

Como é calculado o SLA?

```
# apenas datas e horas úteis
feriados_br = holidays.Brazil()

inicio_expediente = time(8, 0)
fim_manha = time(12, 0)
inicio_tarde = time(13, 15)
fim_expediente = time(18, 0)

def dia_util(data):
    return data.weekday() < 5 and data.date() not in feriados_br

def proximo_dia_util(data):
    proxima = data + timedelta(days=1)
    while not dia_util(proxima):
        proxima += timedelta(days=1)
    return proxima.replace(hour=8, minute=0, second=0)

def ajustar_data_referencia(data_hora_solicitacao):
    if pd.isna(data_hora_solicitacao):
        return pd.NAT

    if not dia_util(data_hora_solicitacao):
        return proximo_dia_util(data_hora_solicitacao)

    horario = data_hora_solicitacao.time()

    if horario < inicio_expediente:
        return data_hora_solicitacao.replace(hour=8, minute=0, second=0)
    elif horario <= fim_manha:
        return data_hora_solicitacao
    elif horario < inicio_tarde:
        return data_hora_solicitacao.replace(hour=13, minute=15, second=0)
    elif horario <= fim_expediente:
        return data_hora_solicitacao
    else:
        return proximo_dia_util(data_hora_solicitacao)
```

No tratamento de dados, utilizamos funções que consideram somente **dias úteis** e **horário comercial**.

Como é calculado o SLA?

```
dados_cards['sla_prazo'] = np.where(
    (dados_cards['finished_at'] - dados_cards['inicio_atendimento']).dt.total_seconds() / 60 <= 480,
    1,
    np.where(
        (dados_cards['current_phase'] == 'Concluído') & (dados_cards['tempo_atendimento_minutos'] <= 480),
        1,
        0
    )
)
```

Objetivo: criar a coluna sla_prazo com 1 se o chamado cumpriu o SLA (\leq 8 horas), caso contrário 0.

Quais fases são consideradas no cálculo?

```
dados_cards['inicio_atendimento'] = dados_cards[
    [
        'first_time_enter_atendimento_tecnico_n2',
        'first_time_enter_atendimento_tecnico_n1'
    ]
].min(axis=1)

dados_cards['fim_atendimento'] = dados_cards[
    [
        'last_time_exit_atendimento_tecnico_n2',
        'last_time_exit_atendimento_tecnico_n1'
    ]
].max(axis=1)
```

Inicio de atendimento e fim de atendimento é contabilizado somente quando um card entra e sai das fases Atendimento N1 ou N2.

Faixas de atendimento

```
dict_index_faixa = {  
    'Até 1 hora': 1,  
    '1 a 4 horas': 2,  
    '4 a 8 horas': 3,  
    '8 horas a 1 dia': 4,  
    '1 a 2 dias': 5,  
    '2 a 5 dias': 6,  
    '5 a 15 dias': 7,  
    'Mais de 15 dias': 8}  
  
concat_data['index_faixa_atendimento'] = concat_data['faixa_atendimento'].map(dict_index_faixa).astype(int)
```

Quais medidas dax são usadas?

```
1 % Concluído =
2     VAR vCont = DIVIDE(
3         [qtd_concluido],
4         [contagem_tickets]
5     )
6
7     RETURN IF(vCont=BLANK(), 0, vCont)
```

```
1 % SLA Cumprido (%) =
2     VAR TotalChamados = CALCULATE(DISTINCTCOUNT(dataset_chamados_ti[id]), dataset_chamados_ti[current_phase] = "Concluído")
3     VAR CumpriuSLA = CALCULATE(DISTINCTCOUNT(dataset_chamados_ti[id]), 'dataset_chamados_ti'[sla_prazo] = 1, dataset_chamados_ti[current_phase] = "Concluído")
4     VAR vCont = DIVIDE(CumpriuSLA, TotalChamados)
5
6     RETURN if(vCont=BLANK(), 0, vCont)
```

Quais medidas dax são usadas?

```
1 mediana_atendimento_horas =
2     VAR vCont = CALCULATE(
3         CEILING(
4             MEDIANX(
5                 dataset_chamados_ti,
6                 dataset_chamados_ti[tempo_atendimento_minutos] / 60
7             ),
8             1
9         ),
10        dataset_chamados_ti[current_phase] = "Concluído",
11        dataset_protheus[tempo_atendimento_minutos] > 0
12    )
13
14    RETURN if(vCont=BLANK(), 0, vCont)
```

```
1 %_concluido_faixa =
2     VAR vQtdTotal = CALCULATE(COUNTA(dataset_chamados_ti[id]), dataset_chamados_ti[current_phase] = "Concluído", ALLSELECTED(dataset_chamados_ti))
3     VAR vQtdSel = CALCULATE(COUNTA(dataset_chamados_ti[id]), dataset_chamados_ti[current_phase] = "Concluído")
4     VAR vPart = DIVIDE(vQtdSel, vQtdTotal)
5
6     RETURN
7     IF(
8         vPart = BLANK(),
9         0,
10        vPart
11    )
```