173 non-null  float64
 5   Temperatura_recristalizador2   41179 non-null  float64
 6   Temperatura_recristalizador3   41178 non-null  float64
 7   Temperatura_recibidor1         41172 non-null  float64
 8   Temperatura_recibidor2         41183 non-null  float64
 9   Temperatura_recibidor3         41184 non-null  float64
 10  Temperatura_Tap                41175 non-null  float64
 11  Presión_recristalizador1       41176 non-null  float64
 12  Presión_recristalizador2       41169 non-null  float64
 13  Presión_recristalizador3       41187 non-null  float64
 14  Time2                          41187 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(13), object(1)
memory usage: 4.7+ MB
```

Variables Date / doubles The same adjustment is made to the dates for cristalizador.

[9]:
```
cristalizador.columns[1:]
for i in cristalizador.columns[1:]:
  cristalizador[i]=cristalizador[i].astype("string").str.replace(",", ".").
 ↪astype(float)

cristalizador['Time2'] = cristalizador['Time'].astype("string").str.
 ↪split(',',expand=True)[0]
```

```
cristalizador['Time2'] = pd.to_datetime(cristalizador['Time2'])

cristalizador.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43395 entries, 0 to 43394
Data columns (total 16 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Time                      43395 non-null  object
 1   Presión_recibidor1        43372 non-null  float64
 2   Presión_recibidor2        43370 non-null  float64
 3   Presión_recibidor3        43390 non-null  float64
 4   Corriente_cristalizador1  43394 non-null  float64
 5   Corriente_cristalizador2  43393 non-null  float64
 6   Corriente_cristalizador3  43374 non-null  float64
 7   Corriente_cristalizador4  43380 non-null  float64
 8   Corriente_cristalizador5  43378 non-null  float64
 9   Corriente_cristalizador6  43382 non-null  float64
 10  Corriente_cristalizador7  43388 non-null  float64
 11  Corriente_cristalizador8  43393 non-null  float64
 12  Conductividad_columna1    43385 non-null  float64
 13  Conductividad_columna2    43395 non-null  float64
 14  Conductividad_columna3    43392 non-null  float64
 15  Time2                     43395 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(14), object(1)
memory usage: 5.3+ MB
```

[10]:
```
eficiencia.columns[2:6]
for i in eficiencia.columns[2:6]:
  eficiencia[i]=eficiencia[i].astype("string").str.replace(",", ".").
 ↪astype(float)

eficiencia['Desague']= eficiencia['Desague'].astype("string").str.replace(".",␣
 ↪"").str.replace(",", ".").astype(float)

eficiencia.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2640 entries, 0 to 2639
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Dia           2640 non-null   object
 1   Hora          2640 non-null   object
 2   porc_hielo1   1066 non-null   float64
 3   porc_hielo2   1152 non-null   float64
```

```
 4   porc_hielo3   1141 non-null   float64
 5   tiempo_ciclo  1058 non-null   float64
 6   Desague       2130 non-null   float64
dtypes: float64(5), object(2)
memory usage: 144.5+ KB
```

```
<ipython-input-10-beb43a508816>:5: FutureWarning: The default value of regex
will change from True to False in a future version. In addition, single
character regular expressions will*not* be treated as literal strings when
regex=True.
  eficiencia['Desague']= eficiencia['Desague'].astype("string").str.replace(".",
"").str.replace(",", ".").astype(float)
```

2.1 Database merge

[11]:
```
eficiencia['time']=pd.to_datetime(eficiencia['Dia'].astype('string') + ' ' +␣
 ↪eficiencia['Hora'].astype('string'))
recristalizador['time']=pd.to_datetime(recristalizador['Time2'].dt.
 ↪strftime('%Y-%m-%d %H:00:00'))
cristalizador['time']=pd.to_datetime(cristalizador['Time2'].dt.
 ↪strftime('%Y-%m-%d %H:00:00'))
datos = recristalizador.groupby(by= 'time').mean().merge(eficiencia,␣
 ↪how='left', left_index= True, right_on= 'time')
datos = cristalizador.groupby(by= 'time').mean().merge(datos, how='left',␣
 ↪left_index= True, right_on= 'time')
datos.head(5)
```

[11]:

	Presión_recibidor1	Presión_recibidor2	Presión_recibidor3	\
3.0	2.040278	-3.21	0.702500	
4.0	2.092667	-3.21	0.666000	
5.0	1.989811	-3.21	0.737170	
6.0	2.044211	-3.21	0.677632	
7.0	2.069024	-3.21	0.767073	

	Corriente_cristalizador1	Corriente_cristalizador2	\
3.0	7.883333	8.438333	
4.0	7.841778	6.510222	
5.0	8.803774	6.082830	
6.0	8.785263	6.192632	
7.0	7.999268	6.094390	

	Corriente_cristalizador3	Corriente_cristalizador4	\
3.0	9.905833	6.451667	
4.0	10.362889	6.540889	
5.0	6.677925	6.356226	
6.0	7.075526	6.342632	
7.0	6.375610	6.405610	

```
     Corriente_cristalizador5  Corriente_cristalizador6  \
3.0                  6.715000                  8.527222
4.0                  6.705778                  8.224889
5.0                  6.823208                  8.303774
6.0                  6.660526                  8.022632
7.0                  6.553659                  8.406829

     Corriente_cristalizador7  …  Presión_recristalizador2  \
3.0                  7.832500  …                  2.511250
4.0                  7.899556  …                  2.393409
5.0                  7.656792  …                  2.333617
6.0                  7.589474  …                  2.346857
7.0                  7.295366  …                  2.492333

     Presión_recristalizador3        Dia  Hora  porc_hielo1  porc_hielo2  \
3.0                  1.748125  1/03/2022  1:00          NaN          NaN
4.0                  1.720227  1/03/2022  2:00         33.3         31.2
5.0                  1.891064  1/03/2022  3:00          NaN          NaN
6.0                  1.799429  1/03/2022  4:00          NaN         30.7
7.0                  1.637667  1/03/2022  5:00          NaN          NaN

     porc_hielo3  tiempo_ciclo  Desague                time
3.0          NaN           NaN    989.0 2022-01-03 01:00:00
4.0         36.6          40.0   1188.0 2022-01-03 02:00:00
5.0          NaN           NaN    874.0 2022-01-03 03:00:00
6.0         36.4           NaN   1088.0 2022-01-03 04:00:00
7.0          NaN           NaN    886.0 2022-01-03 05:00:00

[5 rows x 35 columns]
```

2.1.1 Null values

Null values are analyzed to decide whether or not to remove them

[12]:
```
datos1 = datos.copy()
datos1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Float64Index: 1066 entries, 3.0 to 2342.0
Data columns (total 35 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Presión_recibidor1            1066 non-null   float64
 1   Presión_recibidor2            1066 non-null   float64
 2   Presión_recibidor3            1066 non-null   float64
 3   Corriente_cristalizador1      1066 non-null   float64
 4   Corriente_cristalizador2      1066 non-null   float64
```

```
 5   Corriente_cristalizador3      1066 non-null   float64
 6   Corriente_cristalizador4      1066 non-null   float64
 7   Corriente_cristalizador5      1066 non-null   float64
 8   Corriente_cristalizador6      1066 non-null   float64
 9   Corriente_cristalizador7      1066 non-null   float64
 10  Corriente_cristalizador8      1066 non-null   float64
 11  Conductividad_columna1        1066 non-null   float64
 12  Conductividad_columna2        1066 non-null   float64
 13  Conductividad_columna3        1066 non-null   float64
 14  Corriente_recristalizador1    1066 non-null   float64
 15  Corriente_recristalizador2    1066 non-null   float64
 16  Corriente_recristalizador3    1066 non-null   float64
 17  Temperatura_recristalizador1  1066 non-null   float64
 18  Temperatura_recristalizador2  1066 non-null   float64
 19  Temperatura_recristalizador3  1066 non-null   float64
 20  Temperatura_recibidor1        1066 non-null   float64
 21  Temperatura_recibidor2        1066 non-null   float64
 22  Temperatura_recibidor3        1066 non-null   float64
 23  Temperatura_Tap               1066 non-null   float64
 24  Presión_recristalizador1      1066 non-null   float64
 25  Presión_recristalizador2      1066 non-null   float64
 26  Presión_recristalizador3      1066 non-null   float64
 27  Dia                           1030 non-null   object
 28  Hora                          1030 non-null   object
 29  porc_hielo1                   380 non-null    float64
 30  porc_hielo2                   448 non-null    float64
 31  porc_hielo3                   445 non-null    float64
 32  tiempo_ciclo                  377 non-null    float64
 33  Desague                       831 non-null    float64
 34  time                          1066 non-null   datetime64[ns]
dtypes: datetime64[ns](1), float64(32), object(2)
memory usage: 299.8+ KB
```

After processing the null values of each column a recristalizador and cristalizador dataset, you can see that the average value is zero.

[13]:
```
print(datos1.isnull().mean())
```

```
Presión_recibidor1              0.000000
Presión_recibidor2              0.000000
Presión_recibidor3              0.000000
Corriente_cristalizador1        0.000000
Corriente_cristalizador2        0.000000
Corriente_cristalizador3        0.000000
Corriente_cristalizador4        0.000000
Corriente_cristalizador5        0.000000
Corriente_cristalizador6        0.000000
Corriente_cristalizador7        0.000000
```

```
Corriente_cristalizador8        0.000000
Conductividad_columna1          0.000000
Conductividad_columna2          0.000000
Conductividad_columna3          0.000000
Corriente_recristalizador1      0.000000
Corriente_recristalizador2      0.000000
Corriente_recristalizador3      0.000000
Temperatura_recristalizador1    0.000000
Temperatura_recristalizador2    0.000000
Temperatura_recristalizador3    0.000000
Temperatura_recibidor1          0.000000
Temperatura_recibidor2          0.000000
Temperatura_recibidor3          0.000000
Temperatura_Tap                 0.000000
Presión_recristalizador1        0.000000
Presión_recristalizador2        0.000000
Presión_recristalizador3        0.000000
Dia                             0.033771
Hora                            0.033771
porc_hielo1                     0.643527
porc_hielo2                     0.579737
porc_hielo3                     0.582552
tiempo_ciclo                    0.646341
Desague                         0.220450
time                            0.000000
dtype: float64
```

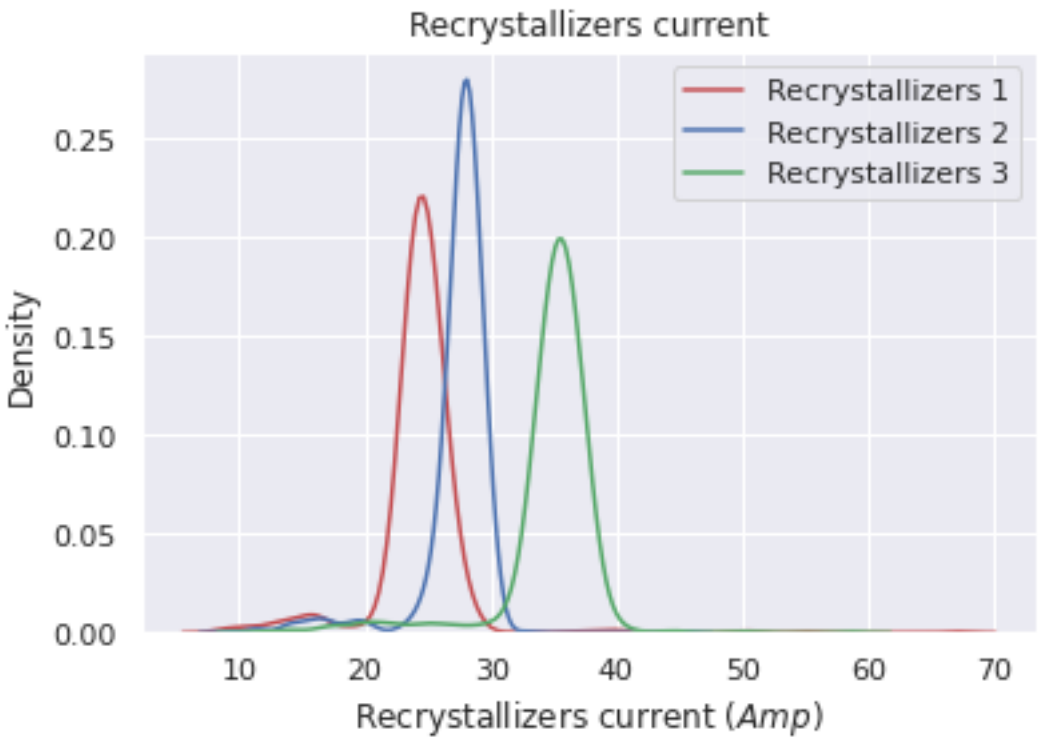
The variable Drain (kg/h) is taken as an indicator of efficiency because it has a directly proportional relationship to the amount of product in kg that is extracted. Therefore, if a value of 'Drain' is not known in any row, the corresponding hourly data cannot be taken into account during the analysis.

[14]:
```
datos1 = datos1.dropna(subset=['Desague'])
```

2.2 Recristalizador graphs

Taking the previous information, it is necessary to plot dataset 1, in order to contrast the data. Since they are variables with elements in common, important differences in density can be appreciated, which can lead us to identify improvement alternatives as points to analyze.

[15]:
```
sns.set(style="darkgrid")
fig1 = sns.kdeplot(datos1.Corriente_recristalizador1, color="r").
 ↪set_title("Recrystallizers current")
fig2 = sns.kdeplot(datos1.Corriente_recristalizador2, color="b")
fig3 = sns.kdeplot(datos1.Corriente_recristalizador3, color="g")
plt.legend(labels = ["Recrystallizers 1", "Recrystallizers 2", "Recrystallizers␣
 ↪3"])
plt.xlabel("Recrystallizers current ($Amp$)")
plt.show()
```

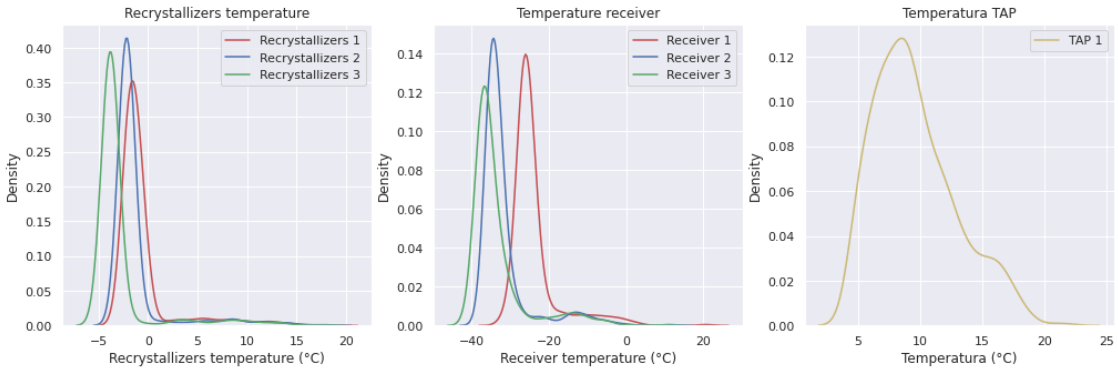


2.2.1 Graphical Analysis

Although most of the data are between 20 and 45 values corresponding to the correinte_recristalizador1, there are also high values in density with a current equal to 0.

[16]:
```
plt.figure(figsize=(17,5))
plt.subplot(1,3,1)
sns.set(style="darkgrid")
fig = sns.kdeplot(datos1.Temperatura_recristalizador1, color="r").
 ↪set_title("Recrystallizers temperature")
fig = sns.kdeplot(datos1.Temperatura_recristalizador2, color="b")
fig = sns.kdeplot(datos1.Temperatura_recristalizador3, color="g")
plt.legend(labels = ["Recrystallizers 1", "Recrystallizers 2", "Recrystallizers␣
 ↪3"])
plt.xlabel("Recrystallizers temperature (°C)")
plt.subplot(1,3,2)
sns.set(style="darkgrid")
fig = sns.kdeplot(datos1.Temperatura_recibidor1, color="r").
 ↪set_title("Temperature receiver")
fig = sns.kdeplot(datos1.Temperatura_recibidor2, color="b")
fig = sns.kdeplot(datos1.Temperatura_recibidor3, color="g")
plt.legend(labels = ["Receiver 1", "Receiver 2", "Receiver 3"])
plt.xlabel("Receiver temperature (°C)")
```

```
plt.subplot(1,3,3)
sns.set(style="darkgrid")
fig = sns.kdeplot(datos1.Temperatura_Tap, color="y").set_title("Temperatura␣
↪TAP")
plt.legend(labels = ["TAP 1"])
plt.xlabel("Temperatura (°C)");
```



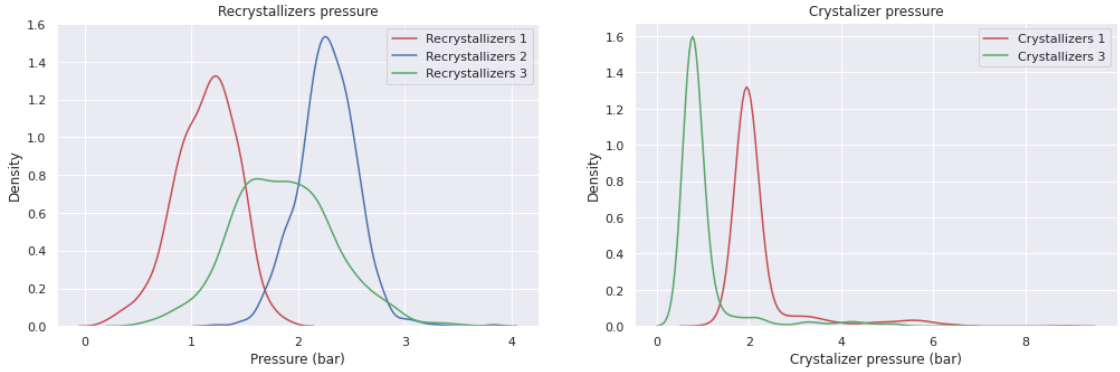
2.2.2 Pressure Analysis

Most of the temperatures identified with the records of dataset1 are negative values. dataset1 records are negative values, this helps us to reaffirm that during the process the ice crystals are maintained in order to separate the coffee extract. The ice crystals are maintained in order to separate the coffee extract leaving the water in the form of ice for later extraction.

[17]:
```
plt.figure(figsize=(17,5))
plt.subplot(1,2,1)
sns.set(style="darkgrid")
fig = sns.kdeplot(datos1.Presión_recristalizador1, color="r").
↪set_title("Recrystallizers pressure")
fig = sns.kdeplot(datos1.Presión_recristalizador2, color="b")
fig = sns.kdeplot(datos1.Presión_recristalizador3, color="g")
plt.legend(labels = ["Recrystallizers 1", "Recrystallizers 2", "Recrystallizers␣
↪3"])
plt.xlabel("Pressure (bar)")

plt.subplot(1,2,2)
sns.set(style="darkgrid")
fig = sns.kdeplot(datos1.Presión_recibidor1, color="r").set_title("Crystalizer␣
↪pressure")
#fig = sns.kdeplot(datos1.Presión_recibidor2, color="b")
fig = sns.kdeplot(datos1.Presión_recibidor3, color="g")
plt.xlabel("Crystalizer pressure (bar)")
plt.legend(labels = ["Crystallizers 1", "Crystallizers 3"])
```

```
;
```

[17]:
```
''
```



[18]:
```
datos1.Presión_recibidor2.unique()
```

[18]:
```
array([-3.21, -3.21, -3.21, -3.21, -3.21, -3.21, -3.21, -3.21, -3.21,
       -3.21])
```

A critical point that is evident in the graph is at a value of 2 for the recrystallizer pressure axis and on the y-axis at a value of 0.6.

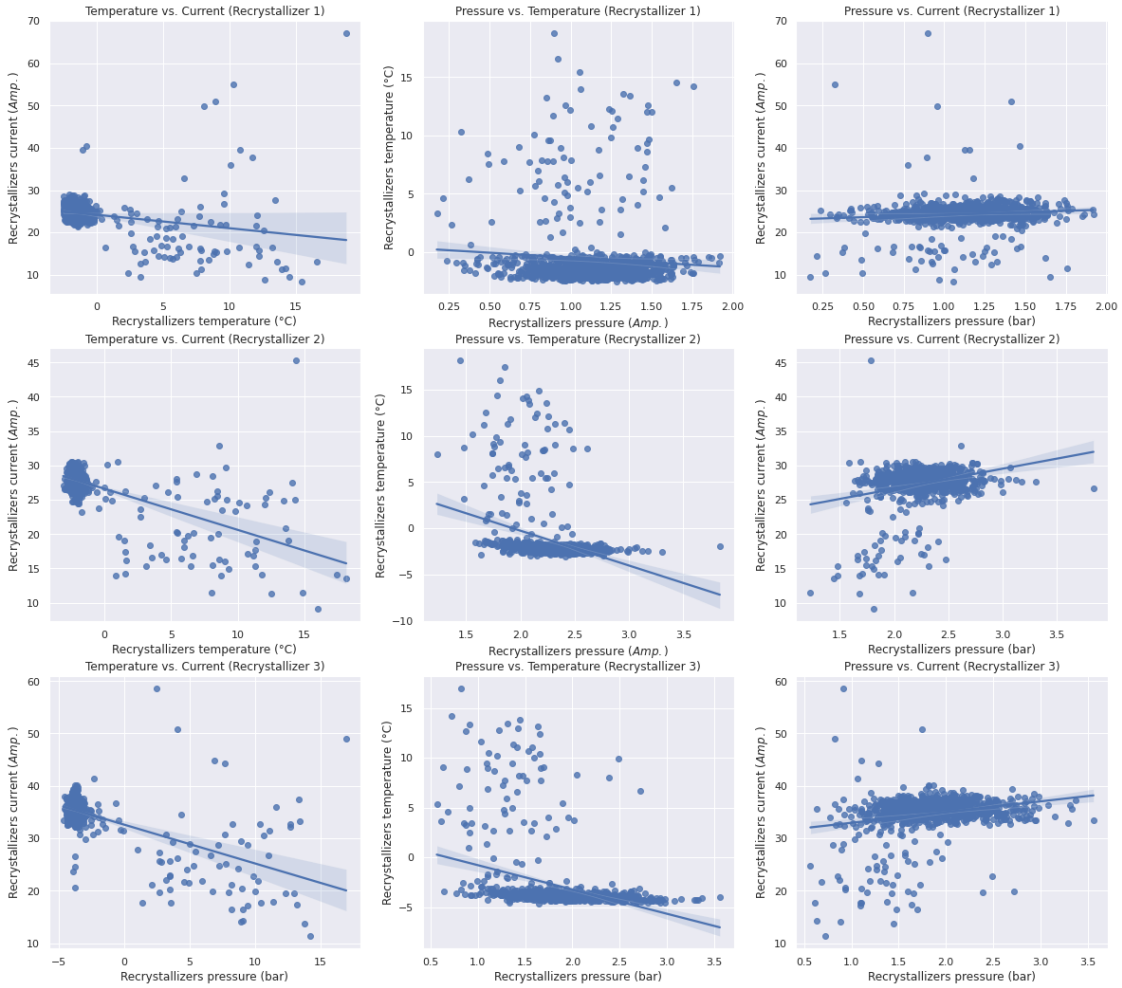
the highest density expressed in the graph is above the value of 2.

most of the records of recrystallizer 1 are in the lower limit of 2, while for recrystallizer 2 they are in the upper limit.

[19]:
```
plt.figure(figsize=(20,18))
plt.subplot(3,3,1)
sns.regplot(x=datos1["Temperatura_recristalizador1"],␣
 ↪y=datos1["Corriente_recristalizador1"]).set_title("Temperature vs. Current␣
 ↪(Recrystallizer 1)")
plt.xlabel("Recrystallizers temperature (°C)")
plt.ylabel("Recrystallizers current ($Amp.$)")

plt.subplot(3,3,2)
sns.regplot(x=datos1["Presión_recristalizador1"],␣
 ↪y=datos1["Temperatura_recristalizador1"]).set_title("Pressure vs.␣
 ↪Temperature (Recrystallizer 1)")
plt.ylabel("Recrystallizers temperature (°C)")
plt.xlabel("Recrystallizers pressure ($Amp.$)")

plt.subplot(3,3,3)
```

```
sns.regplot(x=datos1["Presión_recristalizador1"],␣
 ↪y=datos1["Corriente_recristalizador1"]).set_title("Pressure vs. Current␣
 ↪(Recrystallizer 1)")
plt.xlabel("Recrystallizers pressure (bar)")
plt.ylabel("Recrystallizers current ($Amp.$)")

plt.subplot(3,3,4)
sns.regplot(x=datos1["Temperatura_recristalizador2"],␣
 ↪y=datos1["Corriente_recristalizador2"]).set_title("Temperature vs. Current␣
 ↪(Recrystallizer 2)")
plt.xlabel("Recrystallizers temperature (°C)")
plt.ylabel("Recrystallizers current ($Amp.$)")


plt.subplot(3,3,5)
sns.regplot(x=datos1["Presión_recristalizador2"],␣
 ↪y=datos1["Temperatura_recristalizador2"]).set_title("Pressure vs.␣
 ↪Temperature (Recrystallizer 2)")
plt.ylabel("Recrystallizers temperature (°C)")
plt.xlabel("Recrystallizers pressure ($Amp.$)")

plt.subplot(3,3,6)
sns.regplot(x=datos1["Presión_recristalizador2"],␣
 ↪y=datos1["Corriente_recristalizador2"]).set_title("Pressure vs. Current␣
 ↪(Recrystallizer 2)")
plt.xlabel("Recrystallizers pressure (bar)")
plt.ylabel("Recrystallizers current ($Amp.$)")

plt.subplot(3,3,7)
sns.regplot(x=datos1["Temperatura_recristalizador3"],␣
 ↪y=datos1["Corriente_recristalizador3"]).set_title("Temperature vs. Current␣
 ↪(Recrystallizer 3)")
plt.xlabel("Recrystallizers pressure (bar)")
plt.ylabel("Recrystallizers current ($Amp.$)")

plt.subplot(3,3,8)
sns.regplot(x=datos1["Presión_recristalizador3"],␣
 ↪y=datos1["Temperatura_recristalizador3"]).set_title("Pressure vs.␣
 ↪Temperature (Recrystallizer 3)")
plt.ylabel("Recrystallizers temperature (°C)")
plt.xlabel("Recrystallizers pressure (bar)")

plt.subplot(3,3,9)
sns.regplot(x=datos1["Presión_recristalizador3"],␣
 ↪y=datos1["Corriente_recristalizador3"]).set_title("Pressure vs. Current␣
 ↪(Recrystallizer 3)")
```

```
plt.xlabel("Recrystallizers pressure (bar)")
plt.ylabel("Recrystallizers current ($Amp.$)");
```



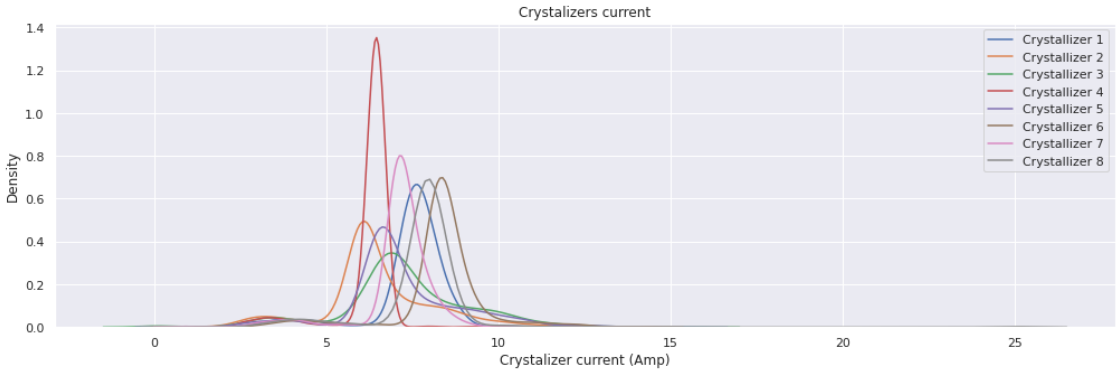
More linear values are evident in recrystallizer 3 as pressure increases and temperature decreases to values less than zero.

2.3 Cristalizador Graphs

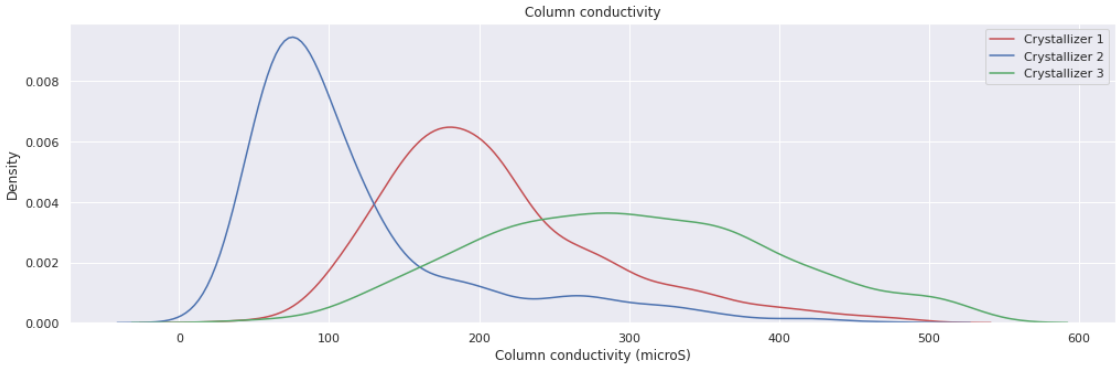
Since they are variables with elements in common, important differences in density can be appreciated, which can lead us to identify improvement alternatives as points to analyze.

[20]:
```
plt.figure(figsize=(17,5))
sns.set(style="darkgrid")
fig = sns.kdeplot(datos1.Corriente_cristalizador1).set_title("Crystalizers␣
 ↪current")
fig = sns.kdeplot(datos1.Corriente_cristalizador2)
fig = sns.kdeplot(datos1.Corriente_cristalizador3)
```

```
fig = sns.kdeplot(datos1.Corriente_cristalizador4)
fig = sns.kdeplot(datos1.Corriente_cristalizador5)
fig = sns.kdeplot(datos1.Corriente_cristalizador6)
fig = sns.kdeplot(datos1.Corriente_cristalizador7)
fig = sns.kdeplot(datos1.Corriente_cristalizador8)
current_palette = sns.color_palette()
plt.xlabel("Crystalizer current (Amp)")
plt.legend(labels = ["Crystallizer 1", "Crystallizer 2","Crystallizer 3",␣
 ↪"Crystallizer 4", "Crystallizer 5", "Crystallizer 6", "Crystallizer 7",␣
 ↪"Crystallizer 8"])
#sns.palplot(current_palette)
plt.show(current_palette)
```



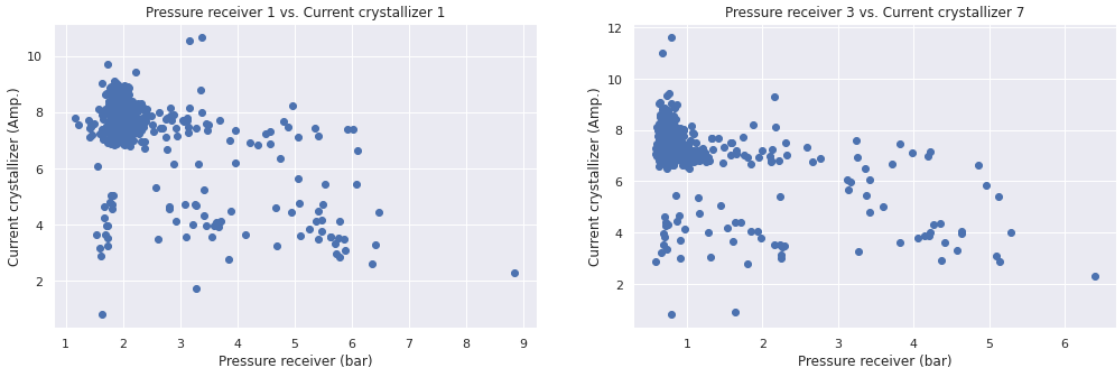
[21]:
```
plt.figure(figsize=(17,5))
sns.set(style="darkgrid")
fig = sns.kdeplot(datos1.Conductividad_columna1, color="r").set_title("Column␣
 ↪conductivity")
fig = sns.kdeplot(datos1.Conductividad_columna2, color="b")
fig = sns.kdeplot(datos1.Conductividad_columna3, color="g")
plt.xlabel("Column conductivity (microS)")
plt.legend(labels = ["Crystallizer 1", "Crystallizer 2","Crystallizer 3"])
plt.show()
```



2.4 Pressure / Stream graph analysis

[22]:
```
plt.figure(figsize=(17,5))

plt.subplot(1,2,1)
plt.scatter(datos1.Presión_recibidor1, datos1.Corriente_cristalizador1 )
plt.title("Pressure receiver 1 vs. Current crystallizer 1")
plt.xlabel("Pressure receiver (bar)")
plt.ylabel("Current crystallizer (Amp.)")

plt.subplot(1,2,2)
plt.scatter(datos1.Presión_recibidor3, datos1.Corriente_cristalizador7 )
plt.title("Pressure receiver 3 vs. Current crystallizer 7")
plt.xlabel("Pressure receiver (bar)")
plt.ylabel("Current crystallizer (Amp.)")
```

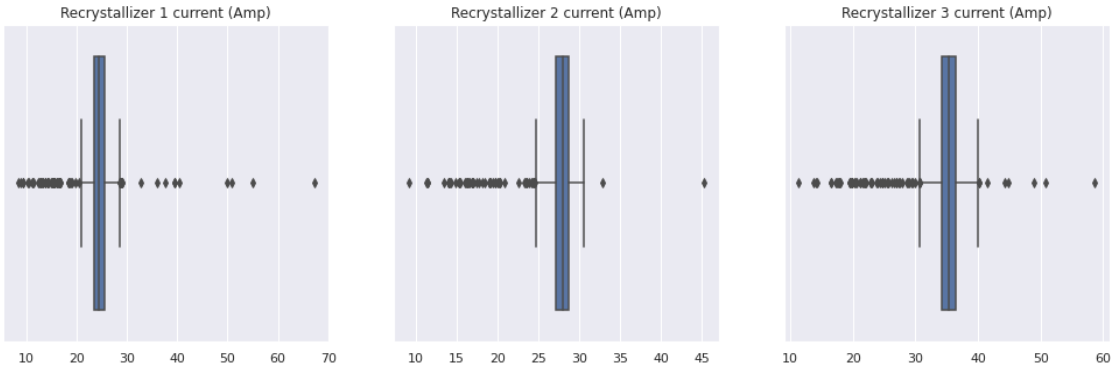
[22]:
```
Text(0, 0.5, 'Current crystallizer (Amp.)')
```



2.5 Graficas boxplot

[23]:
```
plt.figure(figsize=(17,5))

plt.subplot(1,3,1)
sns.boxplot(x=datos1.Corriente_recristalizador1)
plt.title("Recrystallizer 1 current (Amp)")
plt.xlabel("")

plt.subplot(1,3,2)
sns.boxplot(x=datos1.Corriente_recristalizador2)
plt.title("Recrystallizer 2 current (Amp)")
```

```
plt.xlabel("")

plt.subplot(1,3,3)
sns.boxplot(x=datos1.Corriente_recristalizador3)
plt.title("Recrystallizer 3 current (Amp)")
plt.xlabel("");
```

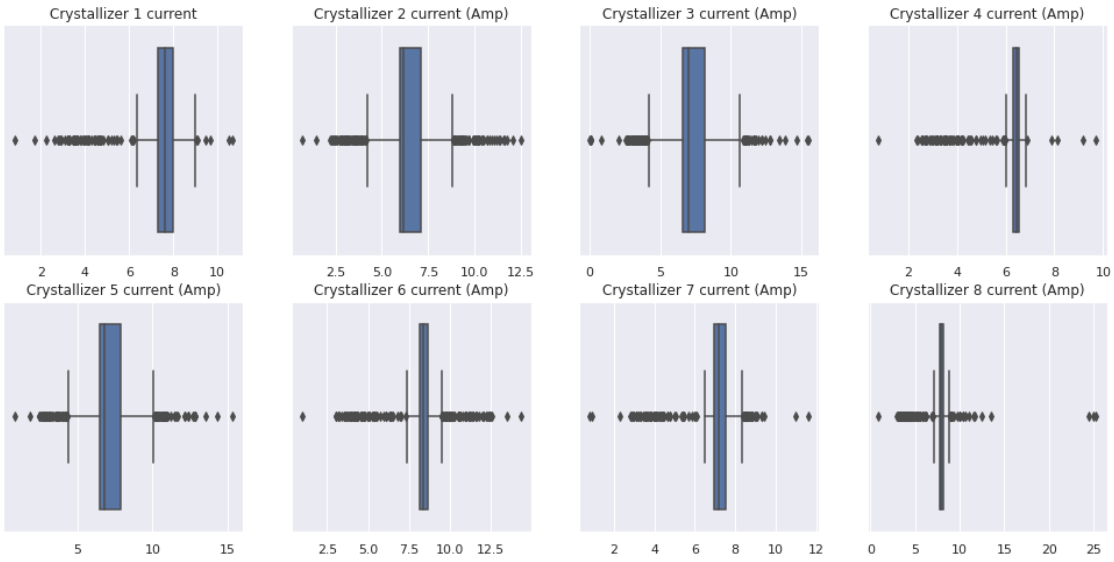


For all recrystallizers, 50% of the data are between 20Amp and 40Amp. The first recrystallizer has several outliers, unlike 2 and 3. The current range of recrystallizer 3 varies with respect to 1 and 2 (30Amp - 40Amp), why is this? Does the latter consume more power?

[24]:
```
plt.figure(figsize=(17,8))

plt.subplot(2,4,1)
sns.boxplot(x=datos1.Corriente_cristalizador1)
plt.title("Crystallizer 1 current")
plt.xlabel("")

plt.subplot(2,4,2)
sns.boxplot(x=datos1.Corriente_cristalizador2)
plt.title("Crystallizer 2 current (Amp)")
plt.xlabel("")

plt.subplot(2,4,3)
sns.boxplot(x=datos1.Corriente_cristalizador3)
plt.title("Crystallizer 3 current (Amp)")
plt.xlabel("")

plt.subplot(2,4,4)
sns.boxplot(x=datos1.Corriente_cristalizador4)
plt.title("Crystallizer 4 current (Amp)")
plt.xlabel("")

plt.subplot(2,4,5)
```

```
sns.boxplot(x=datos1.Corriente_cristalizador5)
plt.title("Crystallizer 5 current (Amp)")
plt.xlabel("")

plt.subplot(2,4,6)
sns.boxplot(x=datos1.Corriente_cristalizador6)
plt.title("Crystallizer 6 current (Amp)")
plt.xlabel("")

plt.subplot(2,4,7)
sns.boxplot(x=datos1.Corriente_cristalizador7)
plt.title("Crystallizer 7 current (Amp)")
plt.xlabel("")

plt.subplot(2,4,8)
sns.boxplot(x=datos1.Corriente_cristalizador8)
plt.title("Crystallizer 8 current (Amp)")
plt.xlabel("");
;
```

[24]: ''



For all crystallizers 50% of the data are between 5 Amp and 10 Amp. For crystallizers 2, 3 and 5 the outliers are more. The current of crystallizer 6 reaches higher values than any other crystallizer, what happened at that time? Why did it show a higher power? Crystallizers 4,5 and 7 are the ones that consume less energy.

[38]:
```
datos1.to_csv('datos1.csv', encoding='UTF-8')
```