CONCLUSIONES

January 28, 2024

Table of Contents

[1]:
```
# Código para Ocultar/Mostrar todo el código
# y solo dejar visibles los resultados

from IPython.display import HTML
HTML('''<script>code_show=true; function code_toggle() {if
(code_show){$('div.input').hide();}
else {$('div.input').show();}code_show = !code_show}$( document )
.ready(code_toggle);</script><form action="javascript:code_toggle()">
<input type="submit" value="Ocultar/Mostrar Código"></form>''')
```

[1]:
```
<IPython.core.display.HTML object>
```

1 Importar módulos y preparación

1.1 Importación

[2]:
```
# Importación paquetes
import numpy as np
import pandas as pd
import datetime as dt
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from statsmodels.stats.proportion import proportions_ztest#########

# Autocompletar rápido
%config IPCompleter.greedy = True

# Display 6 registros
pd.options.display.min_rows = 6

# Estilo seaborn
sns.set_style('darkgrid')

# Importar módulo propio de funciones
import mis_funciones as mf
```

[3]:
```
# Carga dataset
df = pd.read_pickle("0_datos_limpios.pickle")
```

2 Cambios - Temporalidad

2.1 Evolución histórica de los NA

¿Cómo ha mejorado la recogida de datos a lo largo del tiempo?

Mejoras en la recogida de datos

- La **tasa de datos faltantes (na) baja a lo largo del tiempo** en las variables UNIDAD_DESCRIPCIÓN y CONSEJERÍA a partir de 2019.
- Por comparación, la variable ESTADO, sin ningún na, se muestra plana a lo largo de todo el periodo.
- Finalmente, la FECHA_CONTESTACIÓN también disminuye los NA, aunque en 2022 repunta. Descontamos los últimos meses del año (4 meses) por el retardo que hay entre entradas y respuestas.

[4]:
```
# DataFrame auxiliar (mensual)
# Fecha de entrada como índice
df_e = df.set_index("fecha_entrada")

# Resample por mes
dfaux = df_e.resample('M').count()
```

[5]:
```
# Se añaden columnas con el número de NA por mes
dfaux['na_ud'] = dfaux.max(axis=1) - dfaux['unidad_descripcion']
dfaux['na_c'] = dfaux.max(axis=1) - dfaux['consejeria']
dfaux['na_fc'] = dfaux.max(axis=1) - dfaux['fecha_contestacion']
dfaux['na_e'] = dfaux.max(axis=1) - dfaux['estado']

# Se añaden columnas con el porcentaje
dfaux['na_ud_pct'] = 100 * dfaux.na_ud / dfaux.tipo
dfaux['na_c_pct'] = 100 * dfaux.na_c / dfaux.tipo
dfaux['na_fc_pct'] = 100 * dfaux.na_fc / dfaux.tipo
dfaux['na_e_pct'] = 100 * dfaux.na_e / dfaux.tipo
```

[6]:
```
# Gráfico de na a lo largo del tiempo.
f, ax = plt.subplots(4,1, figsize=(18,18))

sns.lineplot(ax=ax[0], data=dfaux, x=dfaux.index, y=dfaux.na_ud_pct,␣
 ↪color='green')
ax[0].set_title('% NA sobre total registros en UNIDAD DESCRIPCIÓN',
                fontsize=16, loc='center', fontdict=dict(weight='bold'))
ax[0].set_xlabel("", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))
ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[0].tick_params(axis='y', which='major', labelsize=15)
```

```
sns.lineplot(ax=ax[1], data=dfaux, x=dfaux.index, y=dfaux.na_c_pct, color='red')
ax[1].set_title('% NA sobre total registros en CONSEJERÍA',
                fontsize=16, loc='center', fontdict=dict(weight='bold'))
ax[1].set_xlabel("", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[1].tick_params(axis='y', which='major', labelsize=15)


sns.lineplot(ax=ax[2], data=dfaux, x=dfaux.index, y=dfaux.na_fc_pct,␣
 ↪color='blue')
ax[2].set_title('% NA sobre total registros en FECHA CONTESTACIÓN',
                fontsize=16, loc='center', fontdict=dict(weight='bold'))
ax[2].set_xlabel("", fontsize=15, fontdict=dict(weight='bold'))
ax[2].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))
ax[2].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[2].tick_params(axis='y', which='major', labelsize=15)


sns.lineplot(ax=ax[3], data=dfaux, x=dfaux.index, y=dfaux.na_e_pct,␣
 ↪color='black')
ax[3].set_title('% NA sobre total registros en ESTADO',
                fontsize=16, loc='center', fontdict=dict(weight='bold'))
ax[3].set_xlabel("", fontsize=15, fontdict=dict(weight='bold'))
ax[3].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))
ax[3].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[3].tick_params(axis='y', which='major', labelsize=15);

# plt.savefig("multimedia/evolucion_na.png")
# ax[2].axvline(x=dfaux.index[-4], color='grey', lw=2, ls='--');  # Octubre.
```

2.2 Evolución histórica de entradas y salidas

Cómo ha variado a lo largo del tiempo la entrada de volumen de trabajo y el volumen de trabajo realizado.

El **número de entradas** es una aproximación a la **carga de trabajo que entra** en la SGAC.

El **número de salidas** (terminadas) es una aproximación al volumen de **trabajo realizado**. Recordar que terminadas pueden ser:

- Con éxito (Contestada al Ciudadano).
- En otra situación (Cancelada, Enviada a Otra Administración, Rechazada por SGAC).

2.2.1 Diario

DIARIO

- Los datos son de años enteros, luego enero refleja salidas de entradas que se produjeron a finales del año anterior o incluso antes.
- Hay muchos días (1282) con 0 salidas. En cambio, los días con 0 entradas son muchos menos (88): no hay tantos "periodos de descanso" de entradas. Los días sin salidas se acumulan en su mayor parte al principio del periodo histórico.
- El volumen medio de entradas es mayor (27) que el de salidas (24).
- El volumen de salidas es más irregular que el volumen de entradas.

Sobre capacidad

1. Es **difícil observar un patrón causa-efecto entre entradas y salidas**: ¿las salidas se adaptan al volumen de entradas? Es decir, ¿los trabajadores reaccionan a la acumulación de trabajo?
2. **¿Es estable el stock medio de entradas sin responder?** –> Como se verá hay cambios en vacaciones (verano y Navidad).
3. El **volumen de entradas se comporta como un fenómeno aleatorio natural** y se estabiliza rápidamente cerca de su media. Las entradas se producen desde el día 1 de puesta en marcha del servicio, mientras que las salidas solo se producen tiempo después y la media se alcanza más tarde.
4. Los **periodos de "parada" en Navidad en las salidas**, parece apuntar en el mismo sentido de existencia de **sobre capacidad** de recursos para atender el volumen de trabajo que llega a la SGAC.

[7]:
```
# Solo datos hasta el 31 de diciembre de 2022
df_sin = df[df.fecha_contestacion < '2022-12-31']
```

[8]:
```
# Eliminación de datos con fecha_contestacion NA
df_sin = df_sin.dropna(axis=0, subset=['fecha_contestacion'])
```

[9]:
```
# Índice fecha de entrada
df_sine = df_sin.set_index('fecha_entrada').copy()
```

[10]:
```
# Índice fecha de contestación
df_sins = df_sin.set_index("fecha_contestacion").copy()
```

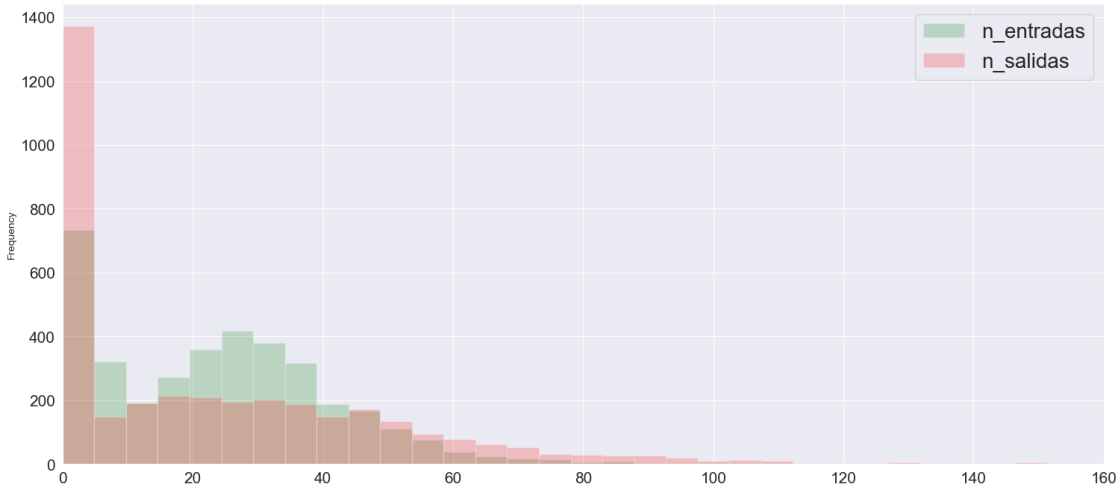
[11]:
```
# Resample de las entradas
df_sined = df_sine.resample('D').count()
```

[12]:
```
# Resample de salidas
df_sinsd = df_sins.resample('D').count()
```

[13]:
```
# Conteo de entradas, salidas y diferencia
es = pd.concat([df_sined['tipo'], df_sinsd['tipo']], axis=1).fillna(0)

es.columns = ['n_entradas', 'n_salidas']
```

```
# Esta columna es REDUCCIÓN STOCK (-) y AUMENTO STOCK (+)
es['diferencia'] = es.n_entradas - es.n_salidas
```

[14]:
```
# Gráfico número diario de Entradas y Salidas superpuesto
es[['n_entradas', 'n_salidas']].plot.hist(bins=50,
    xlim=[0, 160],
    figsize=(18,8),
    fontsize=15, alpha=0.2,
    color=['green', 'red']) \
    .legend(fontsize=20);

# plt.savefig("multimedia/entradas_y_salidas_diarias.png")
```

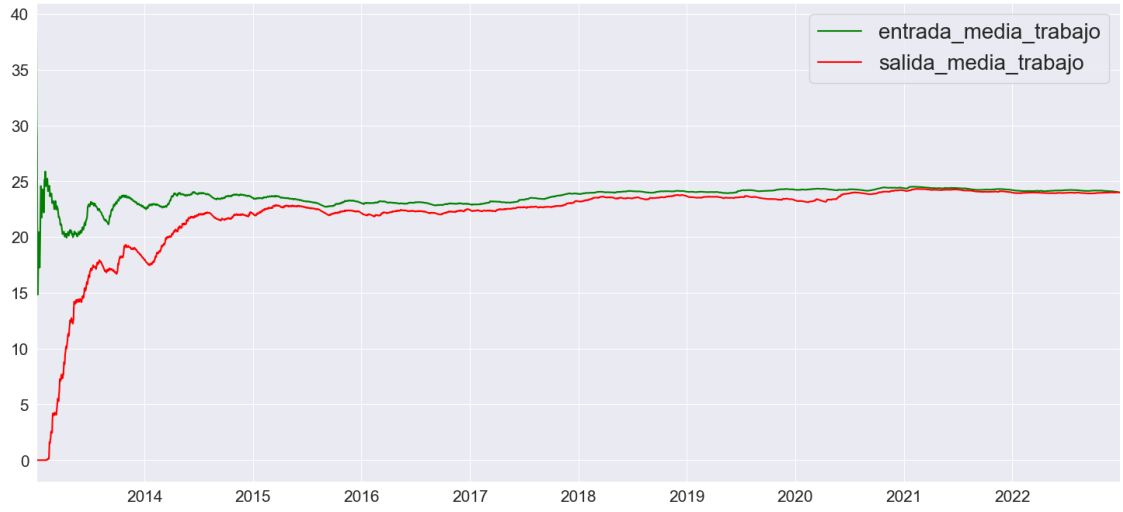


[15]:
```
# Contador de fila
es['n_fila'] = np.arange(1, len(es)+1)
```

[16]:
```
# Añadimos unas columnas para ver el volumen de entrada de trabajo a lo largo␣
  ↪del tiempo
es['entradas_acum'] = es.n_entradas.cumsum()

# Se compara con volumen de trabajo constante cada día (fila)
es['entrada_media_trabajo'] = es.entradas_acum / es.n_fila
```

[17]:
```
# Añadimos unas columnas para ver el volumen de trabajo a lo largo del tiempo
es['salidas_acum'] = es.n_salidas.cumsum()

# Se compara con volumen de trabajo constante cada día (fila)
es['salida_media_trabajo'] = es.salidas_acum / es.n_fila
```

[18]:
```
# Volumen de trabajo (entradas y salidas) a lo largo del tiempo

f, ax = plt.subplots(1,1, figsize=(18,8))

es['entrada_media_trabajo'].plot.line(figsize=(18,8), fontsize=15,␣
 ↪color='green') \
    .legend(fontsize=20)

es['salida_media_trabajo'].plot.line(figsize=(18,8), fontsize=15, color='red') \
    .legend(fontsize=20);

# plt.savefig("multimedia/e-s_diarias_media.png")
```



2.2.2 Mensual

MENSUAL

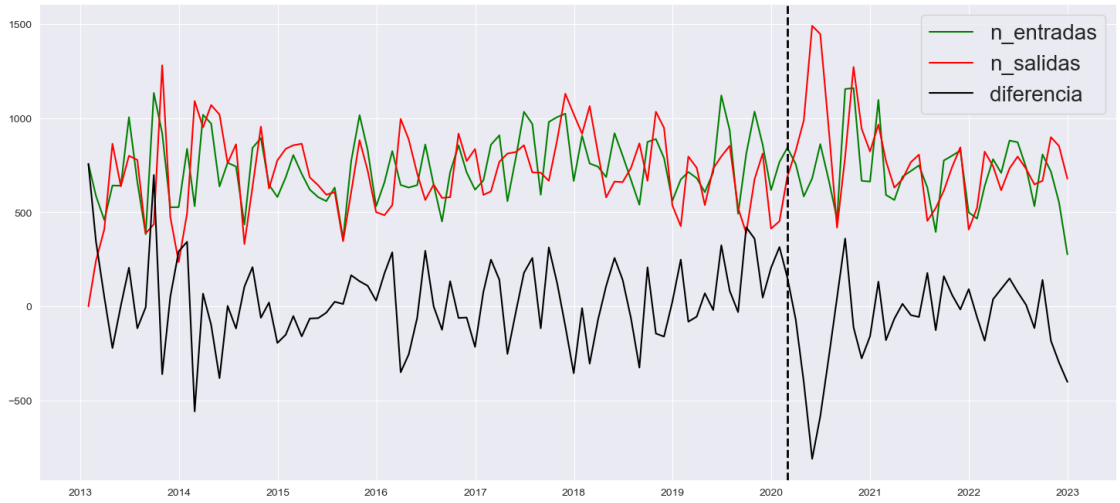
- Parece que, mensualmente, la **línea roja (salidas) se mueve con retardo tras la línea verde (entradas)**.
- El año **2020 tiene 8 meses reduciendo stock (puntos azules)** acumulado de entradas. Nuevamente, parece que hay una "reacción" de los funcionarios ante la acumulación de stock el año anterior.

[19]:
```
# Resample de las entradas
df_sinem = df_sine.resample('M').count()

# Resample de salidas
df_sinsm = df_sins.resample('M').count()
```

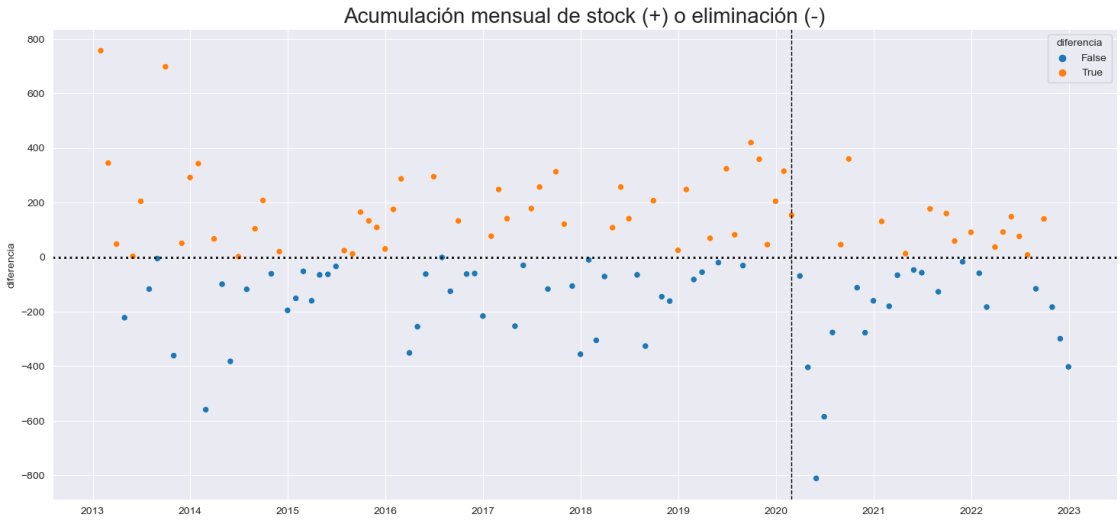
[20]:
```
# Conteo de entradas, salidas y diferencia
esm = pd.concat([df_sinem['tipo'], df_sinsm['tipo']], axis=1).fillna(0)

esm.columns = ['n_entradas', 'n_salidas']

# Esta columna es REDUCCIÓN STOCK (-) y AUMENTO STOCK (+)
esm['diferencia'] = esm.n_entradas - esm.n_salidas
```

[21]:
```
# Contador de fila
esm['n_fila'] = np.arange(1, len(esm)+1)

# Añadimos unas columnas para ver el volumen de entrada de trabajo a lo largo␣
 ↪del tiempo
esm['entradas_acum'] = esm.n_entradas.cumsum()

# Se compara con volumen de trabajo constante cada día (fila)
esm['entrada_media_trabajo'] = esm.entradas_acum / esm.n_fila

# Añadimos unas columnas para ver el volumen de trabajo a lo largo del tiempo
esm['salidas_acum'] = esm.n_salidas.cumsum()

# Se compara con volumen de trabajo constante cada día (fila)
esm['salida_media_trabajo'] = esm.salidas_acum / esm.n_fila
```

[22]:
```
pandemia = pd.to_datetime('2020-02-29')

f, ax = plt.subplots(1,1, figsize=(18,8))

sns.lineplot(ax=ax, data=esm[['n_entradas', 'n_salidas', 'diferencia']],
             palette=['green', 'red', 'black'], dashes=False)

ax.axvline(x=pandemia, color='black', lw=2, ls='--')

ax.legend(fontsize=20);

# plt.savefig("multimedia/e-s_mes.png")
```



[23]:
```
# Acumulación o eliminación de stock de entradas

r = esm.diferencia > 0  # Se acumula stock.

f, ax = plt.subplots(1,1, figsize=(18,8))

sns.scatterplot(x=esm.index, y=esm.diferencia, hue=r)

ax.axvline(x=pandemia, color='black', lw=1, ls='--')

plt.title("Acumulación mensual de stock (+) o eliminación (-)", fontsize=20)

plt.axhline(y=0, color='black', lw=2, ls=':');

# plt.savefig("multimedia/e-s_variacion_stock_mes.png")
```



2.3 Estacionalidad de entradas y salidas

¿Qué movientos diarios y mensuales hay a lo largo del año en volumen de entradas y salidas?

DIARIO

- La entradas (verde) son superiores a la salidas (rojo) al principio de año y final de año (Navidades) y entre los días 240 y 280 (verano).
- Hay una disminución del ritmo de entradas y salidas tanto en verano como en los meses finales del año.

MENSUAL

- El mayor volumen de entradas y salidas se produce en octubre. El segundo mayor número de entradas se produce en septiembre. Las entradas son altas después del verano.

DÍA DE LA SEMANA

- De L-J las entradas y salidas son elevadas y similares.
- S-D-L son los 3 días en los que las entradas son mayores a las salidas.
- El viernes se produce la mayor eliminación de stock y el momento de salidas más elevado.
- El fin de semana entradas y salidas descienden, especialmente las salidas.

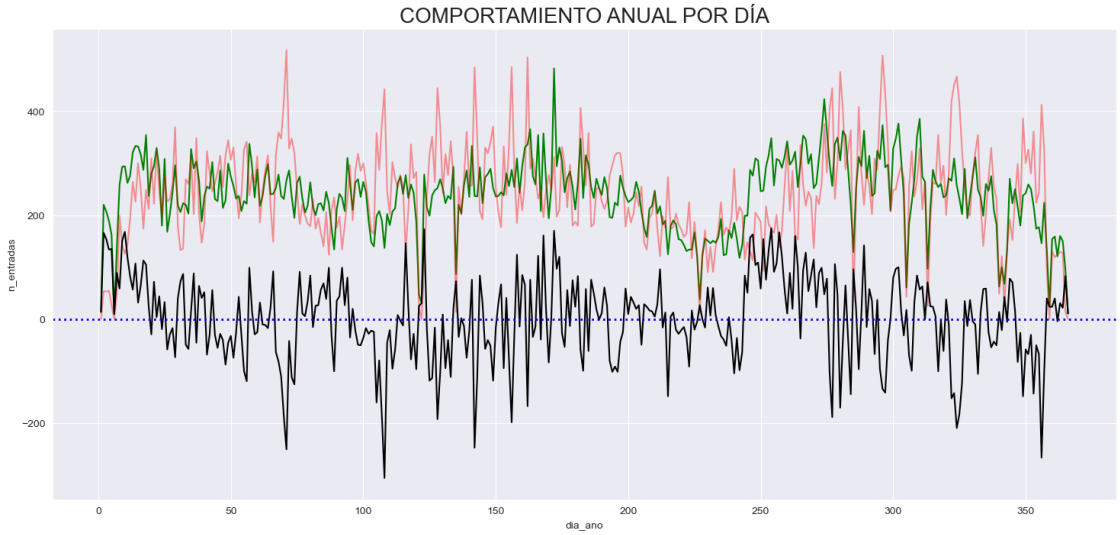
[24]:
```
# Entradas

df_ce = mf.componentes_fecha(df_sin, "fecha_entrada")

df_ce.drop(['hora', 'minuto', 'segundo'], axis=1, inplace=True)

df_ced = df_ce.groupby('dia_ano').count()
```

[25]:
```
# Salidas

df_cs = mf.componentes_fecha(df_sin, "fecha_contestacion")

df_cs.drop(['hora', 'minuto', 'segundo'], axis=1, inplace=True)

df_csd = df_cs.groupby('dia_ano').count()
```

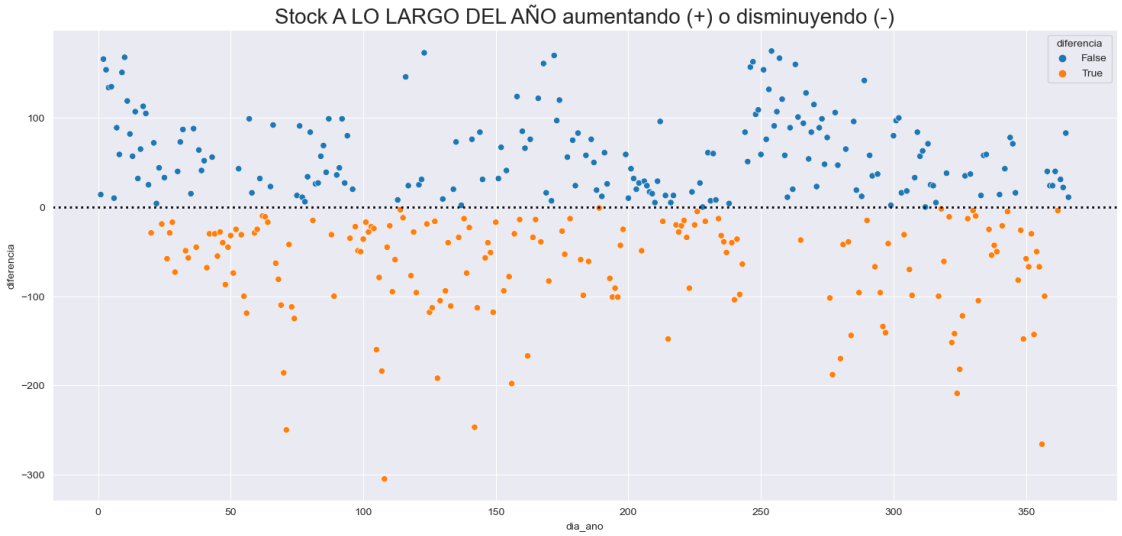
[26]:
```
# DataFrame de entradas, salidas y diferencia en términos anuales
es_da = pd.concat([df_ced['tipo'], df_csd['tipo']], axis=1).fillna(0)
es_da.columns = ['n_entradas', 'n_salidas']

# Esta columna es REDUCCIÓN STOCK (-) o AUMENTO STOCK (+)
es_da['diferencia'] = es_da.n_entradas - es_da.n_salidas

es_da['entradas_acum'] = es_da.n_entradas.cumsum()
```

```
es_da['salidas_acum'] = es_da.n_salidas.cumsum()

# Cuántas entradas se van acumulando sin contestar
es_da['stock'] = es_da['entradas_acum'] - es_da['salidas_acum']
```

[27]:
```
# Entradas, Salidas y Diferencia

plt.figure(figsize=(18,8))

sns.lineplot(data=es_da, x=es_da.index, y=es_da.n_entradas, color='green')
sns.lineplot(data=es_da, x=es_da.index, y=es_da.n_salidas, color='red', alpha=0.
 ↪4)
sns.lineplot(data=es_da, x=es_da.index, y=es_da.diferencia, color='black') \
        .set_title(label='COMPORTAMIENTO ANUAL POR DÍA', loc='center', size=20)

plt.axhline(y=es_da.diferencia.mean(), color='blue', lw=2, ls=':');
```



[28]:
```
# Acumulación o eliminación de stock de entradas

r_ad = es_da.diferencia < 0

plt.figure(figsize=(18,8))

sns.scatterplot(x=es_da.index, y=es_da.diferencia, hue=r_ad)

plt.title("Stock A LO LARGO DEL AÑO aumentando (+) o disminuyendo (-)",␣
 ↪fontsize=20)

plt.axhline(y=0, color='black', lw=2, ls=':');
```

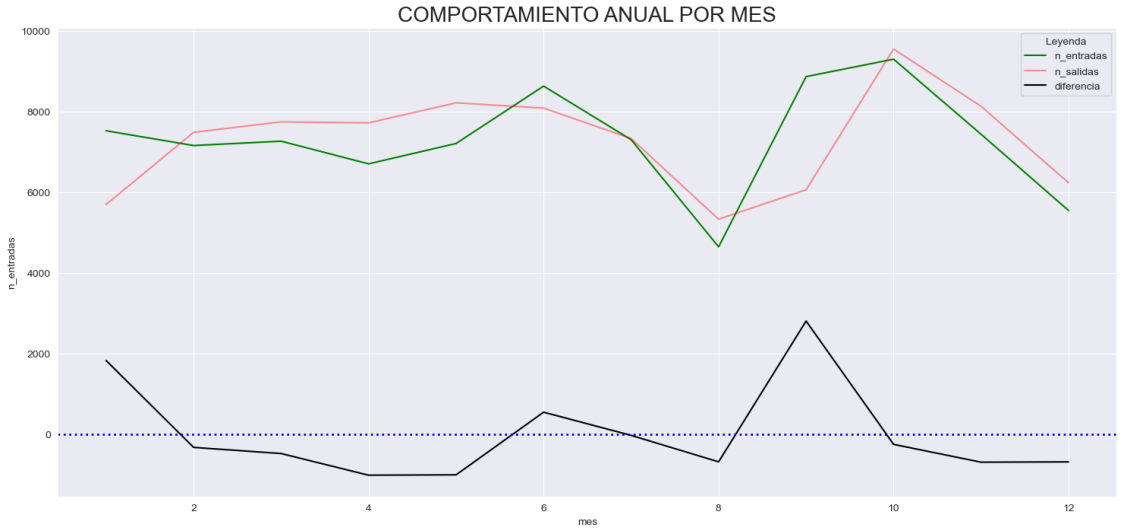


[29]:
```
# Entradas mes

# df_ce = mf.componentes_fecha(df_sin, "fecha_entrada")

# df_ce.drop(['hora', 'minuto', 'segundo'], axis=1, inplace=True)

df_cem = df_ce.groupby('mes').count()
```

[30]:
```
# Salidas mes

# df_cs = mf.componentes_fecha(df_sin, "fecha_contestacion")

# df_cs.drop(['hora', 'minuto', 'segundo'], axis=1, inplace=True)

df_csm = df_cs.groupby('mes').count()
```

[31]:
```
# DataFrame de entradas, salidas y diferencia en términos anuales
es_ma = pd.concat([df_cem['tipo'], df_csm['tipo']], axis=1).fillna(0)
es_ma.columns = ['n_entradas', 'n_salidas']

# Esta columna es REDUCCIÓN STOCK (-) o AUMENTO STOCK (+)
es_ma['diferencia'] = es_ma.n_entradas - es_ma.n_salidas

es_ma['entradas_acum'] = es_ma.n_entradas.cumsum()
es_ma['salidas_acum'] = es_ma.n_salidas.cumsum()

# Cuántas entradas se van acumulando sin contestar
es_ma['stock'] = es_ma['entradas_acum'] - es_ma['salidas_acum']
```

[32]:
```
# Entradas, Salidas y Diferencia

plt.figure(figsize=(18,8))

sns.lineplot(data=es_ma, x=es_ma.index, y=es_ma.n_entradas, color='green',␣
 ↪label='n_entradas')
sns.lineplot(data=es_ma, x=es_ma.index, y=es_ma.n_salidas, color='red', alpha=0.
 ↪4, label='n_salidas')
sns.lineplot(data=es_ma, x=es_ma.index, y=es_ma.diferencia, color='black',␣
 ↪label='diferencia') \
        .set_title(label='COMPORTAMIENTO ANUAL POR MES', loc='center', size=20)

plt.axhline(y=es_ma.diferencia.mean(), color='blue', lw=2, ls=':')
plt.legend(title='Leyenda');

# plt.savefig("multimedia/e-s_y_diferencia_mes.png")
```

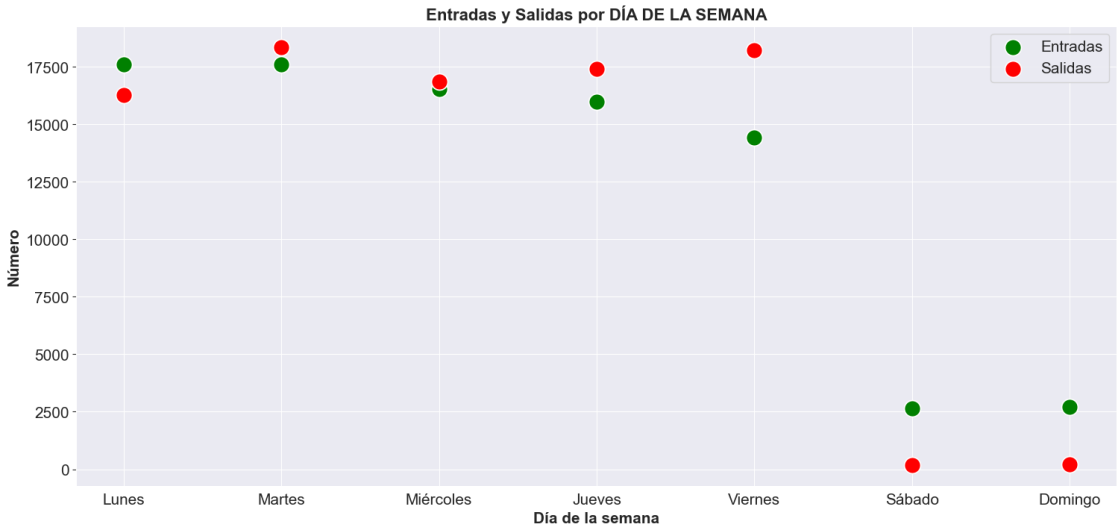


[33]:
```
# Entradas día de la semana

df_ceds = df_ce.groupby('dia_semana').count()

# Salidas día de la semana

df_csds = df_cs.groupby('dia_semana').count()
```

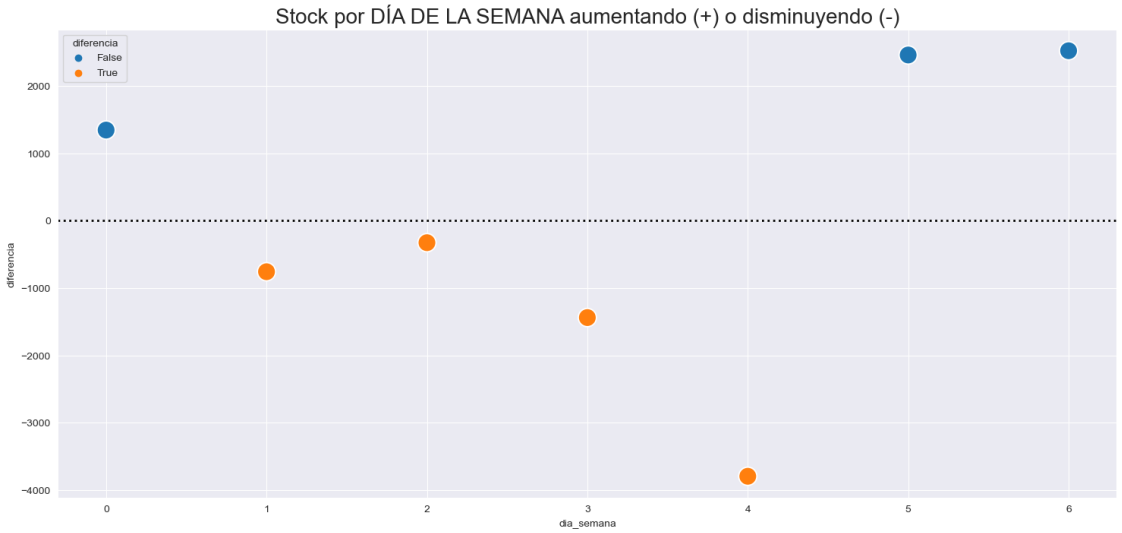
[34]:
```
# DataFrame de entradas, salidas y diferencia en términos diarios semanales
es_dsa = pd.concat([df_ceds['tipo'], df_csds['tipo']], axis=1).fillna(0)
es_dsa.columns = ['n_entradas', 'n_salidas']
```

```
# Esta columna es REDUCCIÓN STOCK (-) o AUMENTO STOCK (+)
es_dsa['diferencia'] = es_dsa.n_entradas - es_dsa.n_salidas

es_dsa['entradas_acum'] = es_dsa.n_entradas.cumsum()
es_dsa['salidas_acum'] = es_dsa.n_salidas.cumsum()

# Cuántas entradas se van acumulando sin contestar
es_dsa['stock'] = es_dsa['entradas_acum'] - es_dsa['salidas_acum']
```

[35]:
```
# Entradas, Salidas y Diferencia

etiquetas = ['Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado',␣
 ↪'Domingo']

f, ax = plt.subplots(1,1, figsize=(18,8))

sns.scatterplot(ax=ax, x=es_dsa.index, y=es_dsa.n_entradas, color='green',␣
 ↪s=250)
sns.scatterplot(ax=ax, x=es_dsa.index, y=es_dsa.n_salidas, color='red', s=250)

ax.set_title('Entradas y Salidas por DÍA DE LA SEMANA', fontsize=16,␣
 ↪loc='center', fontdict=dict(weight='bold'))
ax.set_xlabel('Día de la semana', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=es_dsa.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))
ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)
ax.yaxis.tick_left()
ax.legend(loc='upper right', labels=['Entradas', 'Salidas'], fontsize=15);
```



[36]:
```
# Acumulación o eliminación de stock de entradas

r_dsa = es_dsa.diferencia < 0

plt.figure(figsize=(18,8))

sns.scatterplot(x=es_dsa.index, y=es_dsa.diferencia, hue=r_dsa, s=300)

plt.title("Stock por DÍA DE LA SEMANA aumentando (+) o disminuyendo (-)",␣
 ↪fontsize=20)

plt.axhline(y=0, color='black', lw=2, ls=':');

# plt.savefig("multimedia/variacion_stock_dia_semana.png")
```



2.4 Evolución histórica del procedimiento: éxito, éxito total y terminada

¿Cómo ha cambiado a lo largo del tiempo la tasa de éxito y de preguntas terminadas?

Se establecen 2 estados:

1. **En proceso**: la entrada está siendo tratada.
2. **Terminada**: la entrada ya ha sido tratada.

Dentro de **Terminada** hay 2 posibles estados:

1. **Éxito**: la entrada ha sido contestada al ciudadano.
2. **Descartada**: la entrada ha sido rechazada, cancelada o enviada a otra administración.

- Tasa global de éxito (EFICIENCIA): 89,11%.
- Tasa global de terminada (EFICACIA): 99.52%.
- Tasa global de éxito total sobre volumen de entrada de trabajo: 88,68%.
- Tasa global de en proceso: 0,48%.
- Hay un **mejora** a lo largo del tiempo **en las tasas de éxito (verde) y éxito total (naranja)**, y a partir de 2020 se estabilizan. El número total de entradas (rojo) es estable a lo largo del tiempo. Esto indica que **la mejora en la tasa de éxito no proviene de tener menor carga de trabajo**.
- La **tasa de terminación (azul) empeora a apartir de 2021**.
- La **tasa de descarte (negro) baja a lo largo del tiempo**, y a partir de 2020 se estabiliza.
- El **número de entradas en proceso** (marrón) **se acumula** en torno a 10-20 **desde el año 2021** (la línea discontinua marca el final del año (octubre) por el retardo entrada-respuesta).

[37]:
```
# DataFrame provisional
dfprov = df.copy()
```

[38]:
```
# Sustitución de estados
dfprov['estado'] = dfprov['estado'].replace(
    ['Recepcionada por Coordinador', 'Recepcionada por Unidad', 'Enviada␣
 ↪respuesta a Coordinador'], 'En proceso')
dfprov['estado'] = dfprov['estado'].replace(
    ['Cancelada', 'Enviada a Otra Administración', 'Rechazada por SGAC'],␣
 ↪'Descarte')
dfprov['estado'] = dfprov['estado'].replace(
    ['Contestada al Ciudadano'], 'Exito')

# Número de registros en cada situación
dfprov.estado.value_counts()
```

[38]:
```
Exito         88676
Descarte      10839
En proceso      482
Name: estado, dtype: int64
```

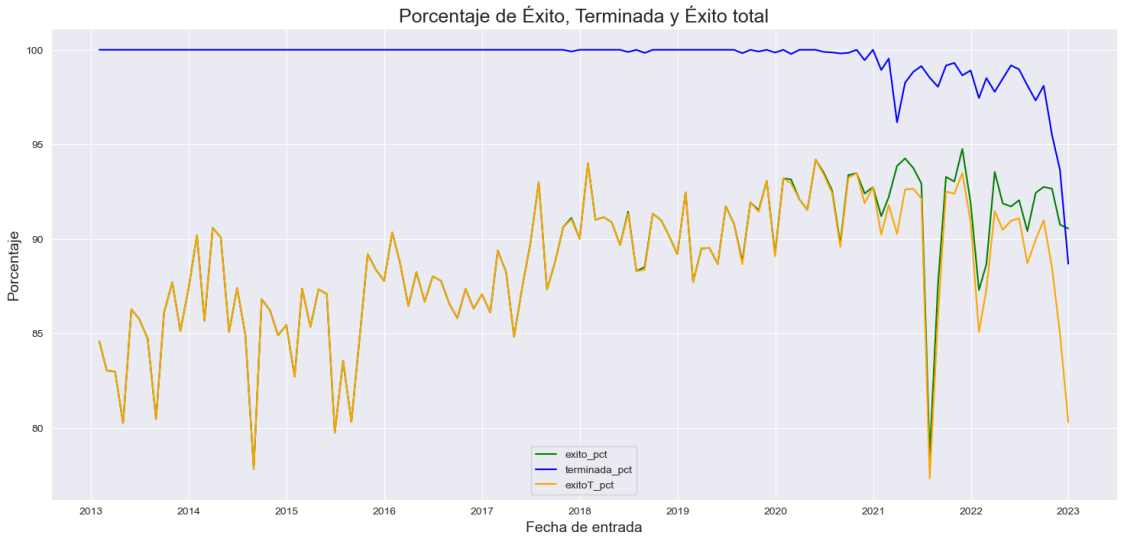
[39]:
```
# Tasa global de éxito, terminada y éxito total
e = dfprov[dfprov.estado == "Exito"].estado.count()
tt = dfprov[dfprov.estado == "Descarte"].estado.count() + e
p = dfprov[dfprov.estado == "En proceso"].estado.count()

print("Tasa global de éxito (EFICIENCIA): ", 100*e/(tt))
print("Tasa global de terminada (EFICACIA): ", 100*tt/(tt+p))
print("Tasa global de éxito total sobre volumen de entrada de trabajo: ", 100*e/
 ↪(tt+p))
print("Tasa global de en proceso: ", 100*p/(tt+p))
```

```
Tasa global de éxito (EFICIENCIA):  89.10817464703814
Tasa global de terminada (EFICACIA):  99.51798553956618
Tasa global de éxito total sobre volumen de entrada de trabajo:
88.67866035981079
Tasa global de en proceso:  0.482014460433813
```

[40]:
```
# Resample de las entradas a MES
df_em = df_e.resample('M').agg(descarte=('descarte', 'sum'),
                               exito=('exito', 'sum'),
                               proceso=('proceso', 'sum'),
                               tipo=('tipo', 'count'))  # Esta última es solo el␣
 ↪recuento.

# Columnas de porcentajes
df_em['exito_pct'] = 100*df_em.exito / (df_em.descarte + df_em.exito)
```

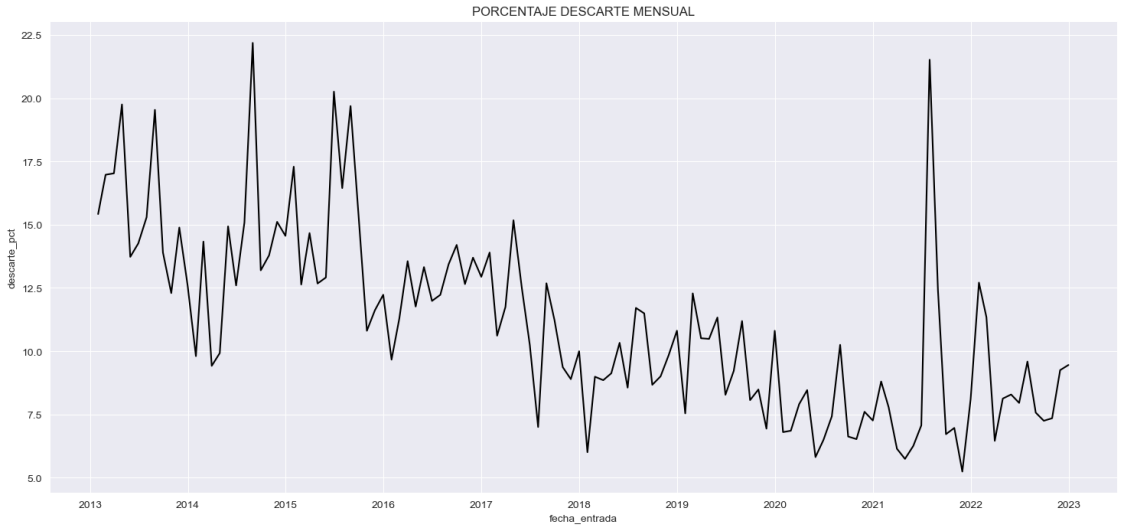
```
df_em['terminada_pct'] = 100*(df_em.descarte + df_em.exito) / (df_em.exito +␣
 ↪df_em.descarte + df_em.proceso)
df_em['exitoT_pct'] = 100*df_em.exito / (df_em.exito + df_em.descarte + df_em.
 ↪proceso)
df_em['descarte_pct'] = 100*df_em.descarte / (df_em.descarte + df_em.exito)
```

[41]:
```
# Gráfico de tasas de éxito, terminada y éxito total, a partir de fecha de␣
 ↪entrada
f, ax = plt.subplots(1,1, figsize=(18,8))

g = df_em[['exito_pct', 'terminada_pct', 'exitoT_pct']]

sns.lineplot(data=g, palette=['green', 'blue', 'orange'], dashes=False)

plt.legend(loc='lower center')

plt.title("Porcentaje de Éxito, Terminada y Éxito total", fontsize=18,␣
 ↪loc='center')
plt.ylabel("Porcentaje", fontsize=14)
plt.xlabel("Fecha de entrada", fontsize=14);

# plt.savefig("multimedia/pct_exito_term_extot.png")
```

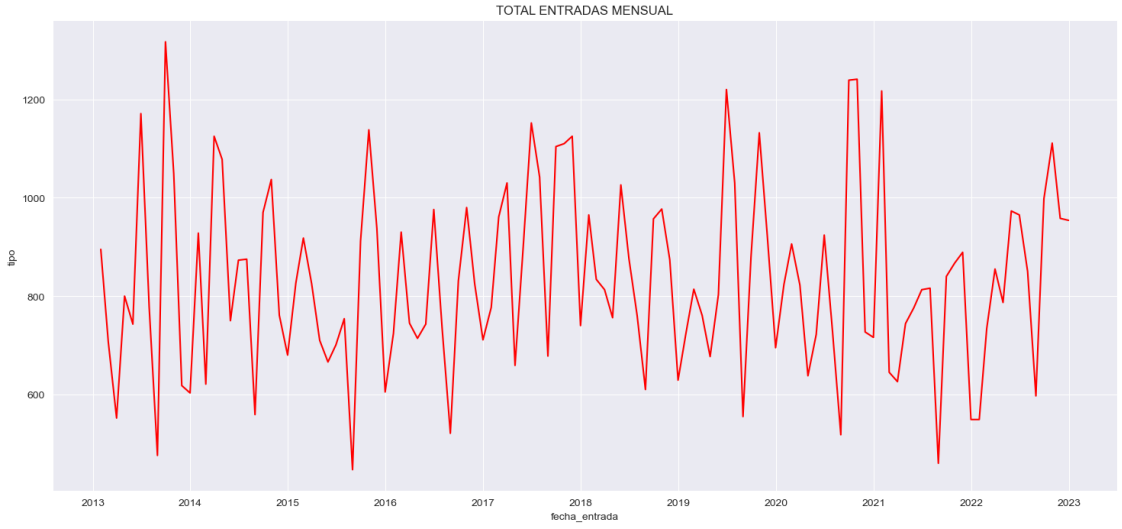


[42]:
```
# Gráfico de descartadas
f, ax = plt.subplots(1,1, figsize=(18,8))

sns.lineplot(ax=ax, data=df_em, x=df_em.index, y=df_em.descarte_pct,␣
 ↪color='black') \
```

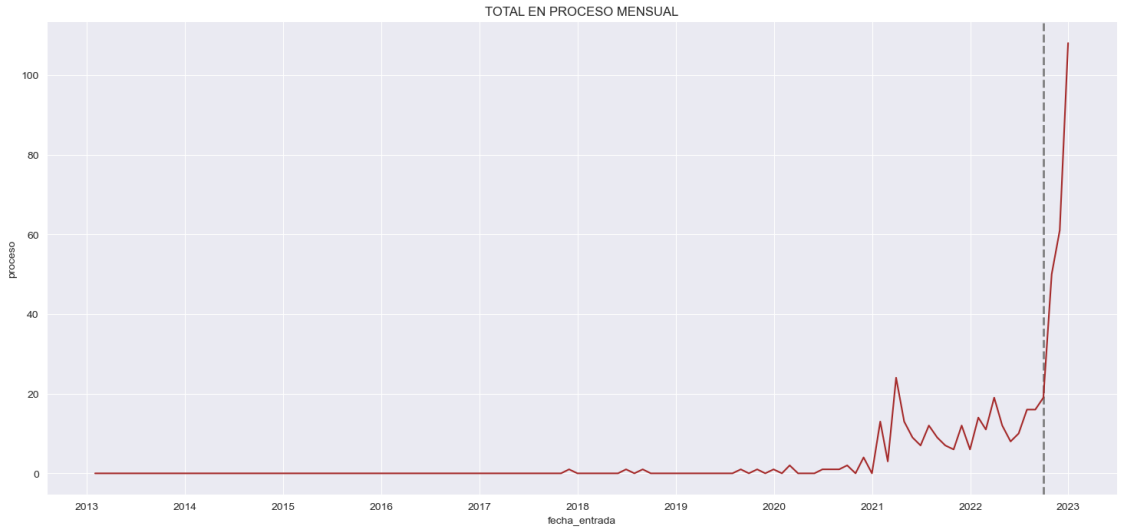
```
            .set_title(label='PORCENTAJE DESCARTE MENSUAL', loc='center');

# plt.savefig("multimedia/pct_descarte.png")
```



[43]:
```
# Gráfico de total entradas
f, ax = plt.subplots(1,1, figsize=(18,8))

sns.lineplot(ax=ax, data=df_em, x=df_em.index, y=df_em.tipo, color='red') \
            .set_title(label='TOTAL ENTRADAS MENSUAL', loc='center');

# plt.savefig("multimedia/total_entradas_mes.png")
```



[44]:
```
# Gráfico de total en proceso
f, ax = plt.subplots(1,1, figsize=(18,8))

plt.axvline(x=df_em.index[-4], color='grey', lw=2, ls='--')  #␣
 ↪Octubre-Diciembre parte final del año.

sns.lineplot(ax=ax, data=df_em, x=df_em.index, y=df_em.proceso, color='brown') \
            .set_title(label='TOTAL EN PROCESO MENSUAL', loc='center');

# plt.savefig("multimedia/total_en_proceso.png")
```



2.5 Estacionalidad del procedimiento: éxito, éxito total y terminada

2.5.1 Mensual

Meses del año en los que el procesamiento de las entradas es mejor.

Eficiencia

- Septiembre y octubre, **tras el verano, son los meses de mayor eficiencia**.

Ineficiencia

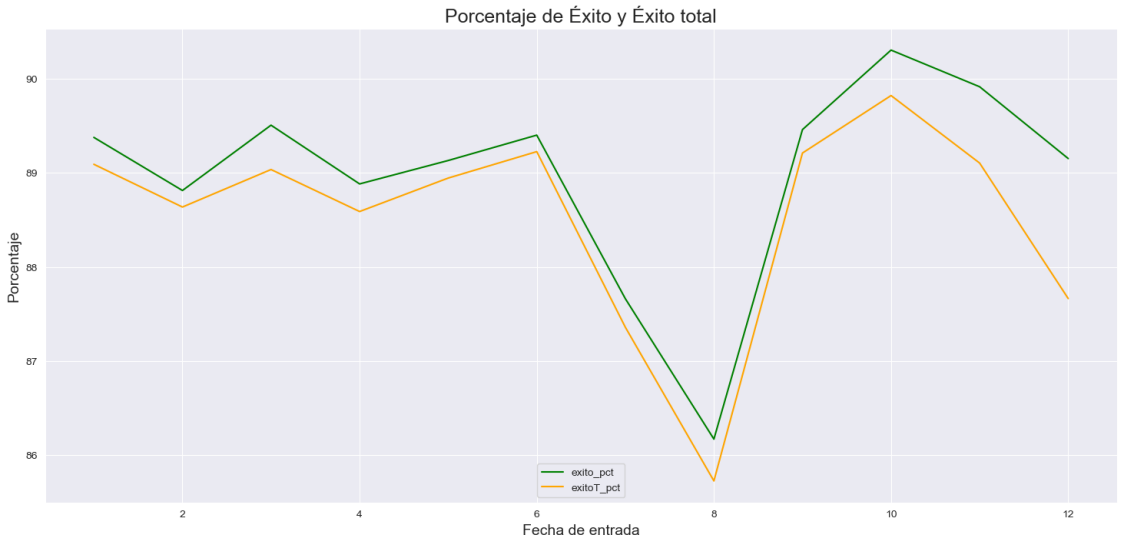
- El **porcentaje de éxito y el éxito total baja en agosto** (mes 8, líneas verde y naranja). En menor medida, también en terminada (línea azul).
- El **porcentaje de descarte sube en agosto** (línea negra). **Pero el número de descartes disminuye y el número de terminadas disminuye aún más**. Esto genera que, en términos relativos, el porcentaje de descartes aumente.

[45]:
```
# Obtener componentes de "fecha_entrada"
df_e = mf.componentes_fecha(df_e.reset_index(), "fecha_entrada")
```

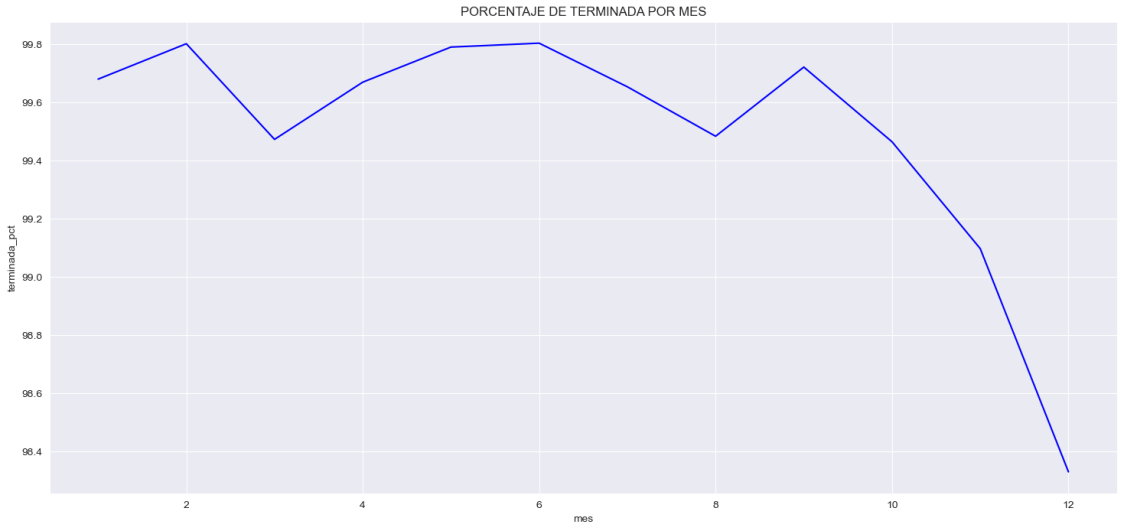
[46]:
```
# Entradas: agrupar por mes del año
df_eem = df_e.groupby('mes').agg(
    exito=('exito', 'sum'),
    descarte=('descarte', 'sum'),
    proceso=('proceso', 'sum'),
    total=('tipo', 'count'))  # Esta última es solo el recuento.
```

[47]:
```
df_eem['terminada'] = df_eem.exito + df_eem.descarte
df_eem['exito_pct'] = 100*df_eem.exito / (df_eem.descarte + df_eem.exito)
df_eem['descarte_pct'] = 100*df_eem.descarte / (df_eem.descarte + df_eem.exito)
df_eem['terminada_pct'] = 100*(df_eem.descarte + df_eem.exito) / (df_eem.exito␣
 ↪+ df_eem.descarte + df_eem.proceso)
df_eem['proceso_pct'] = 100*df_eem.proceso / (df_eem.proceso + df_eem.terminada)
df_eem['exitoT_pct'] = 100*df_eem.exito / (df_eem.exito + df_eem.descarte +␣
 ↪df_eem.proceso)
```

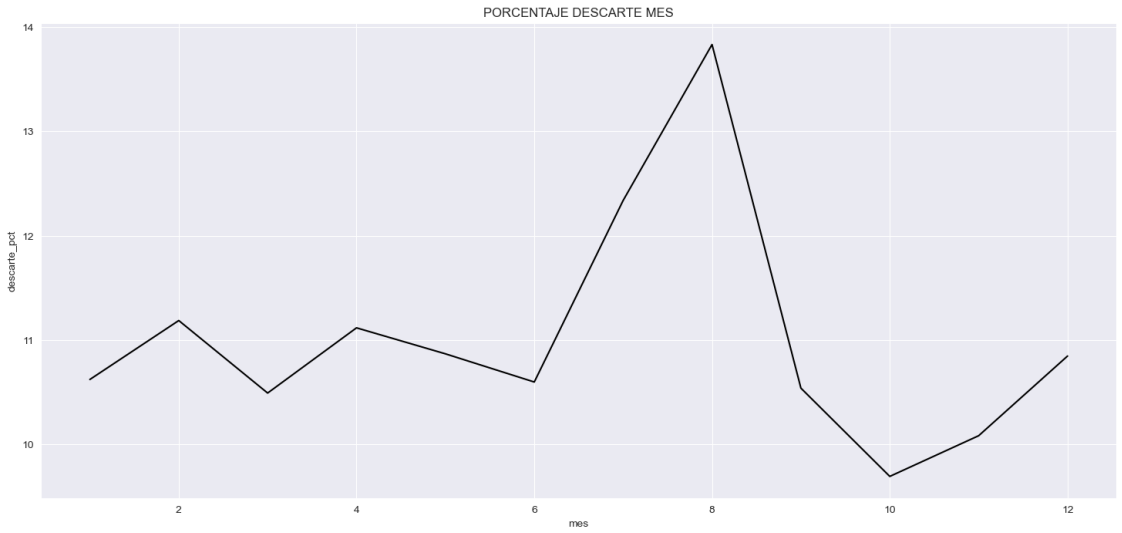
[48]:
```
# Gráfico de tasas de éxito y éxito total por mes
f, ax = plt.subplots(1,1, figsize=(18,8))

gg = df_eem[['exito_pct', 'exitoT_pct']]

sns.lineplot(data=gg, palette=['green', 'orange'], dashes=False)

plt.legend(loc='lower center')
plt.title("Porcentaje de Éxito y Éxito total", fontsize=18, loc='center')
plt.ylabel("Porcentaje", fontsize=14)
plt.xlabel("Fecha de entrada", fontsize=14);
```



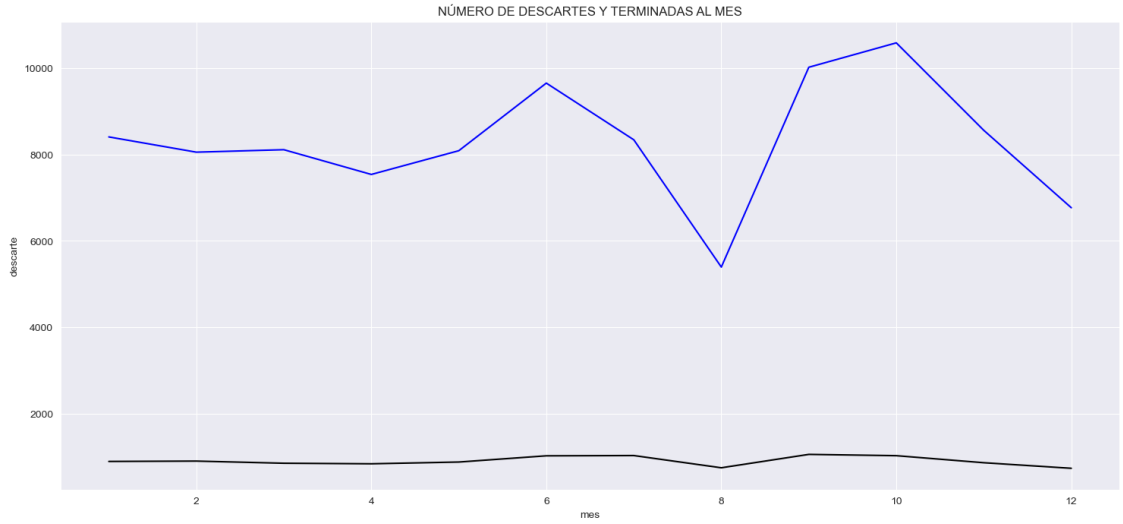
[49]:
```
# Gráfico de terminadas mensual
f, ax = plt.subplots(1,1, figsize=(18,8))

sns.lineplot(ax=ax, data=df_eem, x=df_eem.index, y=df_eem.terminada_pct,␣
 ↪color='blue') \
            .set_title(label='PORCENTAJE DE TERMINADA POR MES', loc='center');
```



[50]:
```
# Gráfico de descartadas mensual
f, ax = plt.subplots(1,1, figsize=(18,8))

sns.lineplot(ax=ax, data=df_eem, x=df_eem.index, y=df_eem.descarte_pct,␣
 ↪color='black') \
            .set_title(label='PORCENTAJE DESCARTE MES', loc='center');
```



[51]:
```
# Gráfico de número descartes y terminadas al mes
f, ax = plt.subplots(1,1, figsize=(18,8))

sns.lineplot(ax=ax, data=df_eem, x=df_eem.index, y=df_eem.descarte,␣
 ↪color='black')

sns.lineplot(ax=ax, data=df_eem, x=df_eem.index, y=df_eem.terminada,␣
 ↪color='blue') \
            .set_title(label='NÚMERO DE DESCARTES Y TERMINADAS AL MES',␣
 ↪loc='center');
```



2.5.2 Día de la semana

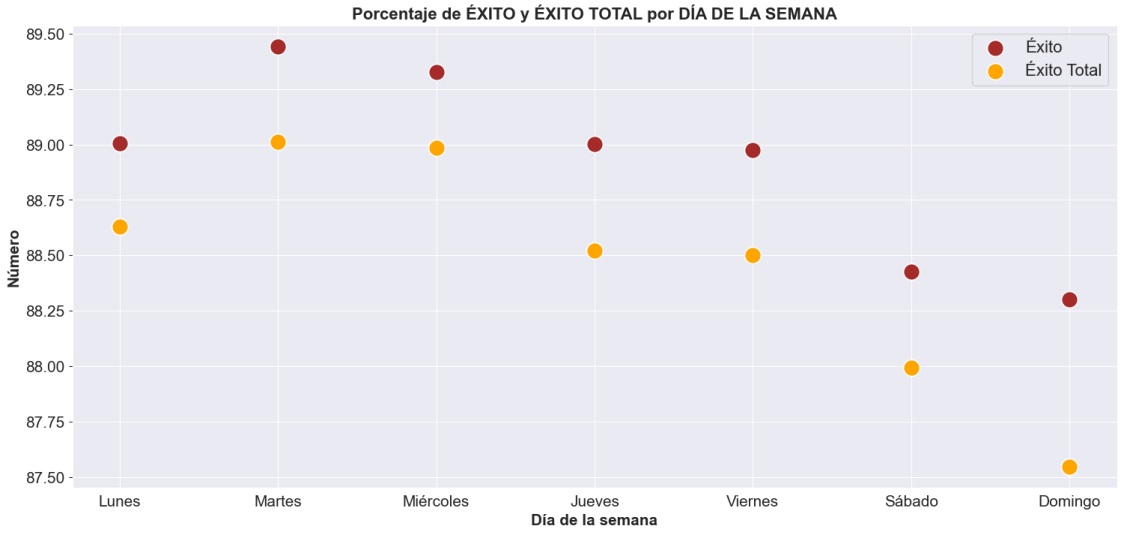
Días de la semana en los que el procesamiento es mejor o peor.

- El porcentaje varía muy poco (1,5%), aunque gráficamente la amplitud parece mayor.
- El porcentaje de descarte sube el fin de semana en más de 1% respecto al valle que se produce el martes (no llega al 10,6%).

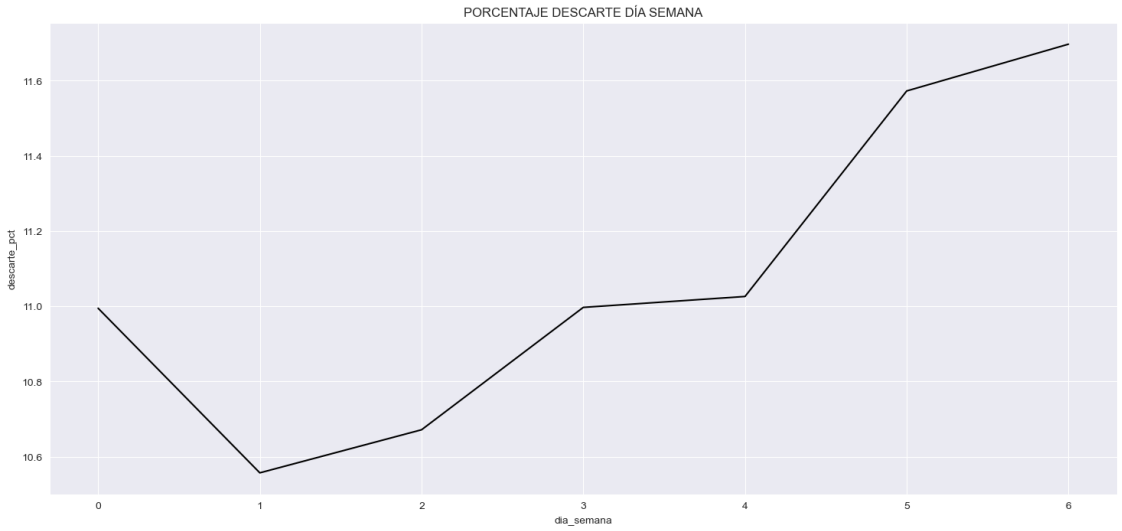
[52]:
```
# Entradas: agrupar por día de la semana
df_eeds = df_e.groupby('dia_semana').agg(
    exito=('exito', 'sum'),
    descarte=('descarte', 'sum'),
    proceso=('proceso', 'sum'),
    total=('tipo', 'count'))  # Esta última es solo el recuento.
```

[53]:
```
df_eeds['terminada'] = df_eeds.exito + df_eeds.descarte
df_eeds['exito_pct'] = 100*df_eeds.exito / (df_eeds.descarte + df_eeds.exito)
df_eeds['descarte_pct'] = 100*df_eeds.descarte / (df_eeds.descarte + df_eeds.
 ↪exito)
df_eeds['terminada_pct'] = 100*(df_eeds.descarte + df_eeds.exito) / (df_eeds.
 ↪exito + df_eeds.descarte + df_eeds.proceso)
df_eeds['proceso_pct'] = 100*df_eeds.proceso / (df_eeds.proceso + df_eeds.
 ↪terminada)
df_eeds['exitoT_pct'] = 100*df_eeds.exito / (df_eeds.exito + df_eeds.descarte +␣
 ↪df_eeds.proceso)
```

[54]:
```
# Gráfico de tasas de éxito y éxito total por día de la semana

f, ax = plt.subplots(1,1, figsize=(18,8))

sns.scatterplot(ax=ax, x=df_eeds.index, y=df_eeds.exito_pct, color='brown',␣
 ↪s=250)
sns.scatterplot(ax=ax, x=df_eeds.index, y=df_eeds.exitoT_pct, color='orange',␣
 ↪s=250)

ax.set_title('Porcentaje de ÉXITO y ÉXITO TOTAL por DÍA DE LA SEMANA',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_eeds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.yaxis.tick_left()

ax.legend(loc='upper right', labels=['Éxito', 'Éxito Total'], fontsize=16);
```



[55]:
```
# Gráfico de descartadas por día de la semana
f, ax = plt.subplots(1,1, figsize=(18,8))

sns.lineplot(ax=ax, data=df_eeds, x=df_eeds.index, y=df_eeds.descarte_pct,␣
↪color='black') \
            .set_title(label='PORCENTAJE DESCARTE DÍA SEMANA', loc='center');
```



3 Diferencias en las variables categóricas - Sin Temporalidad

3.1 Entradas en proceso

¿Hay variables cuyas tasas de procesamiento sean peores, acumulando entradas sin terminar (en proceso)?

Variable **tema** acumula entradas en proceso en:

- **ENERGÍA** (29,16%). Destaca sobre el siguiente: Políticas sociales (0,91%).

Variable **consejeria** acumula entradas en proceso en:

- **Consejería de Medio Ambiente, Vivienda y Agricultura** (21,5%).
- **Consejería de Transportes e Infraestructuras** (10,3%).

Destacan sobre el siguiente: Consejería de Familia, Juventud y Política Social (2,56%).

Esto puede deberse: 1. O bien, cuestiones sobre temas más difíciles de resolver. 2. O bien, consejerías más ineficaces.

[56]:
```
# Sustitución de estados: En proceso y Terminada
df_t = df.copy()
df_t['estado'] = df_t['estado'].replace(
    ['Cancelada', 'Enviada a Otra Administración', 'Rechazada por SGAC',␣
 ↪'Contestada al Ciudadano'], 'Terminada')
df_t['estado'] = df_t['estado'].replace(
    ['Recepcionada por Coordinador', 'Recepcionada por Unidad', 'Enviada␣
 ↪respuesta a Coordinador'], 'En proceso')
```

[57]:
```
pd.crosstab(df_t.tema, df_t.estado, margins=True, margins_name='Total')
```

[57]:
```
estado              En proceso  Terminada  Total
tema
CULTURA                      9       8906   8915
EDUCACIÓN                    8      24848  24856
ENERGÍA                     21         51     72
POLÍTICAS SOCIALES         376      40919  41295
TRANSPORTES                 68      24791  24859
Total                      482      99515  99997
```

[58]:
```
ct_t = pd.crosstab(df_t.tema, df_t.estado, margins=False, normalize='index')
ct_t
```

[58]:
```
estado              En proceso  Terminada
tema
CULTURA               0.001010   0.998990
EDUCACIÓN             0.000322   0.999678
ENERGÍA               0.291667   0.708333
```

```
POLÍTICAS SOCIALES    0.009105   0.990895
TRANSPORTES           0.002735   0.997265
```

[59]:
```
pd.crosstab(df_t.consejeria, df_t.estado, margins=True, margins_name='Total')
```

[59]:
```
estado                                             En proceso  Terminada  \
consejeria
Agencia para la Administración Digital de la Co…           0        565
Consejería de Cultura, Turismo y Deporte                   8       3993
Consejería de Economía, Hacienda y Empleo                  0        181
Consejería de Educación e Investigación                    0       4198
Consejería de Educación, Juventud y Deporte                0       8865
Consejería de Educación, Universidades, Ciencia…           8      10812
Consejería de Familia, Juventud y Política Social        377      14338
Consejería de Medio Ambiente, Ordenación del Te…           0         31
Consejería de Medio Ambiente, Vivienda y Agricu…          20         73
Consejería de Políticas Sociales, Familias, Igu…           0      25377
Consejería de Presidencia, Justicia e Interior             2        230
Consejería de Presidencia, Justicia y Portavocí…           0         61
Consejería de Sanidad                                      0        109
Consejería de Transportes e Infraestructuras              67        578
Consejería de Transportes, Vivienda e Infraestr…           0       1681
Consorcio Regional de Transportes                          0      19471
Oficina de Cultura y Turismo                               0       3975
Vicepresidencia, Consejería de Deportes, Transp…           0         12
Total                                                    482      94550

estado                                             Total
consejeria
Agencia para la Administración Digital de la Co…    565
Consejería de Cultura, Turismo y Deporte           4001
Consejería de Economía, Hacienda y Empleo           181
Consejería de Educación e Investigación            4198
Consejería de Educación, Juventud y Deporte        8865
Consejería de Educación, Universidades, Ciencia…  10820
Consejería de Familia, Juventud y Política Social  14715
Consejería de Medio Ambiente, Ordenación del Te…     31
Consejería de Medio Ambiente, Vivienda y Agricu…     93
Consejería de Políticas Sociales, Familias, Igu…  25377
Consejería de Presidencia, Justicia e Interior      232
Consejería de Presidencia, Justicia y Portavocí…     61
Consejería de Sanidad                               109
Consejería de Transportes e Infraestructuras        645
Consejería de Transportes, Vivienda e Infraestr…   1681
Consorcio Regional de Transportes                 19471
Oficina de Cultura y Turismo                       3975
Vicepresidencia, Consejería de Deportes, Transp…     12
```

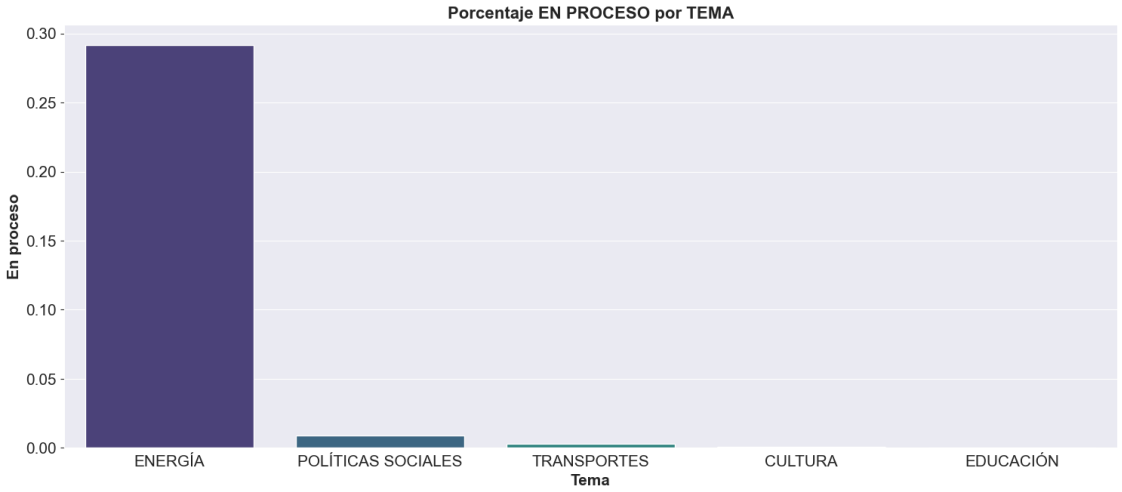
```
Total                                                 95032
```

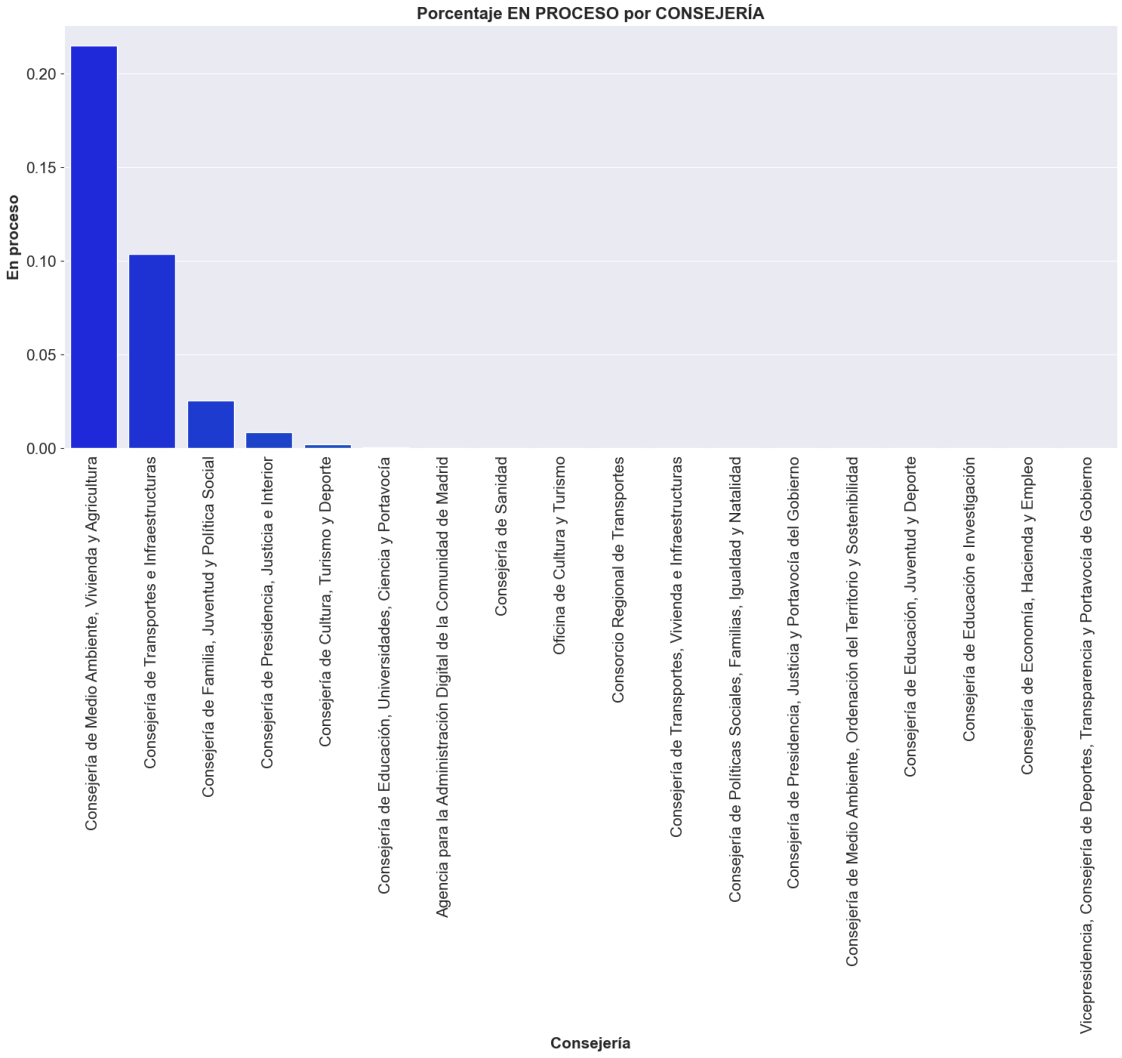
[60]:
```
ct_c = pd.crosstab(df_t.consejeria, df_t.estado, margins=False,␣
 ↪normalize='index')
ct_c
```

[60]:
```
estado                                              En proceso  Terminada
consejeria
Agencia para la Administración Digital de la Co…    0.000000   1.000000
Consejería de Cultura, Turismo y Deporte              0.002000   0.998000
Consejería de Economía, Hacienda y Empleo             0.000000   1.000000
Consejería de Educación e Investigación               0.000000   1.000000
Consejería de Educación, Juventud y Deporte           0.000000   1.000000
Consejería de Educación, Universidades, Ciencia…    0.000739   0.999261
Consejería de Familia, Juventud y Política Social     0.025620   0.974380
Consejería de Medio Ambiente, Ordenación del Te…    0.000000   1.000000
Consejería de Medio Ambiente, Vivienda y Agricu…    0.215054   0.784946
Consejería de Políticas Sociales, Familias, Igu…    0.000000   1.000000
Consejería de Presidencia, Justicia e Interior        0.008621   0.991379
Consejería de Presidencia, Justicia y Portavocí…    0.000000   1.000000
Consejería de Sanidad                                 0.000000   1.000000
Consejería de Transportes e Infraestructuras          0.103876   0.896124
Consejería de Transportes, Vivienda e Infraestr…    0.000000   1.000000
Consorcio Regional de Transportes                     0.000000   1.000000
Oficina de Cultura y Turismo                          0.000000   1.000000
Vicepresidencia, Consejería de Deportes, Transp…    0.000000   1.000000
```

[61]:
```
# Gráficos TEMA y CONSEJERIA para En proceso

f, ax = plt.subplots(2, 1, figsize=(18,16))

# TEMA
sns.barplot(ax=ax[0], x=ct_t.index, y=ct_t['En proceso'], palette='viridis',
            order=ct_t['En proceso'].sort_values(ascending=False).index)

ax[0].set_title('Porcentaje EN PROCESO por TEMA', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[0].set_xlabel('Tema', fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('En proceso', fontsize=15, fontdict=dict(weight='bold'))
ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[0].yaxis.tick_left() # Donde se ponen las marcas de eje
# ax[1].legend(loc='upper right') # Si es múltiple: label=''


# CONSEJERIA
sns.barplot(ax=ax[1], x=ct_c.index, y=ct_c['En proceso'], palette='winter',
```

```
            order=ct_c['En proceso'].sort_values(ascending=False).index)

ax[1].set_title('Porcentaje EN PROCESO por CONSEJERÍA', fontsize=16,␣
 ↪loc='center', fontdict=dict(weight='bold'))
ax[1].set_xlabel('Consejería', fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('En proceso', fontsize=15, fontdict=dict(weight='bold'))
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=90)
ax[1].tick_params(axis='y', which='major', labelsize=15)
ax[1].yaxis.tick_left(); # Donde se ponen las marcas de eje
# ax[1].legend(loc='upper right') # Si es múltiple: label=''

# plt.savefig("multimedia/en_proceso_tema_y_consejeria.png")
```


Porcentaje EN PROCESO por TEMA
En proceso
0.30
0.25
0.20
0.15
0.10
0.05
0.00
ENERGÍA
POLÍTICAS SOCIALES
TRANSPORTES
CULTURA
EDUCACIÓN
Tema


Porcentaje EN PROCESO por CONSEJERÍA
En proceso
0.20
0.15
0.10
0.05
0.00
Consejería de Medio Ambiente, Vivienda y Agricultura
Consejería de Transportes e Infraestructuras
Consejería de Familia, Juventud y Política Social
Consejería de Presidencia, Justicia e Interior
Consejería de Cultura, Turismo y Deporte
Consejería de Educación, Universidades, Ciencia y Portavocía
Agencia para la Administración Digital de la Comunidad de Madrid
Consejería de Sanidad
Oficina de Cultura y Turismo
Consorcio Regional de Transportes
Consejería de Transportes, Vivienda e Infraestructuras
Consejería de Políticas Sociales, Familias, Igualdad y Natalidad
Consejería de Presidencia, Justicia y Portavocía del Gobierno
Consejería de Medio Ambiente, Ordenación del Territorio y Sostenibilidad
Consejería de Educación, Juventud y Deporte
Consejería de Educación e Investigación
Consejería de Economía, Hacienda y Empleo
Vicepresidencia, Consejería de Deportes, Transparencia y Portavocía de Gobierno
Consejería

3.2 Entrada finalizadas con éxito

Observaciones sobre cómo varía la tasa de éxito de las entradas según variable categórica.

TIPO

1. La tasa de **éxito** para los **Agradecimiento es muy baja**. ¿Por qué? La tasa de descarte para "Cancelada", "Enviada a Otra Administración" y "Rechazada por la SGAC" es similar para Agradecimiento –> No se puede trazar el origen de esta baja tasa. No se encuentra explicación, solo se pueden hacer conjeturas acerca de un uso irónico del Agradecimiento.

1.1. El Agradecimiento **se produce más** que Queja y Sugerencia en **PRESENCIAL**, y **menos** en **LLAMADAS 012**. –> Se llama menos para agradecer y se hace más en persona.

1.2. El Agradecimiento tiene **éxito por encima** de la media en **INTERNET**, y muy **por debajo** en **LLAMADAS 012**. –> El Agradecimiento finaliza mejor por Internet que por teléfono.

TEMA

2. Mayor tasa de Envío a Otra Administración en **Transportes** –> **Posibles entradas erróneas** debido a confusión del ciudadano acerca de quién tiene las competencias de **RENFE Cercanías**. El consorcio de transportes (consejería) tiene la tasa más baja de descartes, pero el Transporte (tema) tiene la tasa de descarte más alta –> Parece contradictorio: ¿El agente que lo resuelve (consejería) no coincide en la clasificación realizada por tema?

3. **Políticas sociales y Transportes** tenían el 1er y 3er número de entradas, mientras que son los dos temas con **menor tasa de éxito** –> Parece que hay un **problema real** en Políticas sociales.

4. Políticas Sociales (tema) parece haber recogido durante años las quejas de Sanidad, hasta que esta consejería ha organizado su propio servico de recogida de quejas. Los otros 2 temas de uso masivo y directo del ciudadano de la CAM son: Educación y Transportes. –> **Habría que revisar temporalmente la actividad de Sanidad y Políticas Sociales**.

CANAL

5. Los **canales de entrada** presentan irregularidades:

–> **"INTERNET", "PRESENCIAL", "LLAMADAS 012"** son los canales que **acaparan** la práctica totalidad de las entradas.

–> 'Teléfono 010' era un antiguo servicio del Ayuntamiento de Madrid. **Inactivo** desde mediados de 2016.

–> 'OTROS' lleva **inactivo** desde mediados de 2016.

CONSEJERIA

6. La **Consejería de Sanidad tiene el menor porcentaje de éxito y el mayor porcentaje de Cancelada** –> Parece que ya no se recogen quejas de esta consejería por esta vía y tiene su propio cauce de recogida de quejas **(¿fueron estos datos el motivo para segregarla?)**. La última entrada de Sanidad se registro el 14 de septiembre de 2022.

7. Hay **varías Consejerías** que están por **debajo del 80% de éxito**. La **Consejería de Sanidad** es la que tiene una **tasa de éxito menor**. Su última entrada se registró el 14-09-

2022 –> Tal vez tiene que ver con el hecho de que hay varias entradas que tienen su propio servicio de recogida:

–> Asistencia sanitaria en un hospital o centro de salud. Desde 2005.

–> Conflictos entre consumidores y empresas.

–> Instituciones europeas

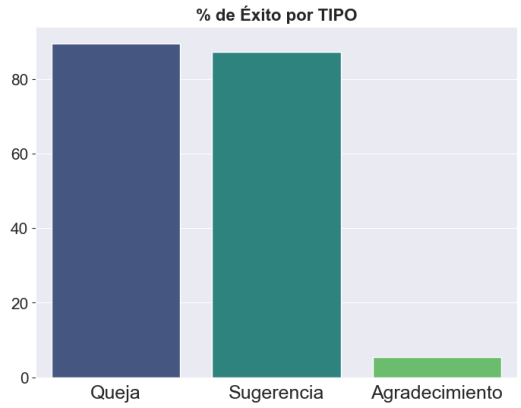
https://gestiona3.madrid.org/archivos/index.php/sugerencias-quejas

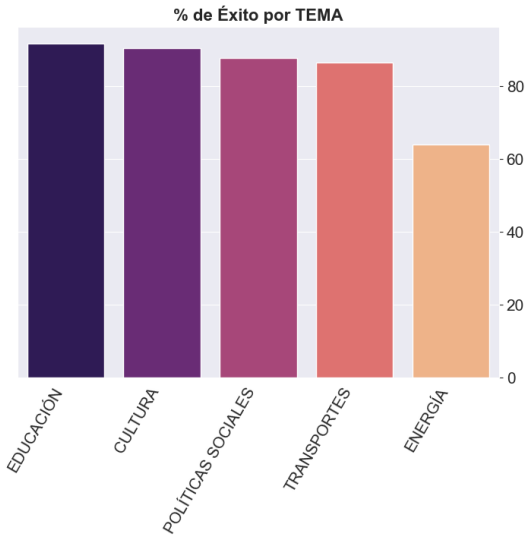
Otros aspectos

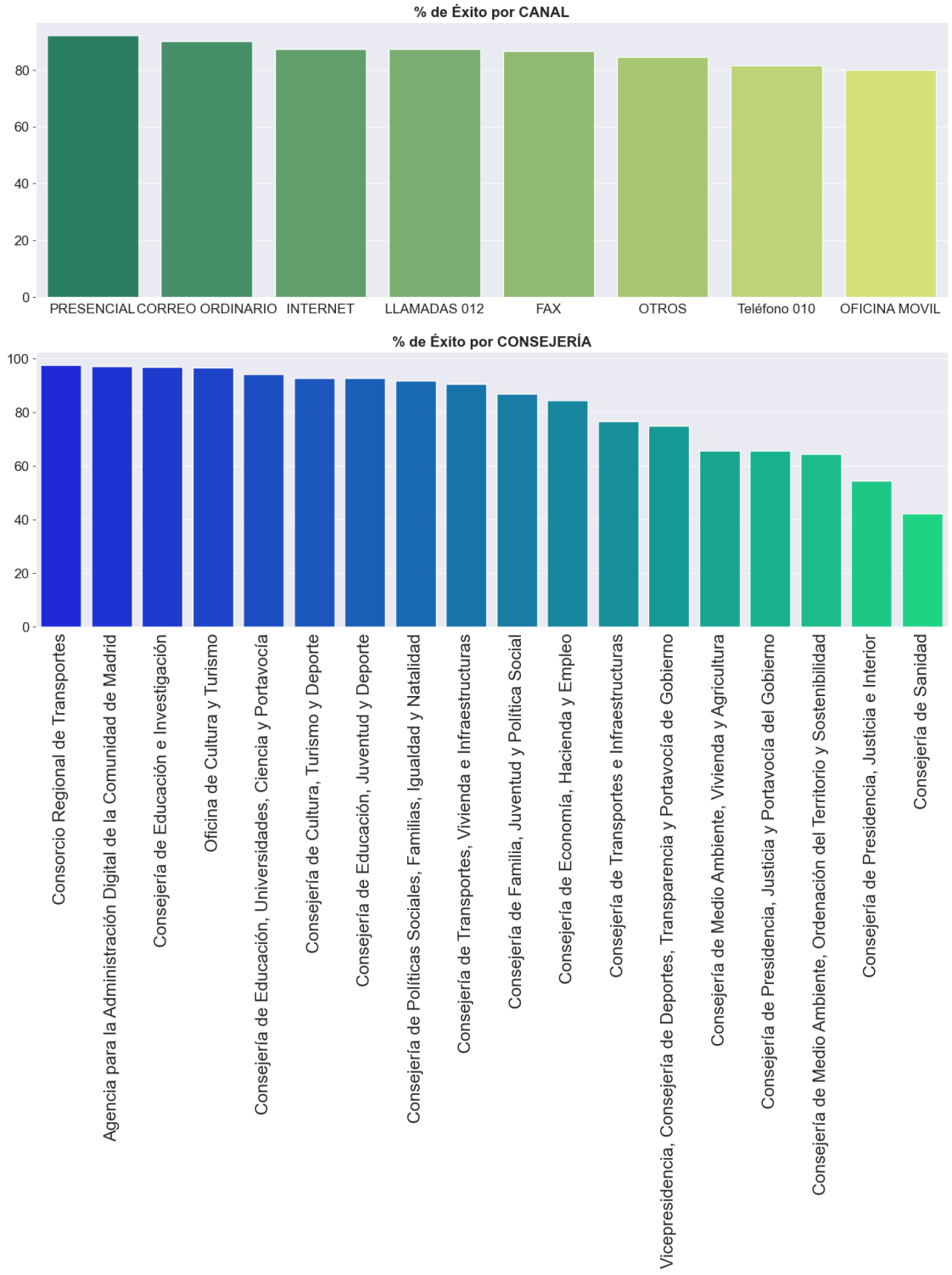
8. Hay correlación positiva entre Éxito y Rechazada por SGAC; y **correlación negativa entre Éxito y (Cancelada, Enviada a Otra Administración)**, en las variables consejería, unidad y canal de entrada –> Parece que al **aumentar la tasa de rechazo aumenta la eficiencia del gestor** (aumenta la tasa de éxito).

[62]:
```
# Gráficos tasa de ÉXITO

f, ax = plt.subplots(1, 2, figsize=(18,6))

# TIPO
sns.barplot(ax=ax[0], x=df.groupby('tipo')['exito'].mean().index,
            y=df.groupby('tipo')['exito'].mean().values*100,
            palette='viridis',
            order=df.groupby('tipo')['exito'].mean().
 ↪sort_values(ascending=False).index)
ax[0].set_title('% de Éxito por TIPO', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[0].set_xlabel('', loc=None, fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('', fontsize=15, fontdict=dict(weight='bold'))
ax[0].tick_params(axis='x', which='major', labelsize=18)
ax[0].set_xticks(ax[0].get_xticks(), ax[0].get_xticklabels())
ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[0].yaxis.tick_left()  # Donde se ponen las marcas de eje
# ax[1].legend(loc='upper right') # Si es múltiple: label=''

# TEMA
sns.barplot(ax=ax[1], x=df.groupby('tema')['exito'].mean().index,
            y=df.groupby('tema')['exito'].mean().values*100,
            palette='magma',
            order=df.groupby('tema')['exito'].mean().
 ↪sort_values(ascending=False).index)
ax[1].set_title('% de Éxito por TEMA', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[1].set_xlabel('', fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('', fontsize=15, fontdict=dict(weight='bold'))
ax[1].tick_params(axis='x', which='major', labelsize=15)
```

```
ax[1].set_xticks(ax[1].get_xticks(), ax[1].get_xticklabels(), rotation=60,␣
 ↪ha='right')
ax[1].tick_params(axis='y', which='major', labelsize=15)
ax[1].yaxis.tick_right();  # Donde se ponen las marcas de eje
# ax[1].legend(loc='upper right') # Si es múltiple: label=''


f, ax = plt.subplots(2, 1, figsize=(18,12))

# CANAL_ENTRADA
sns.barplot(ax=ax[0], x=df.groupby('canal_entrada')['exito'].mean().index,
            y=df.groupby('canal_entrada')['exito'].mean().values*100,
            palette='summer',
            order=df.groupby('canal_entrada')['exito'].mean().
 ↪sort_values(ascending=False).index)
ax[0].set_title('% de Éxito por CANAL', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[0].set_xlabel('', fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('', fontsize=15, fontdict=dict(weight='bold'))
ax[0].tick_params(axis='x', which='major', labelsize=15)
ax[0].set_xticks(ax[0].get_xticks(), ax[0].get_xticklabels())
ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[0].yaxis.tick_left()  # Donde se ponen las marcas de eje
# ax[1].legend(loc='upper right') # Si es múltiple: label=''

# CONSEJERIA
sns.barplot(ax=ax[1], x=df.groupby('consejeria')['exito'].mean().index,
            y=df.groupby('consejeria')['exito'].mean().values*100,
            palette='winter',
            order=df.groupby('consejeria')['exito'].mean().
 ↪sort_values(ascending=False).index) \
        .tick_params(axis='x', labelsize=15, labelrotation=90)
ax[1].set_title('% de Éxito por CONSEJERÍA', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[1].set_xlabel('', fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('', fontsize=15, fontdict=dict(weight='bold'))
ax[1].tick_params(axis='x', which='major', labelsize=19)
ax[1].set_xticks(ax[1].get_xticks(), ax[1].get_xticklabels(), rotation=90,␣
 ↪ha='center')
ax[1].tick_params(axis='y', which='major', labelsize=15)
ax[1].yaxis.tick_left()  # Donde se ponen las marcas de eje
# ax[1].legend(loc='upper right') # Si es múltiple: label=''

# plt.savefig("multimedia/pct_exito_por_variable.png")
```

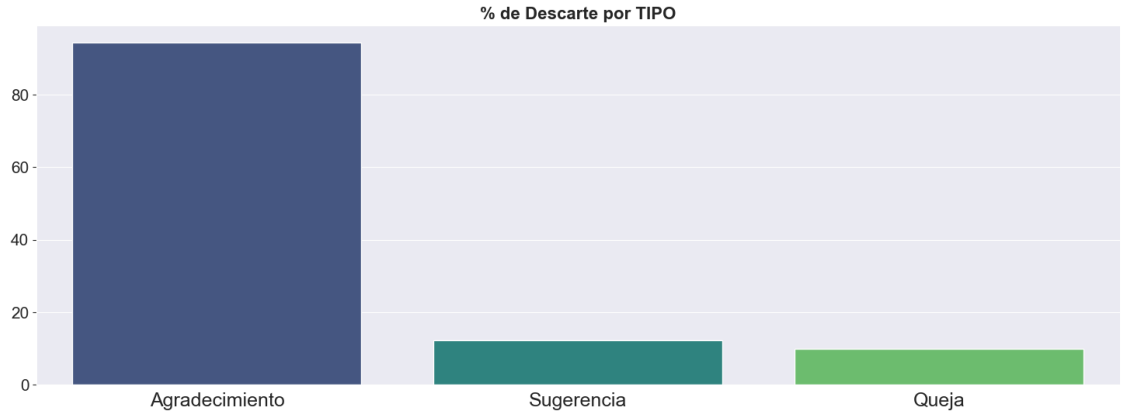

% de Éxito por TIPO
0
20
40
60
80
Queja
Sugerencia
Agradecimiento


% de Éxito por TEMA
0
20
40
60
80
EDUCACIÓN
CULTURA
POLÍTICAS SOCIALES
TRANSPORTES
ENERGÍA



[63]:
```
f, ax = plt.subplots(1, 1, figsize=(18,6))
```

```
# TIPO
sns.barplot(ax=ax, x=df.groupby('tipo')['descarte'].mean().index,
            y=df.groupby('tipo')['descarte'].mean().values*100,
            palette='viridis',
            order=df.groupby('tipo')['descarte'].mean().
 ↪sort_values(ascending=False).index)
ax.set_title('% de Descarte por TIPO', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax.set_xlabel('', loc=None, fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('', fontsize=15, fontdict=dict(weight='bold'))
ax.tick_params(axis='x', which='major', labelsize=18)
ax.set_xticks(ax.get_xticks(), ax.get_xticklabels())
ax.tick_params(axis='y', which='major', labelsize=15)
ax.yaxis.tick_left();

# plt.savefig("multimedia/pct_descarte_tipo.png")
```



4 Diferencias en variables categóricas - Temporalidad

4.1 Tipo

- No se observa nada.

[64]:
```
df_compo = mf.componentes_fecha(df, "fecha_entrada")
df_compo.drop(['hora', 'minuto', 'segundo', 'trimestre'], axis=1, inplace=True)
```

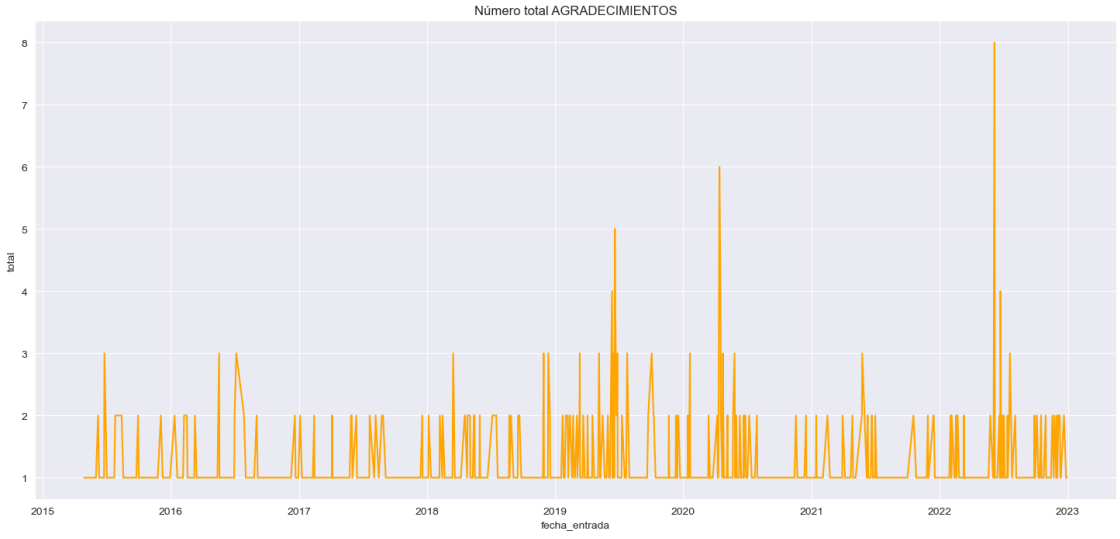
[65]:
```
df_asq = df_compo.reset_index().groupby(['fecha_entrada', 'tipo']).
 ↪agg(descarte=('descarte', 'sum'),
                           exito=('exito', 'sum'),
                           proceso=('proceso', 'sum'),
                           terminada=('terminada', 'sum'),
                           total=('tipo', 'count'))
```

[66]:
```
df_asq.reset_index(inplace=True)
```

[67]:
```
df_asq['terminada'] = df_asq.exito + df_asq.descarte
df_asq['exito_pct'] = 100*df_asq.exito / df_asq.terminada
df_asq['descarte_pct'] = 100*df_asq.descarte / df_asq.terminada
df_asq['terminada_pct'] = 100*df_asq.terminada / (df_asq.terminada + df_asq.
 ↪proceso)
df_asq['proceso_pct'] = 100*df_asq.proceso / (df_asq.terminada + df_asq.proceso)
df_asq['exitoT_pct'] = 100*df_asq.exito / (df_asq.terminada + df_asq.proceso)
```

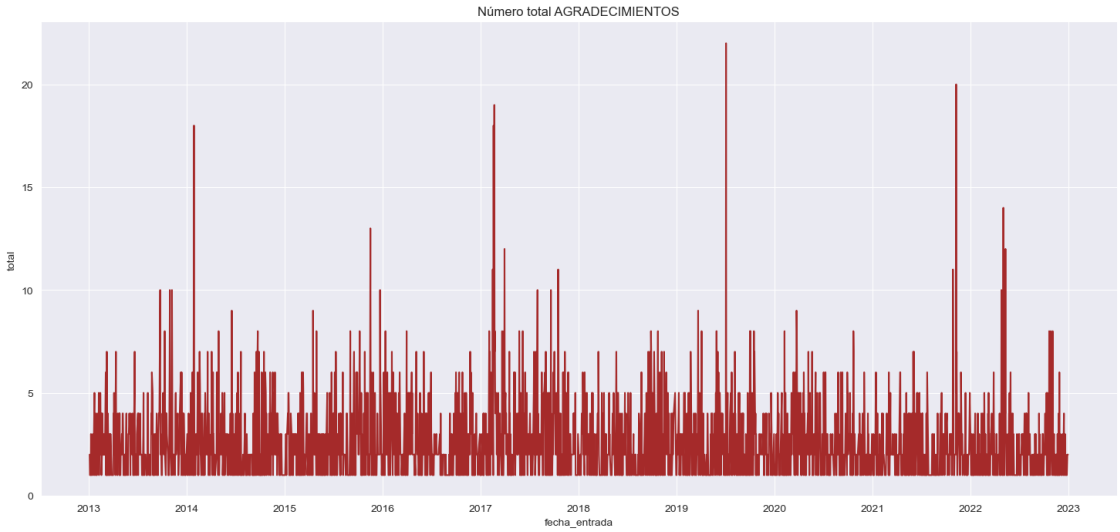
[68]:
```
# Agradecimiento
dfa = df_asq[df_asq.tipo == 'Agradecimiento'].set_index('fecha_entrada')

# Sugerencia
dfs = df_asq[df_asq.tipo == 'Sugerencia'].set_index('fecha_entrada')

# Queja
dfq = df_asq[df_asq.tipo == 'Queja'].set_index('fecha_entrada')
```

[69]:
```
# Gráfico Agradecimiento
f, ax = plt.subplots(1,1, figsize=(18,8))

sns.lineplot(ax=ax, data=dfa, x=dfa.index, y=dfa.total, color='orange') \
            .set_title(label='Número total AGRADECIMIENTOS', loc='center');
```

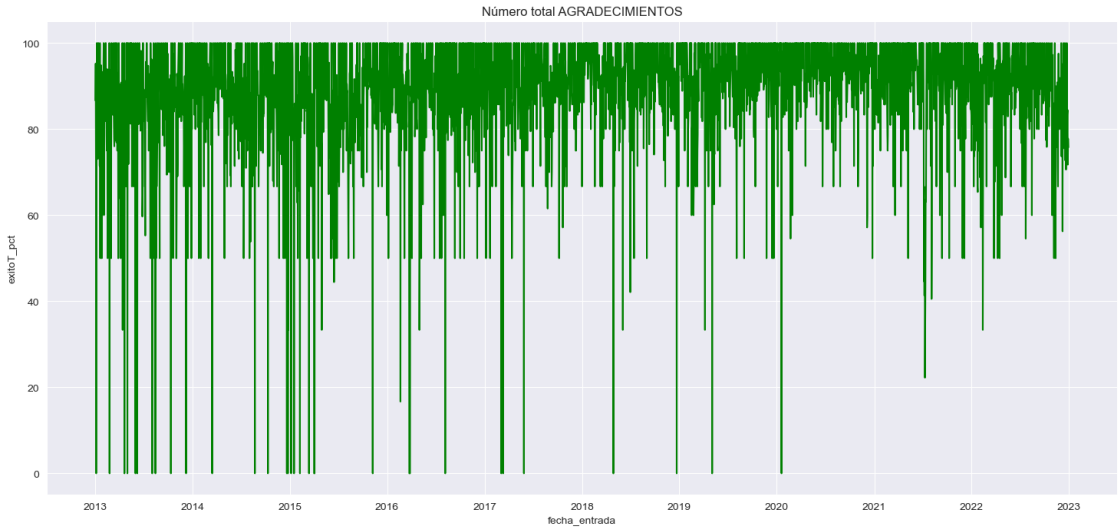


[70]:
```
# Sugerencia

f, ax = plt.subplots(1,1, figsize=(18,8))
```

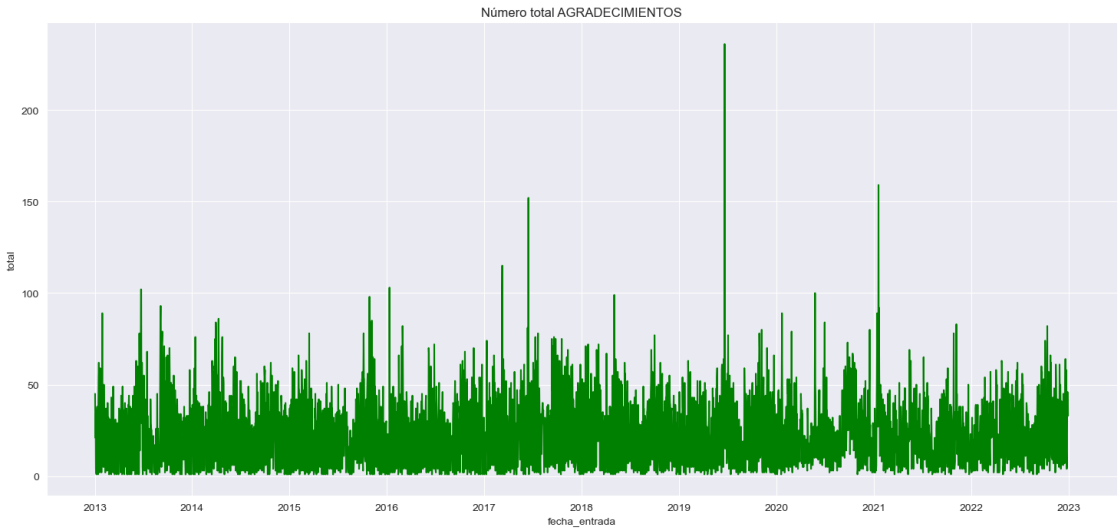
```
sns.lineplot(ax=ax, data=dfs, x=dfs.index, y=dfs.total, color='brown') \
            .set_title(label='Número total AGRADECIMIENTOS', loc='center');
```



[71]:
```
# Queja

f, ax = plt.subplots(1,1, figsize=(18,8))

sns.lineplot(ax=ax, data=dfq, x=dfq.index, y=dfq.exitoT_pct, color='green') \
            .set_title(label='Número total AGRADECIMIENTOS', loc='center');
```



[72]:
```
f, ax = plt.subplots(1,1, figsize=(18,8))

sns.lineplot(ax=ax, data=dfq, x=dfq.index, y=dfq.total, color='green') \
            .set_title(label='Número total AGRADECIMIENTOS', loc='center');
```



4.1.1 Mensual

Antes se ha visto el comportamiento diario, ahora vemos el comportamiento mensual.

- **Queja**. Tasas mejoran y número total de quejas estable o algo creciente.
- **Sugerencia**. Tasas mejoran y el número total estable o con ligero aumento.
- **Agradecimiento**. Muy pocos datos. La entrada de agradecimientos tiende a aumentar a lo largo del tiempo.

[73]:
```
df_tipo = df_compo.reset_index().groupby(['fecha_entrada', 'tipo']).
 ↪agg(descarte=('descarte', 'sum'),
                           exito=('exito', 'sum'),
                           proceso=('proceso', 'sum'),
                           terminada=('terminada', 'sum'),
                           total=('tipo', 'count'))
```

[74]:
```
df_tipo = df_tipo.reset_index().set_index('fecha_entrada').groupby('tipo').
 ↪resample('M').agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('total', 'sum'))
```

[75]:
```
df_tipo = df_tipo.reset_index().set_index('fecha_entrada')
```

[76]:
```
df_tipo['terminada'] = df_tipo.exito + df_tipo.descarte
df_tipo['exito_pct'] = 100*df_tipo.exito / df_tipo.terminada
df_tipo['descarte_pct'] = 100*df_tipo.descarte / df_tipo.terminada
df_tipo['terminada_pct'] = 100*df_tipo.terminada / (df_tipo.terminada + df_tipo.
 ↪proceso)
df_tipo['proceso_pct'] = 100*df_tipo.proceso / (df_tipo.terminada + df_tipo.
 ↪proceso)
df_tipo['exitoT_pct'] = 100*df_tipo.exito / (df_tipo.terminada + df_tipo.
 ↪proceso)
```

[77]:
```
# Queja
df_q_mes = df_tipo[df_tipo.tipo == 'Queja']

# Sugerencia
df_s_mes = df_tipo[df_tipo.tipo == 'Sugerencia']

# Agradecimiento
df_a_mes = df_tipo[df_tipo.tipo == 'Agradecimiento']
```

[78]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_q_mes, x=df_q_mes.index, y=df_q_mes.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO en QUEJA', loc='center')

sns.lineplot(ax=ax[1], data=df_q_mes, x=df_q_mes.index, y=df_q_mes.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE en QUEJA',␣
 ↪loc='center')

sns.lineplot(ax=ax[2], data=df_q_mes, x=df_q_mes.index, y=df_q_mes.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en QUEJA', loc='center');

plt.savefig("multimedia/pct_exito_por_tipo.png")
```

[79]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_s_mes, x=df_s_mes.index, y=df_s_mes.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO en SUGERENCIA',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=df_s_mes, x=df_s_mes.index, y=df_s_mes.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE en SUGERENCIA',␣
 ↪loc='center')

sns.lineplot(ax=ax[2], data=df_s_mes, x=df_s_mes.index, y=df_s_mes.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en SUGERENCIA',␣
 ↪loc='center');
```

[80]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_a_mes, x=df_a_mes.index, y=df_a_mes.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO en AGRADECIMIENTO',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=df_a_mes, x=df_a_mes.index, y=df_a_mes.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE en␣
 ↪AGRADECIMIENTO', loc='center')

sns.lineplot(ax=ax[2], data=df_a_mes, x=df_a_mes.index, y=df_a_mes.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en AGRADECIMIENTO',␣
 ↪loc='center');
```

4.1.2 Estacionalidad Año

- Diferencial tasa de éxito de **Agradecimiento** durante todo el año. Máximo en octubre.
- **Sugerencia** tiene mayor tasa de éxito que **Queja** solo en julio y agosto (verano).

[81]:
```
df_tipo_est = df_compo.reset_index().groupby(['mes', 'tipo']).agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('tipo', 'count'))
```

[82]:
```
df_tipo_est = df_tipo_est.reset_index().set_index('mes')
```

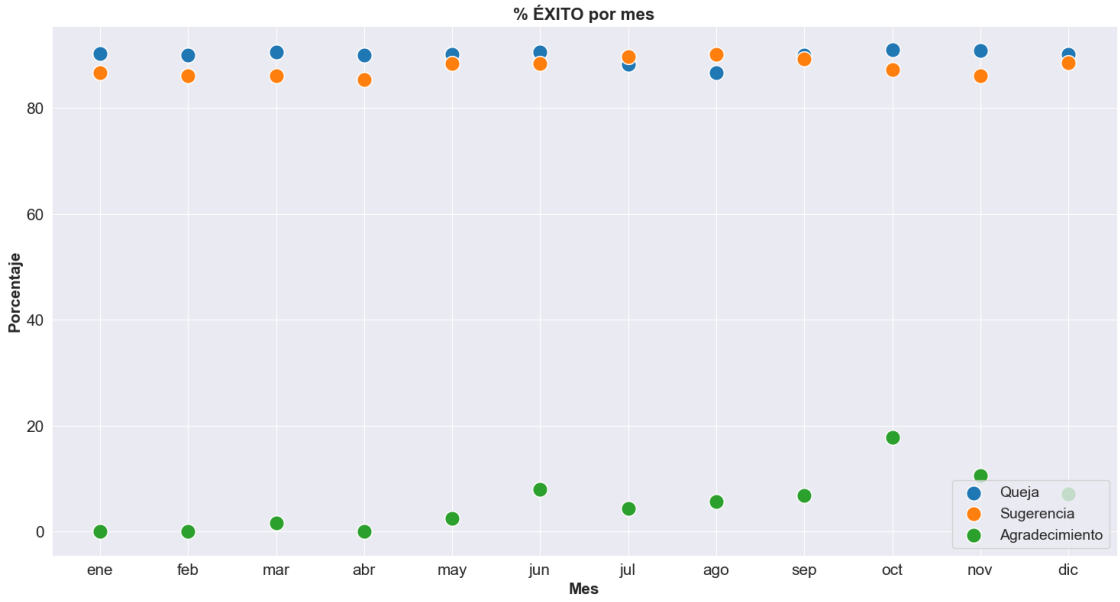
[83]:
```
df_tipo_est['terminada'] = df_tipo_est.exito + df_tipo_est.descarte
df_tipo_est['exito_pct'] = 100*df_tipo_est.exito / df_tipo_est.terminada
df_tipo_est['descarte_pct'] = 100*df_tipo_est.descarte / df_tipo_est.terminada
df_tipo_est['terminada_pct'] = 100*df_tipo_est.terminada / (df_tipo_est.
 ↪terminada + df_tipo_est.proceso)
```

```
df_tipo_est['proceso_pct'] = 100*df_tipo_est.proceso / (df_tipo_est.terminada +␣
 ↪df_tipo_est.proceso)
df_tipo_est['exitoT_pct'] = 100*df_tipo_est.exito / (df_tipo_est.terminada +␣
 ↪df_tipo_est.proceso)
```

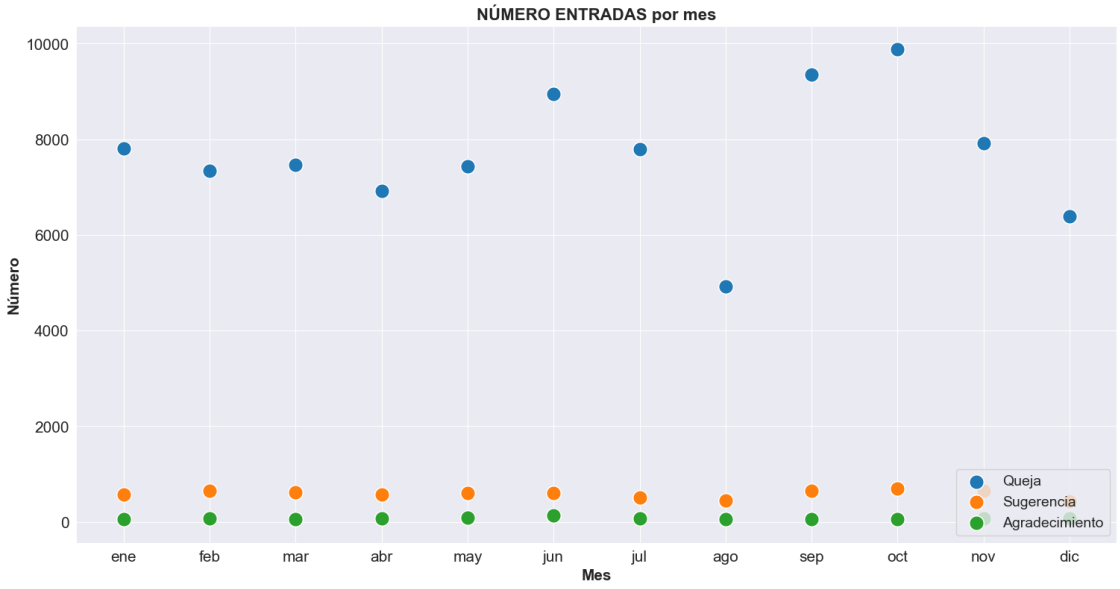
[84]:
```
# Queja
df_q_est = df_tipo_est[df_tipo_est.tipo == 'Queja']

# Sugerencia
df_s_est = df_tipo_est[df_tipo_est.tipo == 'Sugerencia']

# Agradecimiento
df_a_est = df_tipo_est[df_tipo_est.tipo == 'Agradecimiento']
```

GRÁFICOS CONJUNTOS

[85]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_q_est.index, y=df_q_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_s_est.index, y=df_s_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_a_est.index, y=df_a_est.exito_pct, s=200)

ax.set_title('% ÉXITO por mes',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_q_est.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='lower right', labels=['Queja', 'Sugerencia', 'Agradecimiento'],␣
 ↪fontsize=14);
```



[86]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_q_est.index, y=df_q_est.total, s=200)
sns.scatterplot(ax=ax, x=df_s_est.index, y=df_s_est.total, s=200)
sns.scatterplot(ax=ax, x=df_a_est.index, y=df_a_est.total, s=200)

ax.set_title('NÚMERO ENTRADAS por mes',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_q_est.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='lower right', labels=['Queja', 'Sugerencia', 'Agradecimiento'],␣
 ↪fontsize=14);
```



4.1.3 Estacionalidad Día Semana

- La tasa de éxito de **Queja** es superior todos los días de la semana.
- El número de entradas desciende según avanza la semana.

[87]:
```
df_tipo_ds = df_compo.reset_index().groupby(['dia_semana', 'tipo']).agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('tipo', 'count'))
```

[88]:
```
df_tipo_ds = df_tipo_ds.reset_index().set_index('dia_semana')
```

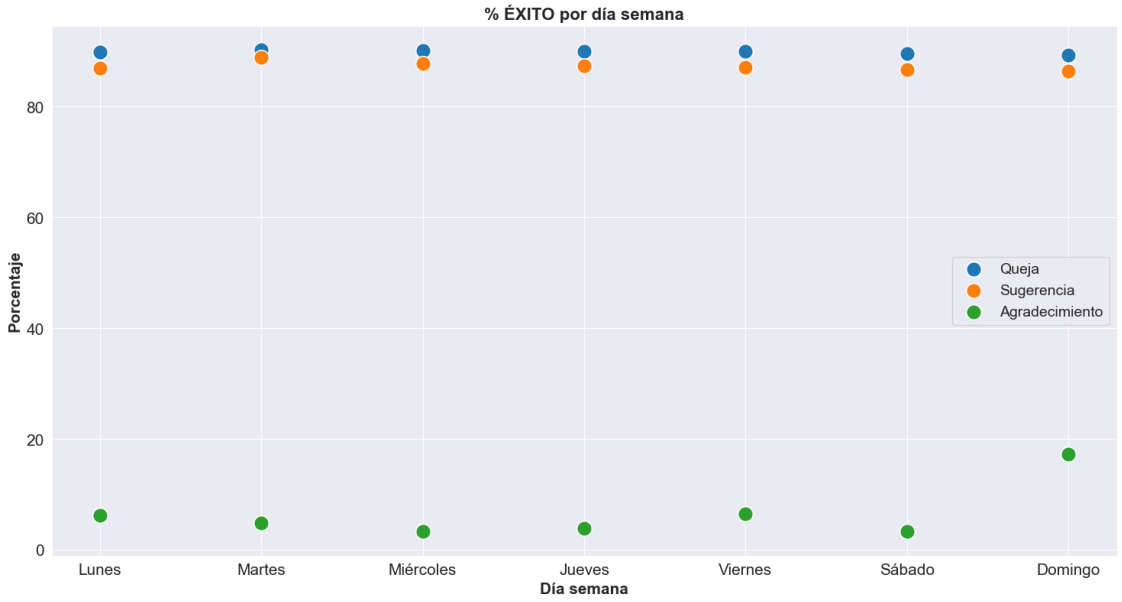
[89]:
```
df_tipo_ds['terminada'] = df_tipo_ds.exito + df_tipo_ds.descarte
df_tipo_ds['exito_pct'] = 100*df_tipo_ds.exito / df_tipo_ds.terminada
df_tipo_ds['descarte_pct'] = 100*df_tipo_ds.descarte / df_tipo_ds.terminada
df_tipo_ds['terminada_pct'] = 100*df_tipo_ds.terminada / (df_tipo_ds.terminada␣
 ↪+ df_tipo_ds.proceso)
df_tipo_ds['proceso_pct'] = 100*df_tipo_ds.proceso / (df_tipo_ds.terminada +␣
 ↪df_tipo_ds.proceso)
df_tipo_ds['exitoT_pct'] = 100*df_tipo_ds.exito / (df_tipo_ds.terminada +␣
 ↪df_tipo_ds.proceso)
```

[90]:
```
# Queja
df_q_ds = df_tipo_ds[df_tipo_ds.tipo == 'Queja']

# Sugerencia
```

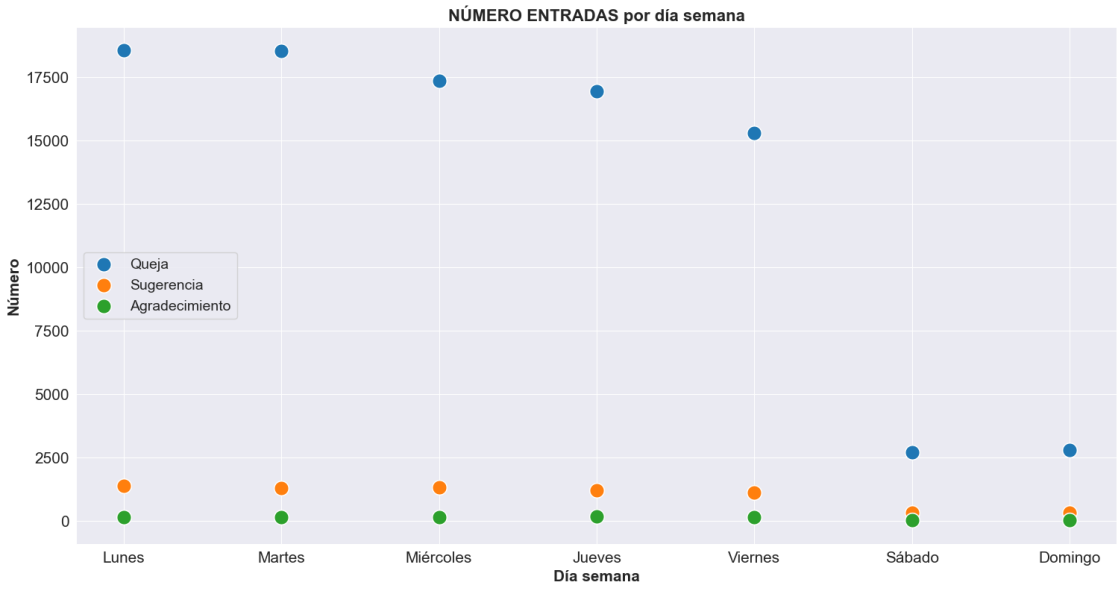
```
df_s_ds = df_tipo_ds[df_tipo_ds.tipo == 'Sugerencia']

# Agradecimiento
df_a_ds = df_tipo_ds[df_tipo_ds.tipo == 'Agradecimiento']
```

GRÁFICOS CONJUNTOS

[91]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_q_ds.index, y=df_q_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_s_ds.index, y=df_s_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_a_ds.index, y=df_a_ds.exito_pct, s=200)

ax.set_title('% ÉXITO por día semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_q_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='center right', labels=['Queja', 'Sugerencia', 'Agradecimiento'],␣
 ↪fontsize=14);
```



[92]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_q_ds.index, y=df_q_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_s_ds.index, y=df_s_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_a_ds.index, y=df_a_ds.total, s=200)

ax.set_title('NÚMERO ENTRADAS por día semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_q_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='center left', labels=['Queja', 'Sugerencia', 'Agradecimiento'],␣
 ↪fontsize=14);
```



4.2 Tema

4.2.1 Diario

- No se observa nada en términos diarios.

[93]:
```
df_tema = df_compo.reset_index().groupby(['fecha_entrada', 'tema']).
 ↪agg(descarte=('descarte', 'sum'),
                            exito=('exito', 'sum'),
                            proceso=('proceso', 'sum'),
                            terminada=('terminada', 'sum'),
                            total=('tipo', 'count'))
```

[94]:
```
df_tema_mes = df_tema.reset_index().set_index('fecha_entrada').groupby('tema').
 ↪resample('M').agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('total', 'sum'))
```

[95]:
```
df_tema_ano = df_tema.reset_index().set_index('fecha_entrada').groupby('tema').
 ↪resample('Y').agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('total', 'sum'))
```

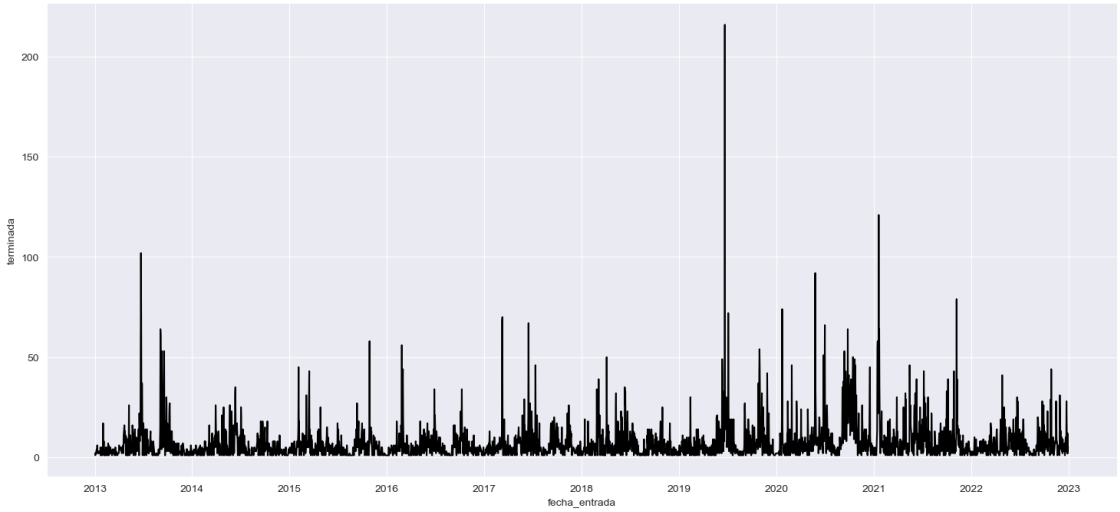
[96]:
```
df_tema.reset_index(inplace=True)
```

[97]:
```
df_tema['terminada'] = df_tema.exito + df_tema.descarte
df_tema['exito_pct'] = 100*df_tema.exito / df_tema.terminada
df_tema['descarte_pct'] = 100*df_tema.descarte / df_tema.terminada
df_tema['terminada_pct'] = 100*df_tema.terminada / (df_tema.terminada + df_tema.
 ↪proceso)
df_tema['proceso_pct'] = 100*df_tema.proceso / (df_tema.terminada + df_tema.
 ↪proceso)
df_tema['exitoT_pct'] = 100*df_tema.exito / (df_tema.terminada + df_tema.
 ↪proceso)
```

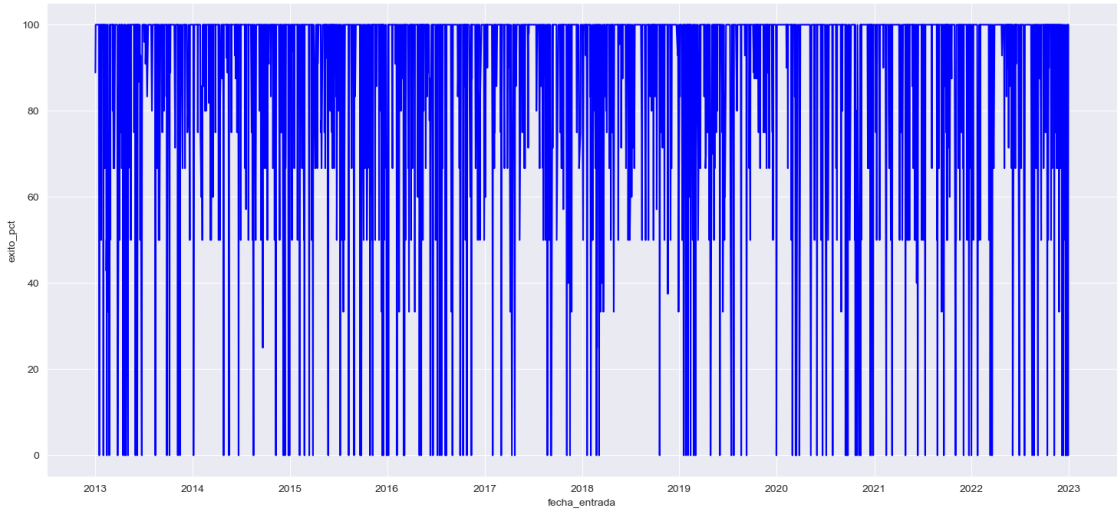
[98]:
```
# Educación
df_edu = df_tema[df_tema.tema == 'EDUCACIÓN'].set_index('fecha_entrada')

# Cultura
df_cul = df_tema[df_tema.tema == 'CULTURA'].set_index('fecha_entrada')

# Políticas sociales
df_pol = df_tema[df_tema.tema == 'POLÍTICAS SOCIALES'].
 ↪set_index('fecha_entrada')

# Transportes
df_tra = df_tema[df_tema.tema == 'TRANSPORTES'].set_index('fecha_entrada')
```

```
# Energía
df_ene = df_tema[df_tema.tema == 'ENERGÍA'].set_index('fecha_entrada')
```

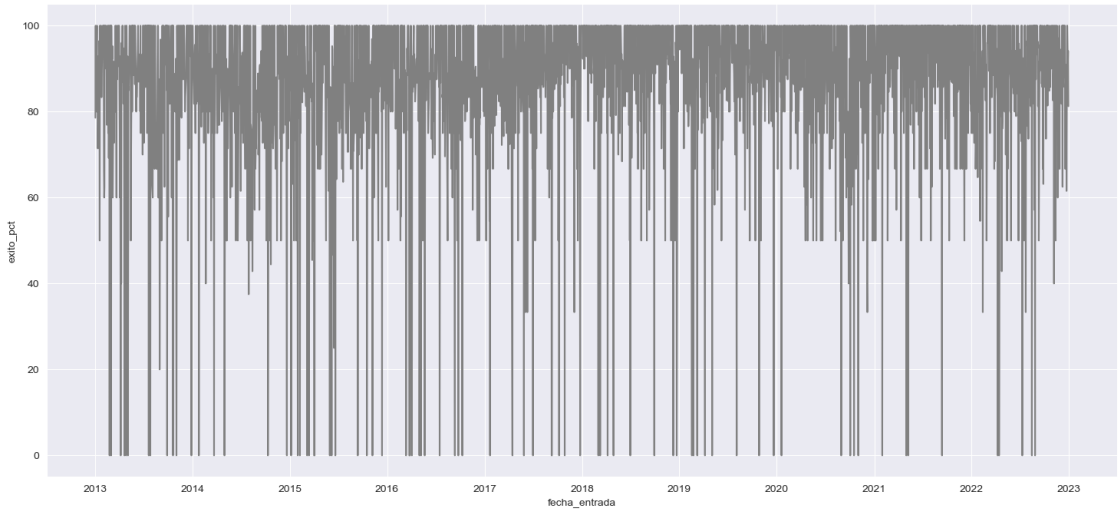
[99]:
```
f, ax = plt.subplots(1,1, figsize=(18,8))
sns.lineplot(ax=ax, data=df_edu, x=df_edu.index, y=df_edu.terminada,␣
 ↪color='black');
```



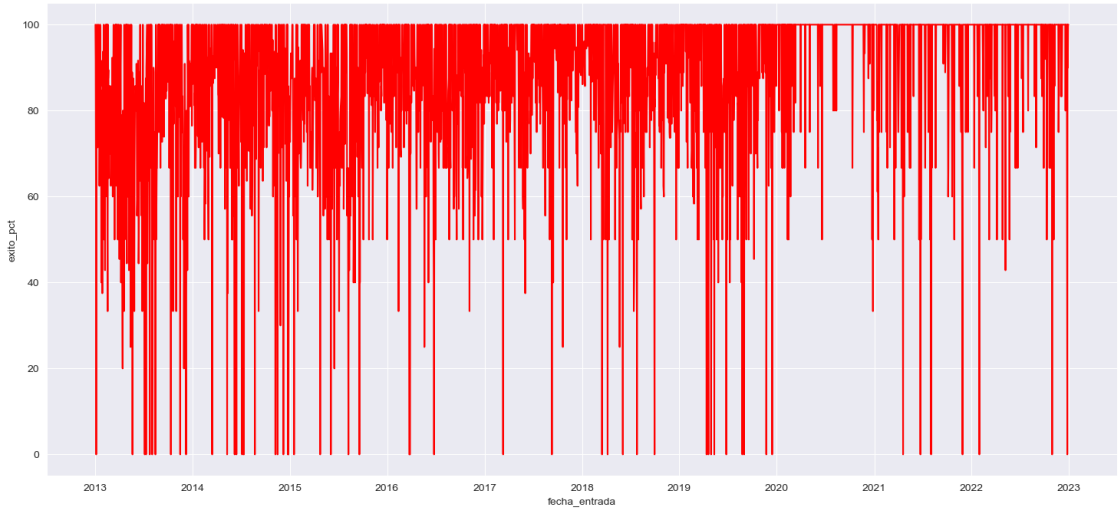
[100]:
```
f, ax = plt.subplots(1,1, figsize=(18,8))
sns.lineplot(ax=ax, data=df_cul, x=df_cul.index, y=df_cul.exito_pct,␣
 ↪color='blue');
```



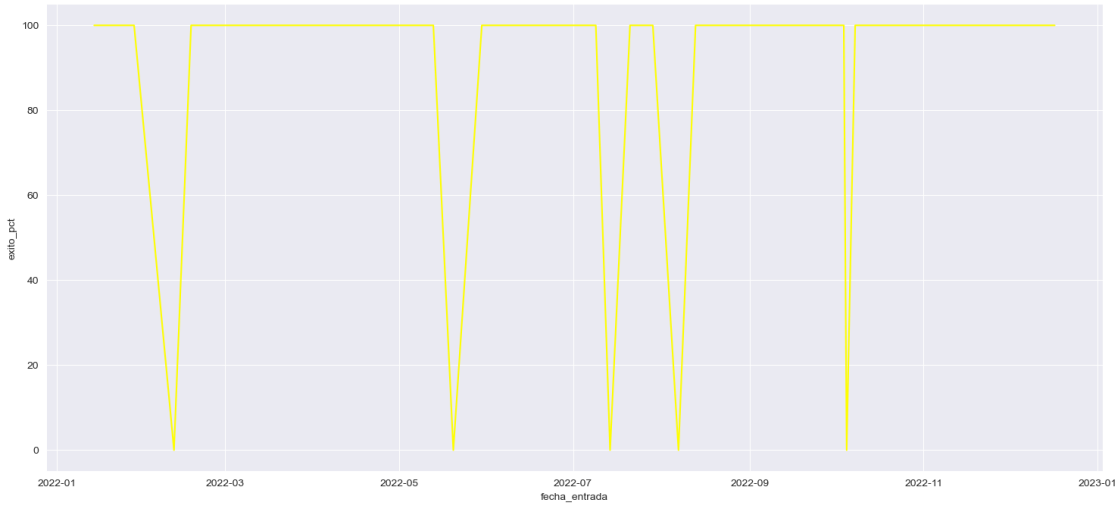
[101]:
```
f, ax = plt.subplots(1,1, figsize=(18,8))
sns.lineplot(ax=ax, data=df_pol, x=df_pol.index, y=df_pol.exito_pct,␣
 ↪color='grey');
```



[102]:
```
f, ax = plt.subplots(1,1, figsize=(18,8))
sns.lineplot(ax=ax, data=df_tra, x=df_tra.index, y=df_tra.exito_pct,␣
 ↪color='red');
```



[103]:
```
f, ax = plt.subplots(1,1, figsize=(18,8))
sns.lineplot(ax=ax, data=df_ene, x=df_ene.index, y=df_ene.exito_pct,␣
 ↪color='yellow');
```



4.2.2 Mensual

- Se examina: tasa de éxito, tasa de descarte y número de entradas a lo largo del tiempo.
- En Políticas Sociales y Educación el número de entradas aumenta a lo largo del tiempo.
- En Educación, Políticas Sociales y Transporte hay una mejora a lo largo del tiempo en ambas tasas.
- En Energía hay pocos datos.

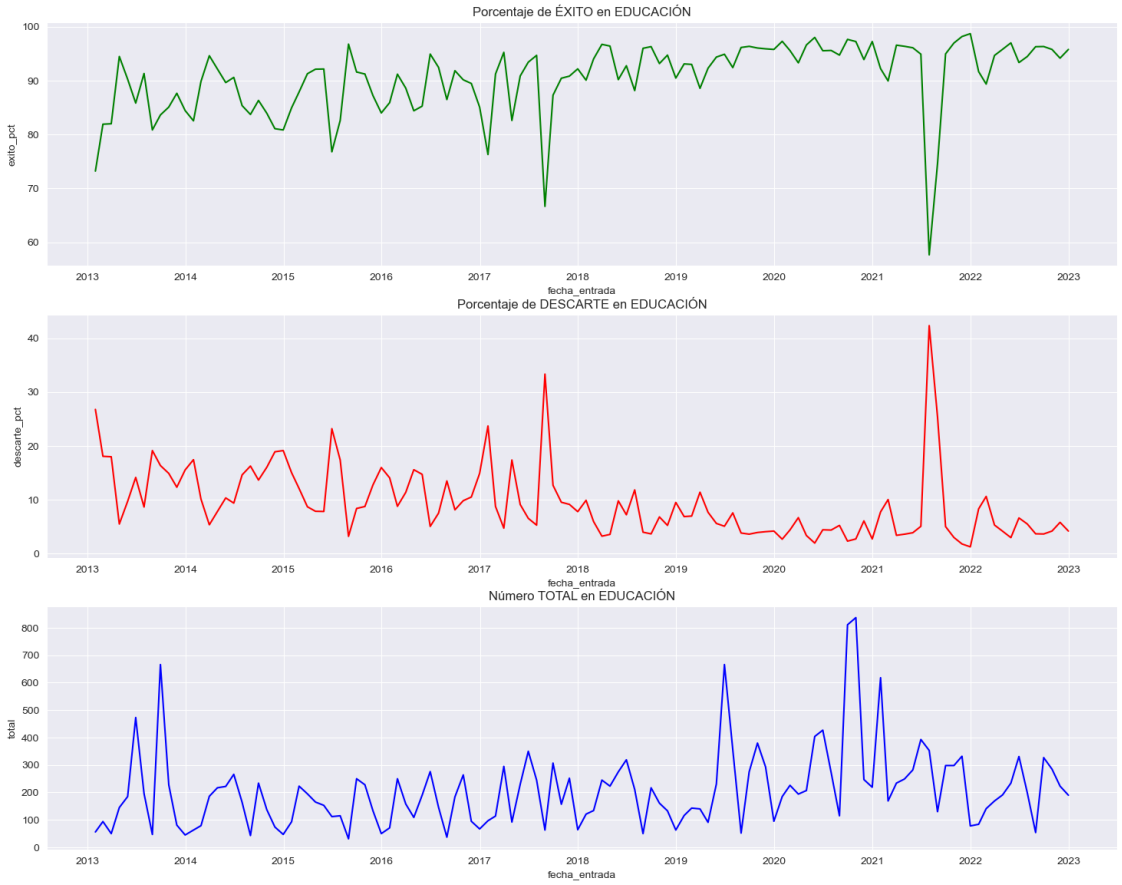
[104]:
```
df_tema_mes = df_tema_mes.reset_index().set_index('fecha_entrada')
```

[105]:
```
df_tema_mes['terminada'] = df_tema_mes.exito + df_tema_mes.descarte
df_tema_mes['exito_pct'] = 100*df_tema_mes.exito / df_tema_mes.terminada
df_tema_mes['descarte_pct'] = 100*df_tema_mes.descarte / df_tema_mes.terminada
df_tema_mes['terminada_pct'] = 100*df_tema_mes.terminada / (df_tema_mes.
 ↪terminada + df_tema_mes.proceso)
df_tema_mes['proceso_pct'] = 100*df_tema_mes.proceso / (df_tema_mes.terminada +␣
 ↪df_tema_mes.proceso)
df_tema_mes['exitoT_pct'] = 100*df_tema_mes.exito / (df_tema_mes.terminada +␣
 ↪df_tema_mes.proceso)
```

[106]:
```
# Educación
df_edu_mes = df_tema_mes[df_tema_mes.tema == 'EDUCACIÓN']

# Cultura
df_cul_mes = df_tema_mes[df_tema_mes.tema == 'CULTURA']

# Políticas sociales
df_pol_mes = df_tema_mes[df_tema_mes.tema == 'POLÍTICAS SOCIALES']

# Transportes
```

```
df_tra_mes = df_tema_mes[df_tema_mes.tema == 'TRANSPORTES']

# Energía
df_ene_mes = df_tema_mes[df_tema_mes.tema == 'ENERGÍA']
```

[107]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_edu_mes, x=df_edu_mes.index, y=df_edu_mes.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO en EDUCACIÓN',␣
 ↪loc='center')  # Exito total es igual.

sns.lineplot(ax=ax[1], data=df_edu_mes, x=df_edu_mes.index, y=df_edu_mes.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE en EDUCACIÓN',␣
 ↪loc='center')

sns.lineplot(ax=ax[2], data=df_edu_mes, x=df_edu_mes.index, y=df_edu_mes.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en EDUCACIÓN', loc='center');
```



- **Educación**. Mejora a lo largo del tiempo de la tasa de éxito y cae de la tasa de descarte. El número de entradas crece a lo largo del tiempo.
- Educación mejora su tasa de éxito sin, aparantemente, mejorar su tasa de reenvío (ver más abajo).

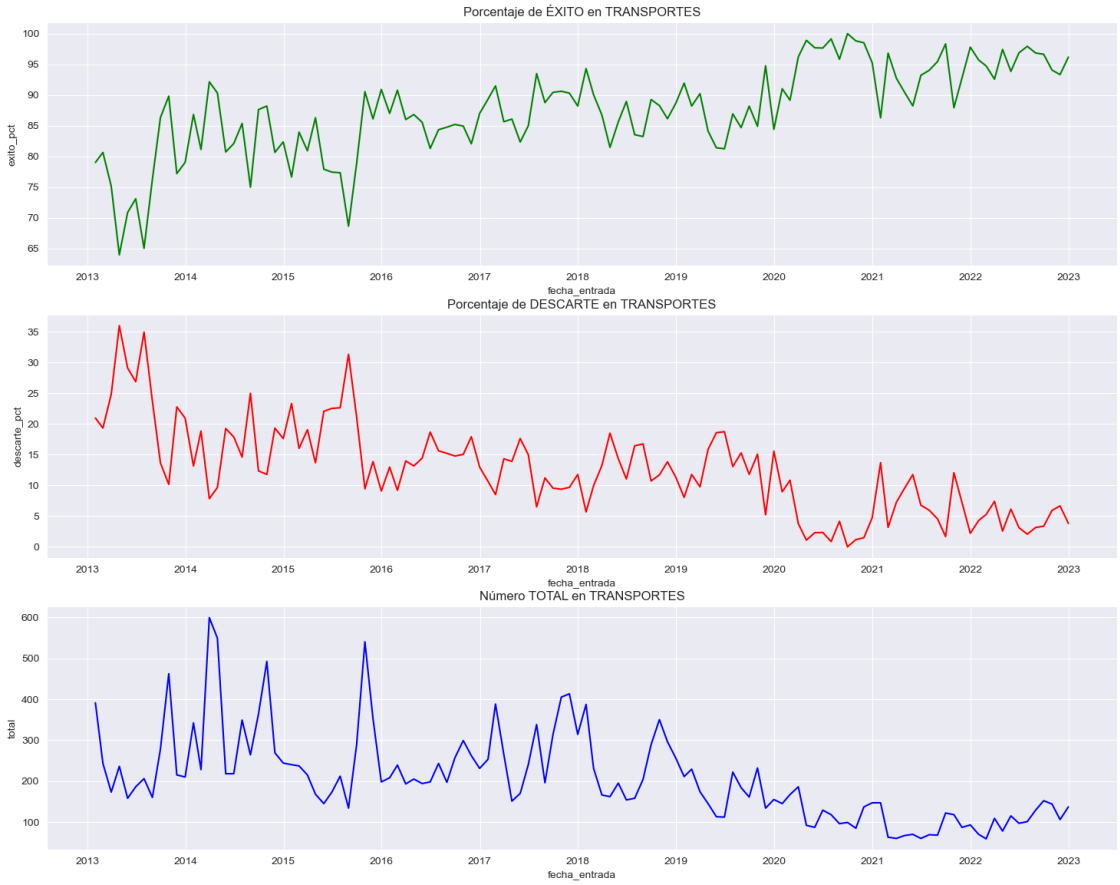
[108]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_cul_mes, x=df_cul_mes.index, y=df_cul_mes.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO en CULTURA',␣
 ↪loc='center')  # Exito total es igual.

sns.lineplot(ax=ax[1], data=df_cul_mes, x=df_cul_mes.index, y=df_cul_mes.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE en CULTURA',␣
 ↪loc='center')

sns.lineplot(ax=ax[2], data=df_cul_mes, x=df_cul_mes.index, y=df_cul_mes.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en CULTURA', loc='center');
```

- **Cultura**. Las tasas de éxito y descarte permanecen estables a lo largo del tiempo, excepto a finales de 2023. ¿Por qué? Entradas más o menos constantes, con caída abrupta en 2020.
- La tasa de reenvío aumenta, el número de entradas disminuye y el número de reenvíos aumenta. ¿Qué ocurre para este deterioro?

[109]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_pol_mes, x=df_pol_mes.index, y=df_pol_mes.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO en POLÍTICAS SOCIALES',␣
 ↪loc='center')  # Exito total es igual.

sns.lineplot(ax=ax[1], data=df_pol_mes, x=df_pol_mes.index, y=df_pol_mes.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE en POLÍTICAS␣
 ↪SOCIALES', loc='center')

sns.lineplot(ax=ax[2], data=df_pol_mes, x=df_pol_mes.index, y=df_pol_mes.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en POLÍTICAS SOCIALES',␣
 ↪loc='center');
```

- **Políticas Sociales**. Hay una mejora en ambas tasas, pero menos pronunciada que en educación. El número de entradas crece conforme pasa el tiempo (aprox. 23 a 30).
- Aparentemente comparte situación respecto a la tasa de reenvío con Educación.

[110]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_tra_mes, x=df_tra_mes.index, y=df_tra_mes.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO en TRANSPORTES',␣
 ↪loc='center')  # Exito total es igual.

sns.lineplot(ax=ax[1], data=df_tra_mes, x=df_tra_mes.index, y=df_tra_mes.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE en TRANSPORTES',␣
 ↪loc='center')

sns.lineplot(ax=ax[2], data=df_tra_mes, x=df_tra_mes.index, y=df_tra_mes.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en TRANSPORTES',␣
 ↪loc='center');
```



- **Transporte**. Mejora continua y sostenida en ambas tasas. El volumen de entradas es constante hasta 2019 y a partir de ahí disminuye, aunque repunta a partir de mitad de 2022.
- El número de reenviadas (ver más abajo) disminuye antes de que el número de entradas de Transporte empiece a bajar. Entradas de transporte: 2019, reenviadas (porcentaje y número): a partir de mitad de 2015.

[111]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_ene_mes, x=df_ene_mes.index, y=df_ene_mes.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO en ENERGÍA',␣
 ↪loc='center')  # Exito total es igual.

sns.lineplot(ax=ax[1], data=df_ene_mes, x=df_ene_mes.index, y=df_ene_mes.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE en ENERGÍA',␣
 ↪loc='center')

sns.lineplot(ax=ax[2], data=df_ene_mes, x=df_ene_mes.index, y=df_ene_mes.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en ENERGÍA', loc='center');
```

- **Energía**. Muy pocos datos. Entradas crecientes.

Caso especial: tasa de reenvío REENVÍO como TEMA

TASA

- **Educación**. Tasa de reenviadas más o menos constante con un pico notable en la segunda mitad de 2021. ¿Causa? Tal vez cuestiones sobre el reinicio de las clases tras la pandemia.
- **Cultura**. Tasa creciente de reenviadas. Ha hecho el camino inverso a Transportes, inicialmente tenía tasas parecidas a Educación y Políticas Sociales, pero con el tiempo ha incrementando su tasa de reenvío hasta ser el tema que tiene mayor tasa de reenvío. En 2020 registra un pico creciente. ¿Por qué?
- **Políticas Sociales**. Tasa oscilante, tal vez decreciente.
- **Transportes**. Tasa claramente decreciente de reenvíos, desde el 15-20% al 3-4%. Transportes tenía la tasa de reenvío más alta al princpio del histórico, y se ha ido aproximando a Educación y Políticas Sociales. En 2020 cae a mínimos la tasa, entendible durante el confinamiento.
- **Energía**. No hay muchos datos.

NÚMERO

- **Educación**. Se observa el pico en el número de reenviadas en la segunda mitad de 2021, tal vez consecuencia del reinicio del curso tras la pandemia.
- **Cultura**. Número creciente de reenvíos, pero a unos niveles muy bajos, entre 2-10.
- **Políticas Sociales**. Mantiene oscilación en torno a 7,5 reenvíos al mes.
- **Transportes**. Claramente hay notable descenso en el número de reenvíos, desde 40-50 hasta menos de 10 al mes.
- **Energía**. Pocos datos.

[112]:
```
df_compo['reenviada'] = np.where(df_compo.estado == 'Enviada a Otra␣
↪Administración', 1, 0)
```

[113]:
```
df_tte = df_compo.reset_index().groupby(['fecha_entrada', 'tema']).agg(
    reenviada=('reenviada', 'sum'),
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('tipo', 'count'))
```

[114]:
```
df_tte = df_tte.reset_index().set_index('fecha_entrada').groupby('tema').
↪resample('M').agg(
    reenviada=('reenviada', 'sum'),
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('total', 'sum'))
```

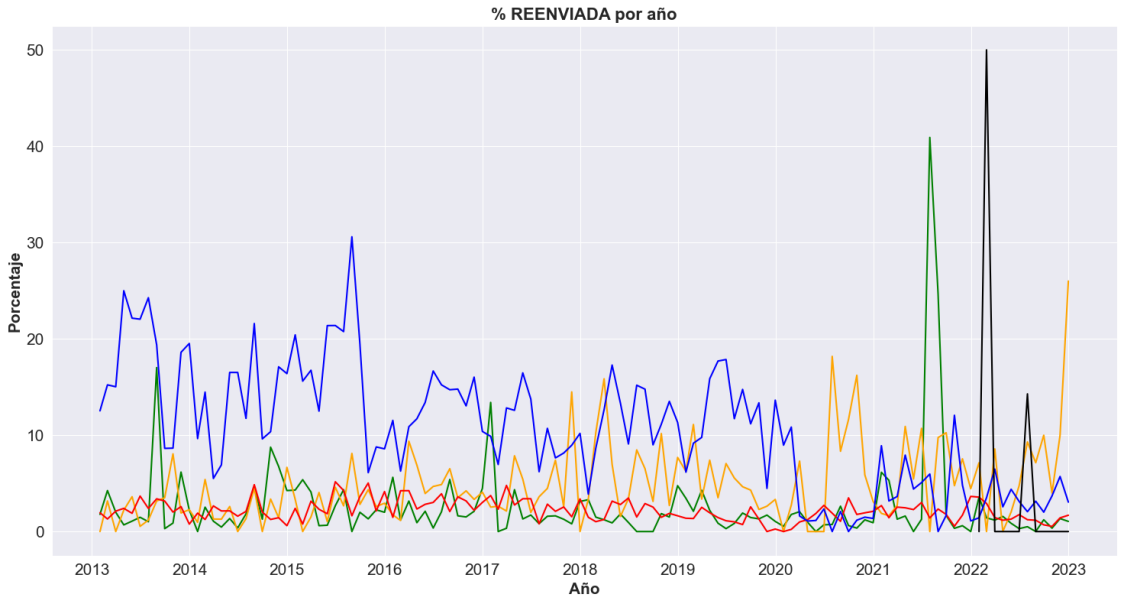
[115]:
```
df_tte = df_tte.reset_index().set_index('fecha_entrada')
```

[116]:
```
df_tte['terminada'] = df_tte.exito + df_tte.descarte
df_tte['exito_pct'] = 100*df_tte.exito / df_tte.terminada
df_tte['descarte_pct'] = 100*df_tte.descarte / df_tte.terminada
df_tte['terminada_pct'] = 100*df_tte.terminada / (df_tte.terminada + df_tte.
 ↪proceso)
df_tte['proceso_pct'] = 100*df_tte.proceso / (df_tte.terminada + df_tte.proceso)
df_tte['exitoT_pct'] = 100*df_tte.exito / (df_tte.terminada + df_tte.proceso)
df_tte['reenviada_pct'] = 100*df_tte.reenviada / df_tte.terminada
df_tte['reenviadaT_pct'] = 100*df_tte.reenviada / (df_tte.terminada + df_tte.
 ↪proceso)
```

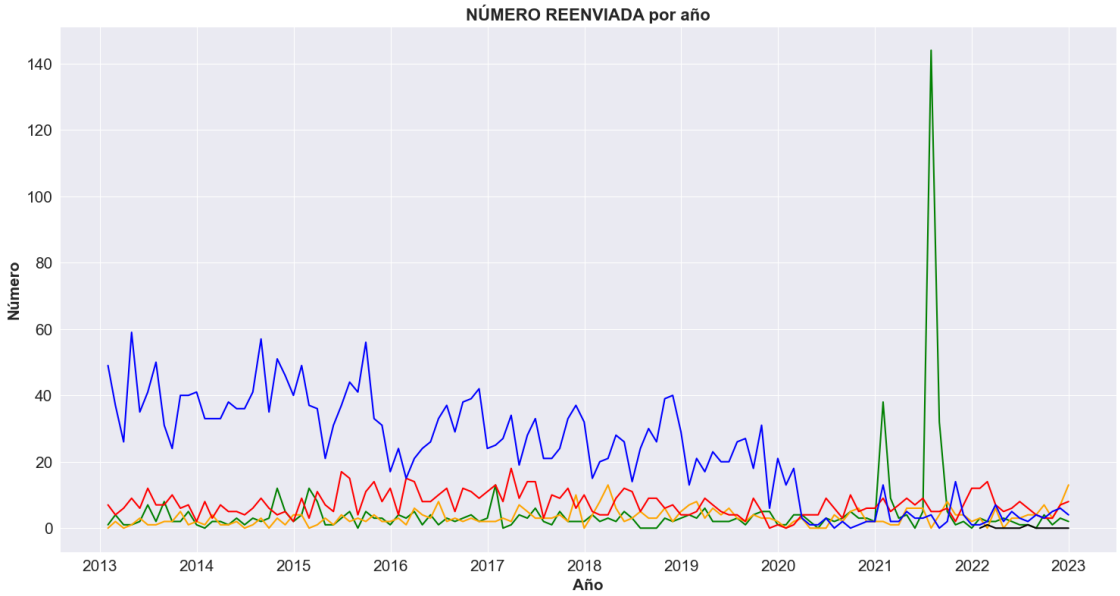
[117]:
```
# Educación
df_edu_env = df_tte[df_tte.tema == 'EDUCACIÓN']
# Cultura
df_cul_env = df_tte[df_tte.tema == 'CULTURA']
# Políticas sociales
df_pol_env = df_tte[df_tte.tema == 'POLÍTICAS SOCIALES']
# Transportes
df_tte_env = df_tte[df_tte.tema == 'TRANSPORTES']
# Energía
df_ene_env = df_tte[df_tte.tema == 'ENERGÍA']
```

[118]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=df_edu_env.index, y=df_edu_env.reenviada_pct,␣
 ↪color='green')
sns.lineplot(ax=ax, x=df_cul_env.index, y=df_cul_env.reenviada_pct,␣
 ↪color='orange')
sns.lineplot(ax=ax, x=df_pol_env.index, y=df_pol_env.reenviada_pct, color='red')
sns.lineplot(ax=ax, x=df_tte_env.index, y=df_tte_env.reenviada_pct,␣
 ↪color='blue')
sns.lineplot(ax=ax, x=df_ene_env.index, y=df_ene_env.reenviada_pct,␣
 ↪color='black')

ax.set_title('% REENVIADA por año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15);

# ax.legend(loc='upper left', labels=['Educación',
#                                     'Cultura',
```

```
#                                          'Políticas Sociales',
#                                          'Transportes',
#                                          'Energía'
#                                        ], fontsize=14);
```



[119]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=df_edu_env.index, y=df_edu_env.reenviada, color='green')
sns.lineplot(ax=ax, x=df_cul_env.index, y=df_cul_env.reenviada, color='orange')
sns.lineplot(ax=ax, x=df_pol_env.index, y=df_pol_env.reenviada, color='red')
sns.lineplot(ax=ax, x=df_tte_env.index, y=df_tte_env.reenviada, color='blue')
sns.lineplot(ax=ax, x=df_ene_env.index, y=df_ene_env.reenviada, color='black')

ax.set_title('NÚMERO REENVIADA por año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15);
```



4.2.3 Anual

- Revisado. No hay novedades.

4.2.4 Estacionalidad Año

[120]:
```
df_tema_est = df_compo.reset_index().groupby(['mes', 'tema']).agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('tipo', 'count'))
```

[121]:
```
df_tema_est = df_tema_est.reset_index().set_index('mes')
```

[122]:
```
df_tema_est['terminada'] = df_tema_est.exito + df_tema_est.descarte
df_tema_est['exito_pct'] = 100*df_tema_est.exito / df_tema_est.terminada
df_tema_est['descarte_pct'] = 100*df_tema_est.descarte / df_tema_est.terminada
df_tema_est['terminada_pct'] = 100*df_tema_est.terminada / (df_tema_est.
 ↪terminada + df_tema_est.proceso)
df_tema_est['proceso_pct'] = 100*df_tema_est.proceso / (df_tema_est.terminada +␣
 ↪df_tema_est.proceso)
df_tema_est['exitoT_pct'] = 100*df_tema_est.exito / (df_tema_est.terminada +␣
 ↪df_tema_est.proceso)
```

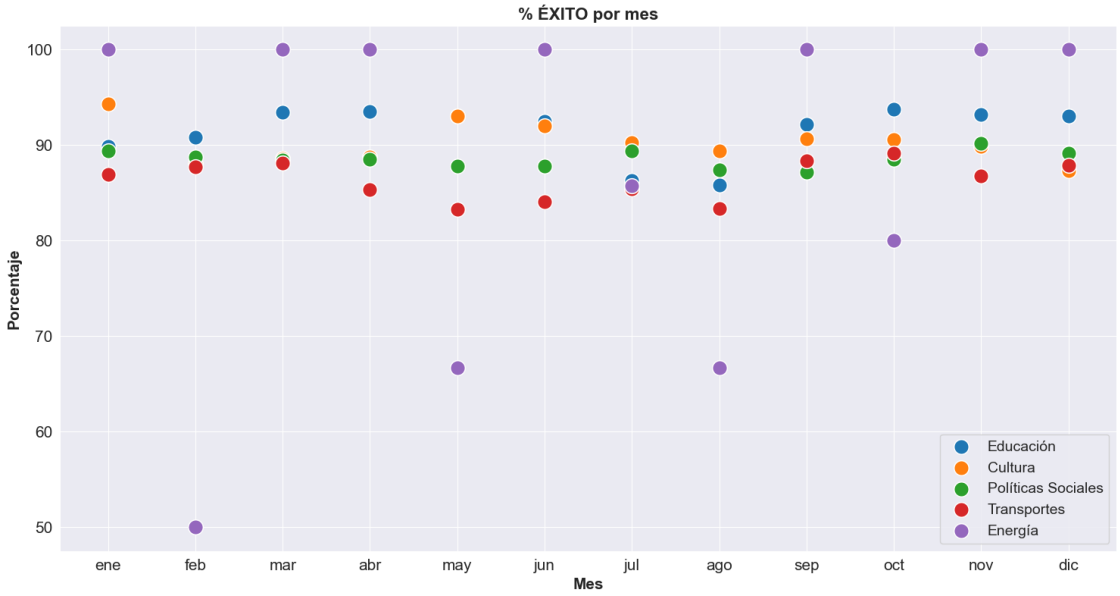
[123]:
```
# Educación
df_edu_est = df_tema_est[df_tema_est.tema == 'EDUCACIÓN']
```

```
# Cultura
df_cul_est = df_tema_est[df_tema_est.tema == 'CULTURA']

# Políticas Sociales
df_pol_est = df_tema_est[df_tema_est.tema == 'POLÍTICAS SOCIALES']

# Transportes
df_tra_est = df_tema_est[df_tema_est.tema == 'TRANSPORTES']

# Energía
df_ene_est = df_tema_est[df_tema_est.tema == 'ENERGÍA']
```

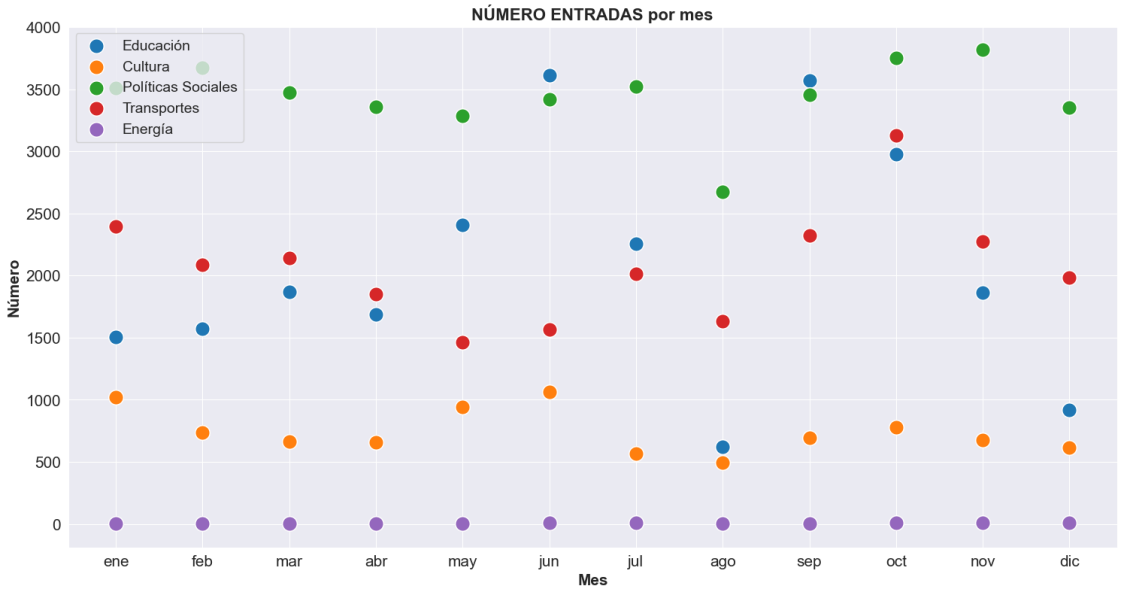
GRÁFICOS CONJUNTOS

- En general, tras el verano (septiembre y octubre) las tasas de éxito mejoran.
- **Energía** tiene tasas de éxito muy altas y muy bajas.
- **Educación** tiene un número de entradas que supera a las entradas de Política Social en junio y septiembre (final y principio de curso escolar).
- **Transportes** tiene un pico de entradas en octubre.

[124]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_edu_est.index, y=df_edu_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_cul_est.index, y=df_cul_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_pol_est.index, y=df_pol_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_tra_est.index, y=df_tra_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_ene_est.index, y=df_ene_est.exito_pct, s=200)

ax.set_title('% ÉXITO por mes',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_edu_est.index, labels=['ene', 'feb', 'mar',
                                              'abr', 'may', 'jun',
                                              'jul', 'ago', 'sep',
                                              'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='lower right', labels=['Educación', 'Cultura', 'Políticas␣
 ↪Sociales',
                                     'Transportes', 'Energía'], fontsize=14);
```



[125]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_edu_est.index, y=df_edu_est.total, s=200)
sns.scatterplot(ax=ax, x=df_cul_est.index, y=df_cul_est.total, s=200)
sns.scatterplot(ax=ax, x=df_pol_est.index, y=df_pol_est.total, s=200)
sns.scatterplot(ax=ax, x=df_tra_est.index, y=df_tra_est.total, s=200)
sns.scatterplot(ax=ax, x=df_ene_est.index, y=df_ene_est.total, s=200)

ax.set_title('NÚMERO ENTRADAS por mes',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_edu_est.index, labels=['ene', 'feb', 'mar',
                                              'abr', 'may', 'jun',
                                              'jul', 'ago', 'sep',
                                              'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper left', labels=['Educación', 'Cultura', 'Políticas␣
 ↪Sociales',
                                    'Transportes', 'Energía'], fontsize=14);
```



4.2.5 Estacionalidad Día Semana

- El porcentaje de éxito es muy similar en todos los temas.
- No hay diferencias en las tasas de éxito entre días de fin de semana y resto de días.
- La entradas disminuyen ha medida que avanza la semana.

[126]:
```
df_tema_ds = df_compo.reset_index().groupby(['dia_semana', 'tema']).agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('tipo', 'count'))
```

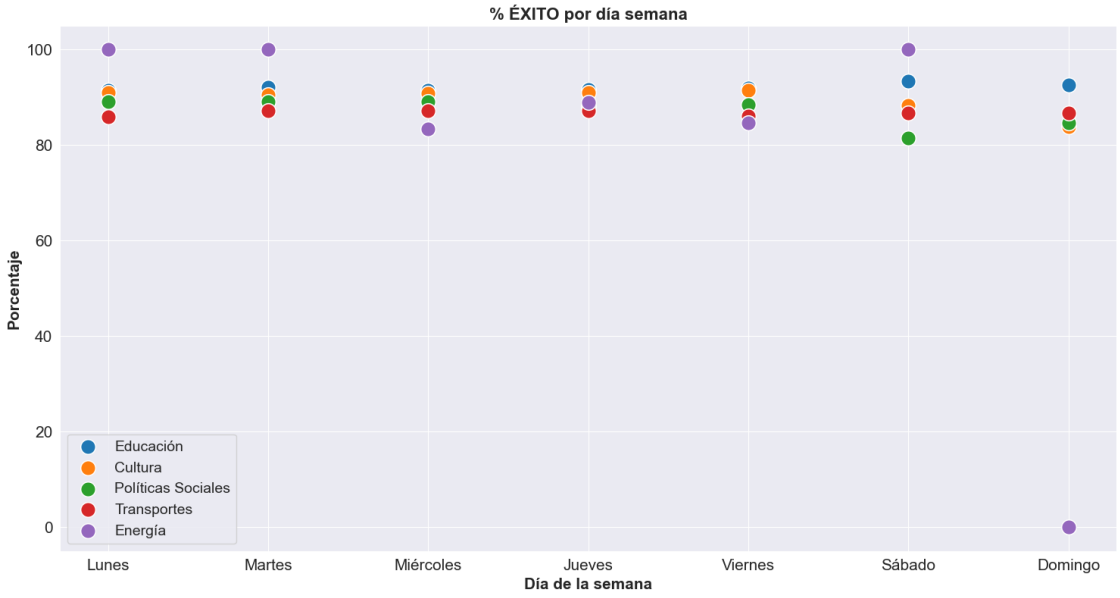
[127]:
```
df_tema_ds = df_tema_ds.reset_index().set_index('dia_semana')
```

[128]:
```
df_tema_ds['terminada'] = df_tema_ds.exito + df_tema_ds.descarte
df_tema_ds['exito_pct'] = 100*df_tema_ds.exito / df_tema_ds.terminada
df_tema_ds['descarte_pct'] = 100*df_tema_ds.descarte / df_tema_ds.terminada
df_tema_ds['terminada_pct'] = 100*df_tema_ds.terminada / (df_tema_ds.terminada␣
↪+ df_tema_ds.proceso)
df_tema_ds['proceso_pct'] = 100*df_tema_ds.proceso / (df_tema_ds.terminada +␣
↪df_tema_ds.proceso)
df_tema_ds['exitoT_pct'] = 100*df_tema_ds.exito / (df_tema_ds.terminada +␣
↪df_tema_ds.proceso)
```

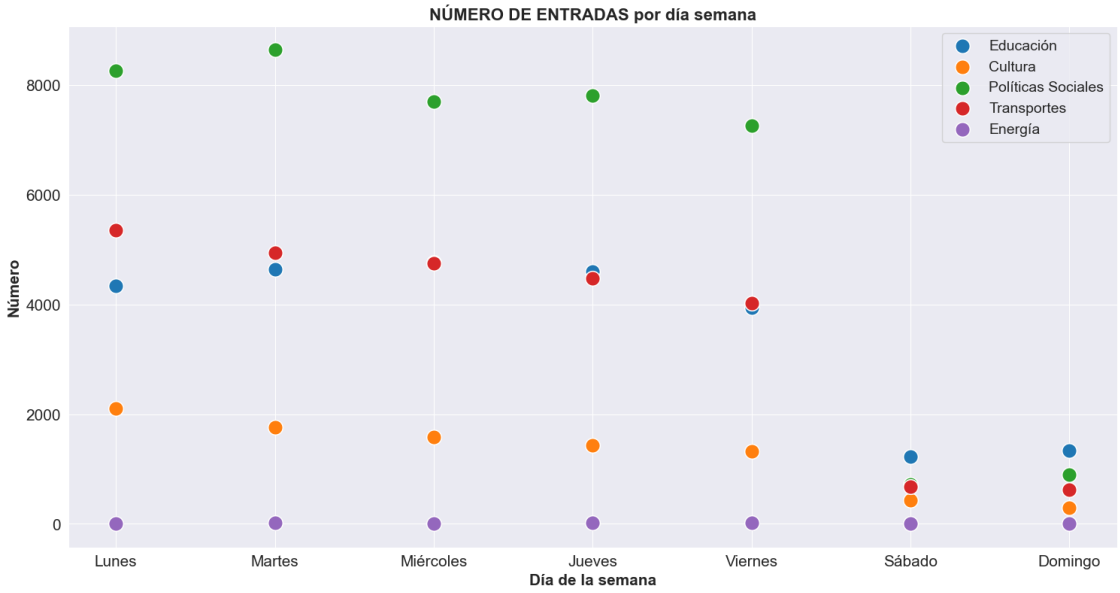
[129]:
```
# Educación
df_edu_ds = df_tema_ds[df_tema_ds.tema == 'EDUCACIÓN']

# Cultura
df_cul_ds = df_tema_ds[df_tema_ds.tema == 'CULTURA']

# Políticas Sociales
df_pol_ds = df_tema_ds[df_tema_ds.tema == 'POLÍTICAS SOCIALES']

# Transportes
df_tra_ds = df_tema_ds[df_tema_ds.tema == 'TRANSPORTES']

# Energía
df_ene_ds = df_tema_ds[df_tema_ds.tema == 'ENERGÍA']
```

GRÁFICOS CONJUNTOS

[130]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_edu_ds.index, y=df_edu_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_cul_ds.index, y=df_cul_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_pol_ds.index, y=df_pol_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_tra_ds.index, y=df_tra_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_ene_ds.index, y=df_ene_ds.exito_pct, s=200)

ax.set_title('% ÉXITO por día semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_edu_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='lower left', labels=['Educación', 'Cultura', 'Políticas␣
 ↪Sociales',
                                    'Transportes', 'Energía'], fontsize=14);
```



[131]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_edu_ds.index, y=df_edu_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_cul_ds.index, y=df_cul_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_pol_ds.index, y=df_pol_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_tra_ds.index, y=df_tra_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_ene_ds.index, y=df_ene_ds.total, s=200)

ax.set_title('NÚMERO DE ENTRADAS por día semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_edu_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Educación', 'Cultura', 'Políticas␣
 ↪Sociales',
                                     'Transportes', 'Energía'], fontsize=14);
```



4.3 Canal

4.3.1 Mensual

Solo se revisan: Internet, Presencial y Llamadas 012. El resto tienen muy pocos datos.

- **Internet**. Tasas mejorando. Número total se mantiene más o menos constante. No hay hundimiento en las entradas en 2020, aunque las tasas si que se alteran gravemente.
- **Presencial**. Tasas estables. En 2020 se percibe el fuerte empeoramiento de las tasas, pero también la caída en las entradas al ser un método presencial.
- **Llamadas 012**. Mejoran las tasas a lo largo del tiempo. El número de entradas tiene tendencia decreciente.

[132]:
```
df_canal = df_compo.reset_index().groupby(['fecha_entrada', 'canal_entrada']).
↪agg(descarte=('descarte', 'sum'),
                          exito=('exito', 'sum'),
                          proceso=('proceso', 'sum'),
                          terminada=('terminada', 'sum'),
                          total=('tipo', 'count'))
```

[133]:
```
df_canal = df_canal.reset_index().set_index('fecha_entrada').
↪groupby('canal_entrada').resample('M').agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('total', 'sum'))
```

[134]:
```
df_canal = df_canal.reset_index().set_index('fecha_entrada')
```

[135]:
```
df_canal
```

[135]:
```
                  canal_entrada  descarte  exito  proceso  terminada  total
fecha_entrada
2013-01-31     CORREO ORDINARIO         0      4        0          4      4
2013-02-28     CORREO ORDINARIO         0      3        0          3      3
2013-03-31     CORREO ORDINARIO         1      2        0          3      3
…                            …       …      …        …          …      …
2016-02-29         Teléfono 010         0      0        0          0      0
2016-03-31         Teléfono 010         0      3        0          3      3
2016-04-30         Teléfono 010         1      1        0          2      2

[760 rows x 6 columns]
```

[136]:
```
df_canal['terminada'] = df_canal.exito + df_canal.descarte
df_canal['exito_pct'] = 100*df_canal.exito / df_canal.terminada
df_canal['descarte_pct'] = 100*df_canal.descarte / df_canal.terminada
df_canal['terminada_pct'] = 100*df_canal.terminada / (df_canal.terminada +␣
 ↪df_canal.proceso)
df_canal['proceso_pct'] = 100*df_canal.proceso / (df_canal.terminada + df_canal.
 ↪proceso)
df_canal['exitoT_pct'] = 100*df_canal.exito / (df_canal.terminada + df_canal.
 ↪proceso)
```

[137]:
```
# Internet
df_int = df_canal[df_canal.canal_entrada == 'INTERNET']

# Presencial
df_pre = df_canal[df_canal.canal_entrada == 'PRESENCIAL']

# Llamadas 012
df_lla = df_canal[df_canal.canal_entrada == 'LLAMADAS 012']
```

[138]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_int, x=df_int.index, y=df_int.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO en INTERNET',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=df_int, x=df_int.index, y=df_int.descarte_pct,␣
 ↪color='red') \
                    .set_title(label='Porcentaje de DESCARTE en INTERNET',␣
 ↪loc='center')
```

```
sns.lineplot(ax=ax[2], data=df_int, x=df_int.index, y=df_int.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en INTERNET', loc='center');
```

[139]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_pre, x=df_pre.index, y=df_pre.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO en PRESENCIAL',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=df_pre, x=df_pre.index, y=df_pre.descarte_pct,␣
 ↪color='red') \
                    .set_title(label='Porcentaje de DESCARTE en PRESENCIAL',␣
 ↪loc='center')

sns.lineplot(ax=ax[2], data=df_pre, x=df_pre.index, y=df_pre.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en PRESENCIAL',␣
 ↪loc='center');
```

[140]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_lla, x=df_lla.index, y=df_lla.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO en LLAMADAS 012',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=df_lla, x=df_lla.index, y=df_lla.descarte_pct,␣
 ↪color='red') \
                    .set_title(label='Porcentaje de DESCARTE en LLAMADAS 012',␣
 ↪loc='center')

sns.lineplot(ax=ax[2], data=df_lla, x=df_lla.index, y=df_lla.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en LLAMADAS 012',␣
 ↪loc='center');
```

4.3.2 Estacionalidad Año

[141]:
```
df_canal_est = df_compo.reset_index().groupby(['mes', 'canal_entrada']).agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('tipo', 'count'))
```

[142]:
```
df_canal_est = df_canal_est.reset_index().set_index('mes')
```

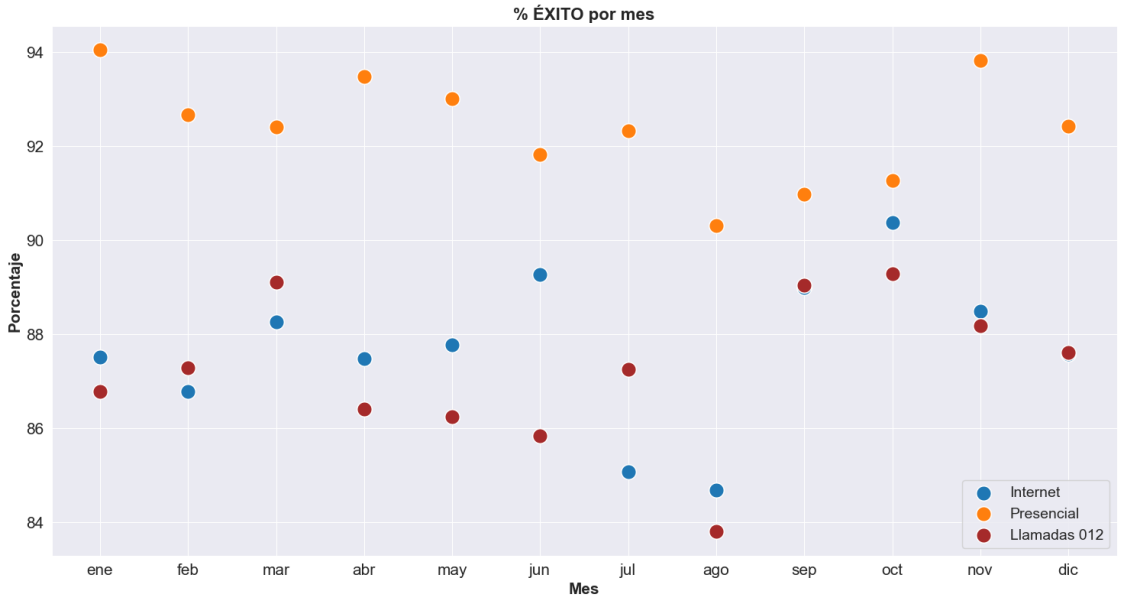
[143]:
```
df_canal_est['terminada'] = df_canal_est.exito + df_canal_est.descarte
df_canal_est['exito_pct'] = 100*df_canal_est.exito / df_canal_est.terminada
df_canal_est['descarte_pct'] = 100*df_canal_est.descarte / df_canal_est.
 ↪terminada
df_canal_est['terminada_pct'] = 100*df_canal_est.terminada / (df_canal_est.
 ↪terminada + df_canal_est.proceso)
df_canal_est['proceso_pct'] = 100*df_canal_est.proceso / (df_canal_est.
 ↪terminada + df_canal_est.proceso)
```

```
df_canal_est['exitoT_pct'] = 100*df_canal_est.exito / (df_canal_est.terminada +␣
 ↪df_canal_est.proceso)
```

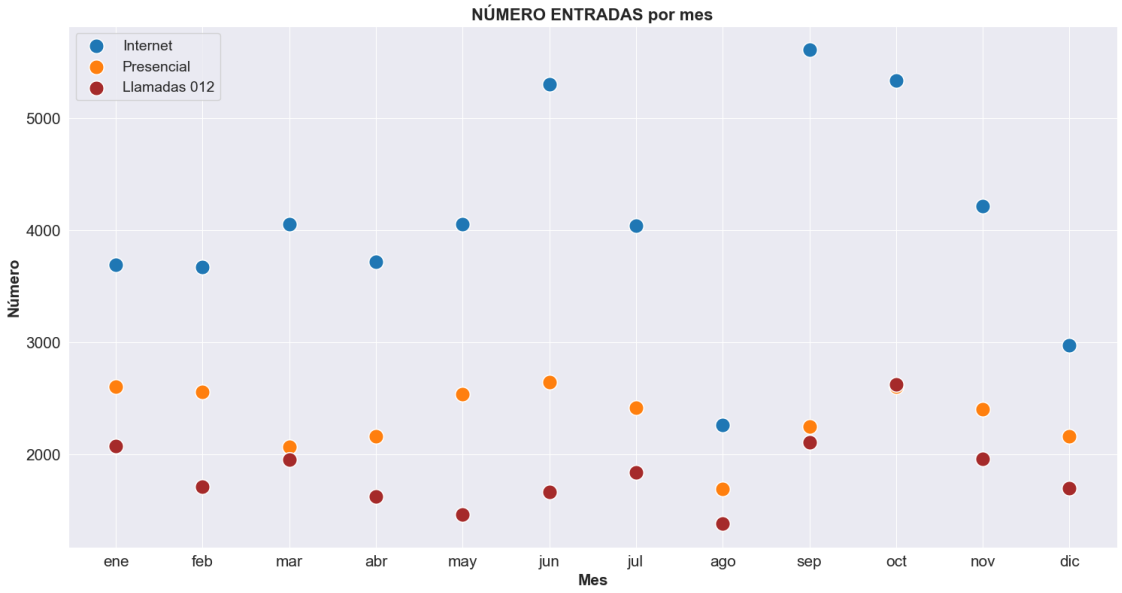
[144]:
```
# Internet
df_int_est = df_canal_est[df_canal_est.canal_entrada == 'INTERNET']

# Presencial
df_pre_est = df_canal_est[df_canal_est.canal_entrada == 'PRESENCIAL']

# Llamadas 012
df_lla_est = df_canal_est[df_canal_est.canal_entrada == 'LLAMADAS 012']
```

GRÁFICOS CONJUNTOS

[145]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_int_est.index, y=df_int_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_pre_est.index, y=df_pre_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_lla_est.index, y=df_lla_est.exito_pct, s=200,␣
 ↪color='brown')

ax.set_title('% ÉXITO por mes',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_int_est.index, labels=['ene', 'feb', 'mar',
                                              'abr', 'may', 'jun',
                                              'jul', 'ago', 'sep',
                                              'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='lower right', labels=['Internet', 'Presencial', 'Llamadas 012'],␣
 ↪fontsize=14);
```



[146]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_int_est.index, y=df_int_est.total, s=200)
sns.scatterplot(ax=ax, x=df_pre_est.index, y=df_pre_est.total, s=200)
sns.scatterplot(ax=ax, x=df_lla_est.index, y=df_lla_est.total, s=200,␣
 ↪color='brown')

ax.set_title('NÚMERO ENTRADAS por mes',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_int_est.index, labels=['ene', 'feb', 'mar',
                                              'abr', 'may', 'jun',
                                              'jul', 'ago', 'sep',
                                              'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper left', labels=['Internet', 'Presencial', 'Llamadas 012'],␣
 ↪fontsize=14);
```



4.3.3 Estacionalidad Día Semana

[147]:
```
df_canal_ds = df_compo.reset_index().groupby(['dia_semana', 'canal_entrada']).
 ↪agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('tipo', 'count'))
```

[148]:
```
df_canal_ds = df_canal_ds.reset_index().set_index('dia_semana')
```

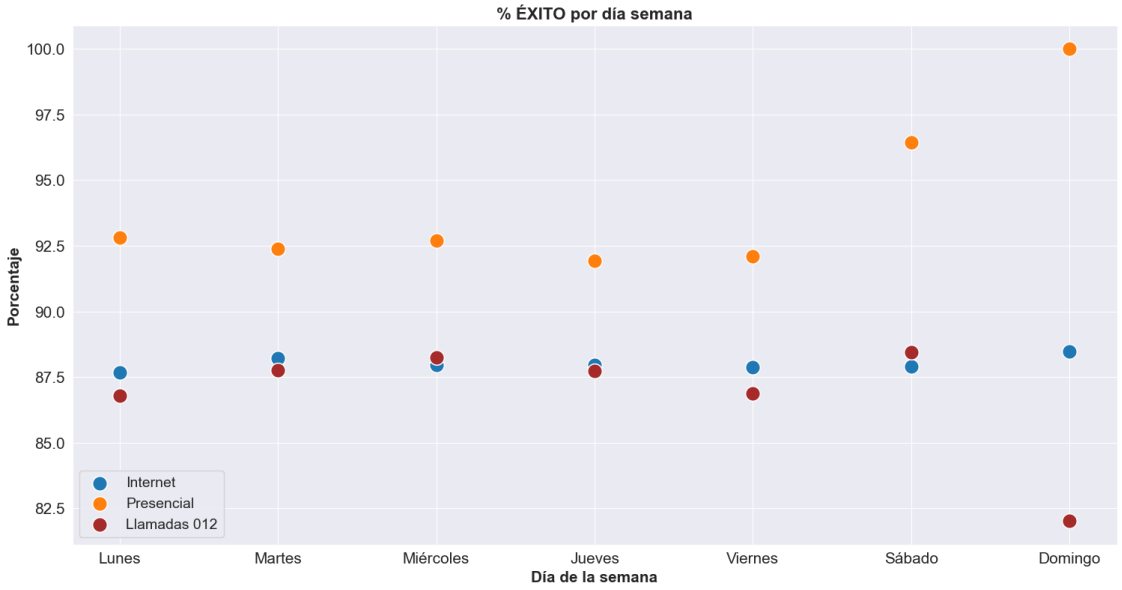
[149]:
```
df_canal_ds['terminada'] = df_canal_ds.exito + df_canal_ds.descarte
df_canal_ds['exito_pct'] = 100*df_canal_ds.exito / df_canal_ds.terminada
df_canal_ds['descarte_pct'] = 100*df_canal_ds.descarte / df_canal_ds.terminada
df_canal_ds['terminada_pct'] = 100*df_canal_ds.terminada / (df_canal_ds.
 ↪terminada + df_canal_ds.proceso)
df_canal_ds['proceso_pct'] = 100*df_canal_ds.proceso / (df_canal_ds.terminada +␣
 ↪df_canal_ds.proceso)
df_canal_ds['exitoT_pct'] = 100*df_canal_ds.exito / (df_canal_ds.terminada +␣
 ↪df_canal_ds.proceso)
```

[150]:
```
# Internet
df_int_ds = df_canal_ds[df_canal_ds.canal_entrada == 'INTERNET']

# Presencial
df_pre_ds = df_canal_ds[df_canal_ds.canal_entrada == 'PRESENCIAL']
```

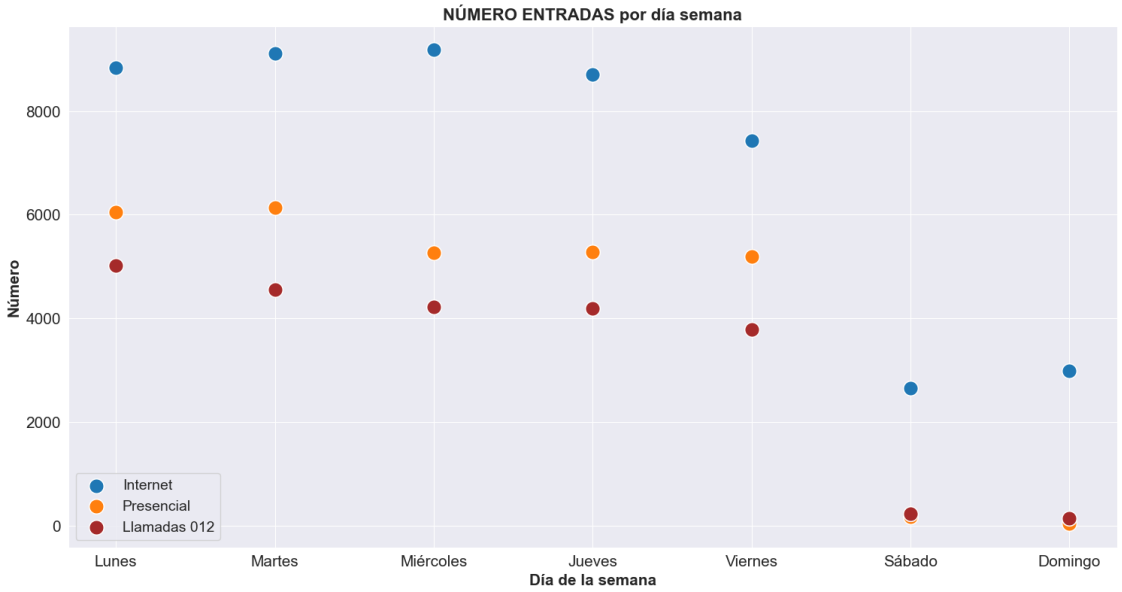
```
# Llamadas 012
df_lla_ds = df_canal_ds[df_canal_ds.canal_entrada == 'LLAMADAS 012']
```

GRÁFICOS CONJUNTOS

[151]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_int_ds.index, y=df_int_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_pre_ds.index, y=df_pre_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_lla_ds.index, y=df_lla_ds.exito_pct, s=200,␣
 ↪color='brown')

ax.set_title('% ÉXITO por día semana', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))

ax.set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_int_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='lower left', labels=['Internet', 'Presencial', 'Llamadas 012'],␣
 ↪fontsize=14);
```



[152]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_int_ds.index, y=df_int_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_pre_ds.index, y=df_pre_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_lla_ds.index, y=df_lla_ds.total, s=200,␣
 ↪color='brown')

ax.set_title('NÚMERO ENTRADAS por día semana', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))

ax.set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_int_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='lower left', labels=['Internet', 'Presencial', 'Llamadas 012'],␣
 ↪fontsize=14);
```



4.4 Consejería

4.4.1 Mensual

Reagrupamiento de Consejerías para crear Consejerías estables a lo largo del tiempo y más homogéneas.

[153]:
```
# Agrupamiento de consejerías
dic_sust = {'Medioambiental': ['Consejería de Medio Ambiente, Vivienda y␣
↪Agricultura',
                               'Consejería de Medio Ambiente, Ordenación del␣
↪Territorio y Sostenibilidad'],
            'Transporte': ['Consorcio Regional de Transportes',
                           'Consejería de Transportes, Vivienda e␣
↪Infraestructuras',
                           'Consejería de Transportes e Infraestructuras'],
            'Educacion': ['Consejería de Educación, Juventud y Deporte',
                          'Consejería de Educación, Universidades, Ciencia y␣
↪Portavocía',
                          'Consejería de Educación e Investigación'],
            'Cultura': ['Consejería de Cultura, Turismo y Deporte',
                        'Oficina de Cultura y Turismo'],
            'Subvenciones': ['Consejería de Familia, Juventud y Política␣
↪Social',
                             'Consejería de Políticas Sociales, Familias,␣
↪Igualdad y Natalidad',
                             'Consejería de Sanidad'],
            'Comunicacion': ['Consejería de Presidencia, Justicia e Interior',
                             'Consejería de Presidencia, Justicia y Portavocía␣
↪del Gobierno',
                             'Vicepresidencia, Consejería de Deportes,␣
↪Transparencia y Portavocía de Gobierno'],
            'Economia': ['Consejería de Economía, Hacienda y Empleo'],
            'Digitalizacion': ['Agencia para la Administración Digital de la␣
↪Comunidad de Madrid']}
```

[154]:
```
def sustituir(dfs, col, valores, nombre):
    """
    Devuelve un numpy.ndarray con los valores indicados sustituidos.

    dfs: dataframe donde esta la columna a sustituir.
    col: nombre de la columna con los valores a sustituir.
    valores: conjunto de valores que serán sustituidos.
    nombre: cadena que sustituirá a los valores.
    """

    return np.where(dfs[col].isin(valores), nombre, dfs[col])
```

[155]:
```
# Copia de columna
df_compo['nueva_conse'] = df_compo['consejeria']
```

[156]:
```
# Sustitución en bucle
for k, v in dic_sust.items():
    df_compo['nueva_conse'] = sustituir(df_compo, "nueva_conse", v, k)
```

[157]:
```
df_compo[df_compo.nueva_conse.isna()].tema.value_counts()
```

[157]:
```
TRANSPORTES           2997
POLÍTICAS SOCIALES     942
EDUCACIÓN              638
CULTURA                388
Name: tema, dtype: int64
```

[158]:
```
# Sustitución de consejeria=NA por el valor en tema
df_compo['nueva_conse'] = np.where(df_compo.nueva_conse.isna(),
                                   np.where(df_compo.tema == 'EDUCACIÓN',␣
 ↪'Educacion',
                                            df_compo.nueva_conse),
                                   df_compo.nueva_conse)
df_compo['nueva_conse'] = np.where(df_compo.nueva_conse.isna(),
                                   np.where(df_compo.tema == 'TRANSPORTES',␣
 ↪'Transporte',
                                            df_compo.nueva_conse),
                                   df_compo.nueva_conse)
df_compo['nueva_conse'] = np.where(df_compo.nueva_conse.isna(),
                                   np.where(df_compo.tema == 'POLÍTICAS␣
 ↪SOCIALES', 'Subvenciones',
                                            df_compo.nueva_conse),
                                   df_compo.nueva_conse)
df_compo['nueva_conse'] = np.where(df_compo.nueva_conse.isna(),
                                   np.where(df_compo.tema == 'CULTURA',␣
 ↪'Cultura',
                                            df_compo.nueva_conse),
                                   df_compo.nueva_conse)
```

[159]:
```
df_compo[df_compo.nueva_conse.isna()].tema.value_counts()
```

[159]:
```
Series([], Name: tema, dtype: int64)
```

[160]:
```
df_conse = df_compo.reset_index().groupby(['fecha_entrada', 'nueva_conse']).agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('tipo', 'count'))
```

[161]:
```
df_conse = df_conse.reset_index().set_index('fecha_entrada').
 ↪groupby('nueva_conse').resample('M').agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
```

```
    total=('total', 'sum'))
```

[162]:
```
df_conse = df_conse.reset_index().set_index('fecha_entrada')
```

[163]:
```
df_conse['terminada'] = df_conse.exito + df_conse.descarte
df_conse['exito_pct'] = 100*df_conse.exito / df_conse.terminada
df_conse['descarte_pct'] = 100*df_conse.descarte / df_conse.terminada
df_conse['terminada_pct'] = 100*df_conse.terminada / (df_conse.terminada +␣
 ↪df_conse.proceso)
df_conse['proceso_pct'] = 100*df_conse.proceso / (df_conse.terminada + df_conse.
 ↪proceso)
df_conse['exitoT_pct'] = 100*df_conse.exito / (df_conse.terminada + df_conse.
 ↪proceso)
```

[164]:
```
df_conse
```

[164]:
```
                nueva_conse  descarte  exito  proceso  terminada  total  \
fecha_entrada
2013-02-28     Comunicacion         2      2        0          4      4
2013-03-31     Comunicacion         2      0        0          2      2
2013-04-30     Comunicacion         0      0        0          0      0
…                       …       …     …       …          …    …
2022-10-31       Transporte         8    127        9        135    144
2022-11-30       Transporte         5     98        1        103    104
2022-12-31       Transporte         4    126        6        130    136

                exito_pct  descarte_pct  terminada_pct  proceso_pct  exitoT_pct
fecha_entrada
2013-02-28      50.000000     50.000000     100.000000     0.000000   50.000000
2013-03-31       0.000000    100.000000     100.000000     0.000000    0.000000
2013-04-30            NaN           NaN            NaN          NaN         NaN
…                       …             …              …            …           …
2022-10-31      94.074074      5.925926      93.750000     6.250000   88.194444
2022-11-30      95.145631      4.854369      99.038462     0.961538   94.230769
2022-12-31      96.923077      3.076923      95.588235     4.411765   92.647059

[952 rows x 11 columns]
```

[165]:
```
# Transporte
df_trans = df_conse[df_conse.nueva_conse == 'Transporte']

# Cultura
df_cul = df_conse[df_conse.nueva_conse == 'Cultura']

# Digitalizacion
df_digi = df_conse[df_conse.nueva_conse == 'Digitalizacion']
```

```
# Educacion
df_educ = df_conse[df_conse.nueva_conse == 'Educacion']

# Subvenciones
df_subv = df_conse[df_conse.nueva_conse == 'Subvenciones']

# Comunicacion
df_comu = df_conse[df_conse.nueva_conse == 'Comunicacion']

# Economia
df_econ = df_conse[df_conse.nueva_conse == 'Economia']

# Medioambiental
df_medio = df_conse[df_conse.nueva_conse == 'Medioambiental']
```

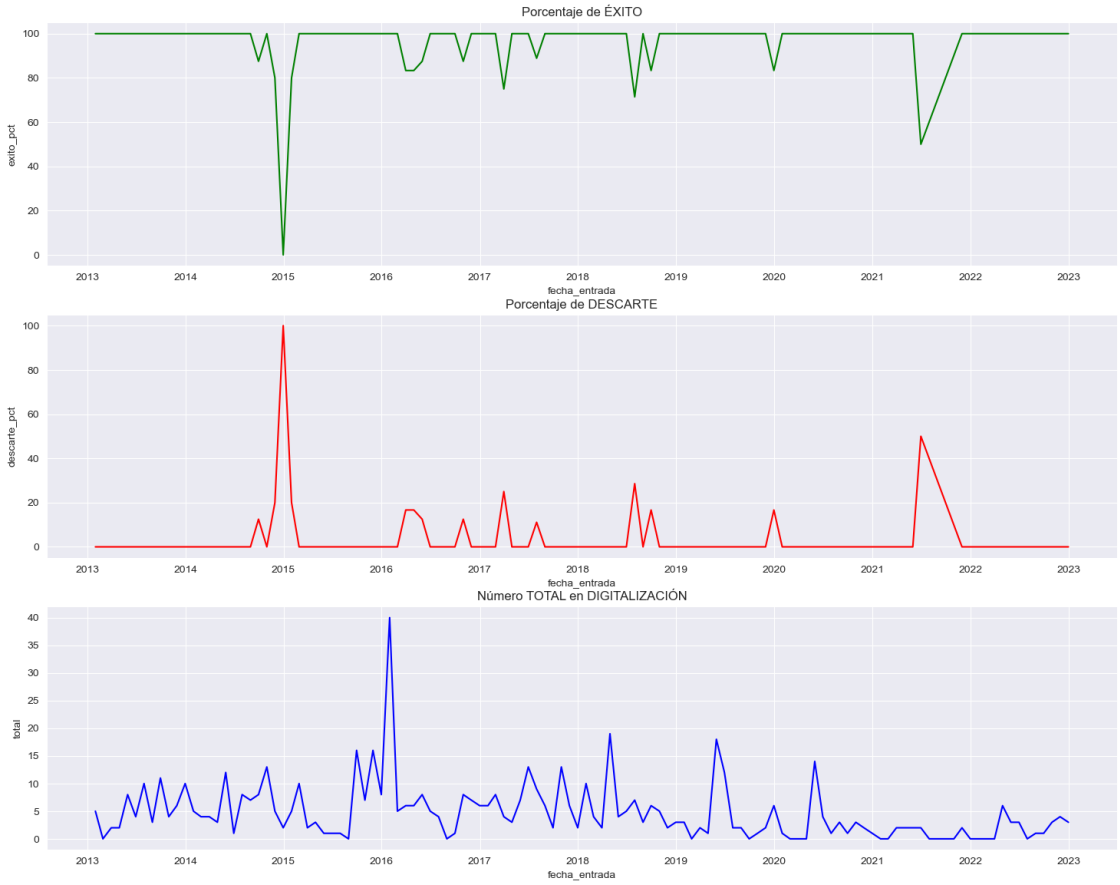
[166]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_trans, x=df_trans.index, y=df_trans.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_trans, x=df_trans.index, y=df_trans.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_trans, x=df_trans.index, y=df_trans.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en TRANSPORTE',␣
 ↪loc='center');
```

- **Transporte**. Mejoran las tasas y disminuye el número de entradas.

[167]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_cul, x=df_cul.index, y=df_cul.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_cul, x=df_cul.index, y=df_cul.descarte_pct,␣
 ↪color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_cul, x=df_cul.index, y=df_cul.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en CULTURA', loc='center');
```

- **Cultura**. Tasas estables. El número de entradas es estable o en ligero descenso a partir de 2020.

[168]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_digi, x=df_digi.index, y=df_digi.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_digi, x=df_digi.index, y=df_digi.descarte_pct,␣
 ↪color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_digi, x=df_digi.index, y=df_digi.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en DIGITALIZACIÓN',␣
 ↪loc='center');
```



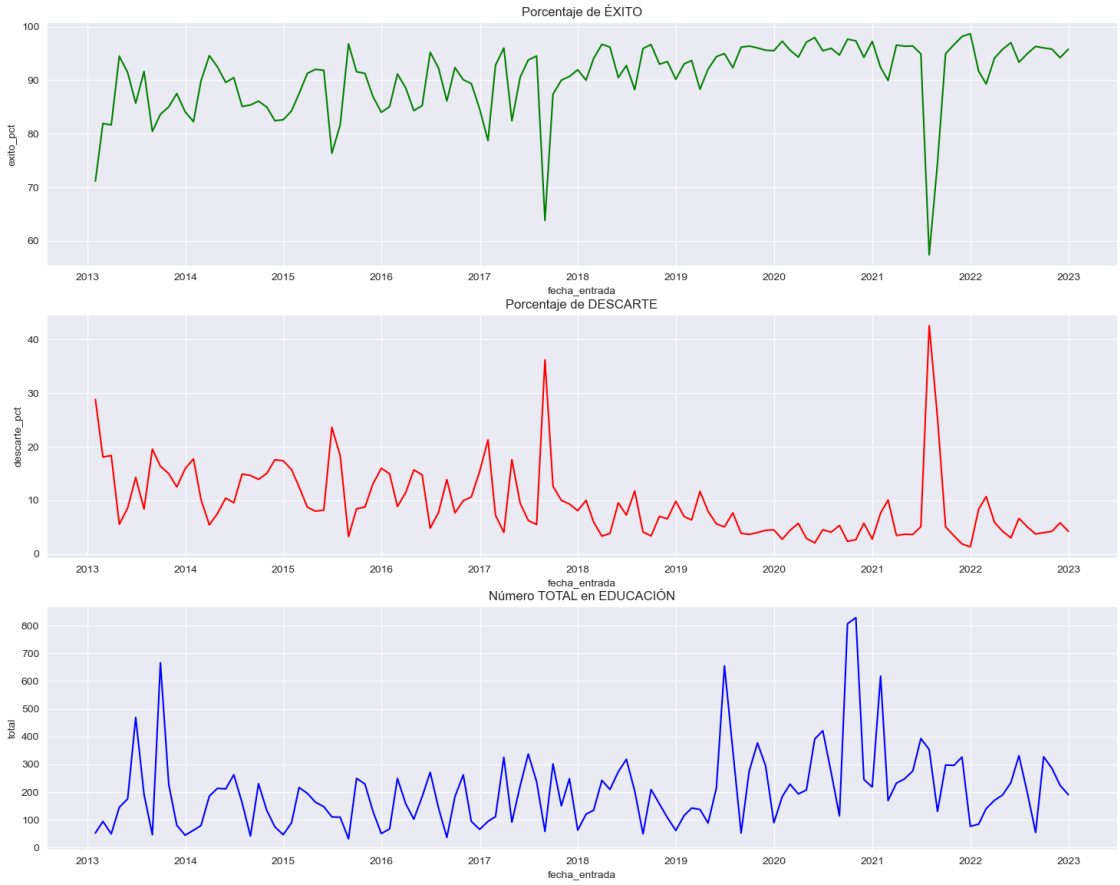
- **Digitalización**. Las tasas son estables y pobres. Las entradas han disminuido ligeramente a lo largo del tiempo.
- Hay un **pico en las tasas en 2015** (el dataset empieza en 2013). Tal vez tiene que ver con https://www.comunidad.madrid/servicios/sede-electronica/madrid-digital:

Creación

Ley de creación, Ley 7/2005, de 23 de diciembre, de Medidas Fiscales y Administrativas. Se crea la Agencia de Informática y Comunicaciones de la Comunidad de Madrid y establecen sus funciones, Ley 9/2015, de 28 de diciembre, de Medidas Fiscales y Administrativas, que en su artículo 4 modifica parcialmente el artículo 10 de la Ley 7/2005, pasando la Agencia a denominarse "Agencia para la Administración Digital de la Comunidad de Madrid".

[169]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_educ, x=df_educ.index, y=df_educ.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_educ, x=df_educ.index, y=df_educ.descarte_pct,␣
 ↪color='red') \
```

```
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_educ, x=df_educ.index, y=df_educ.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en EDUCACIÓN', loc='center');
```



- **Educación**. Mejora en las tasas. El número de entradas parece aumentar a partir de 2019.
- Hay un dato extrañamente bajo en la segunda mitad de 2021. No sé a qué es debido.

[170]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_subv, x=df_subv.index, y=df_subv.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_subv, x=df_subv.index, y=df_subv.descarte_pct,␣
 ↪color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')
```

```
sns.lineplot(ax=ax[2], data=df_subv, x=df_subv.index, y=df_subv.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en SUBVENCIONES (POLÍTICAS␣
 ↪SOCIALES)', loc='center');
```

- **Subvenciones**. Las tasas han mejorado, pero con grandes altibajos. Fuerte empeoramiento (y recuperación) en el último trimestre de 2020. No sé a qué es debido.
- El número de entradas es creciente, pero también tiene fuertes oscilaciones.

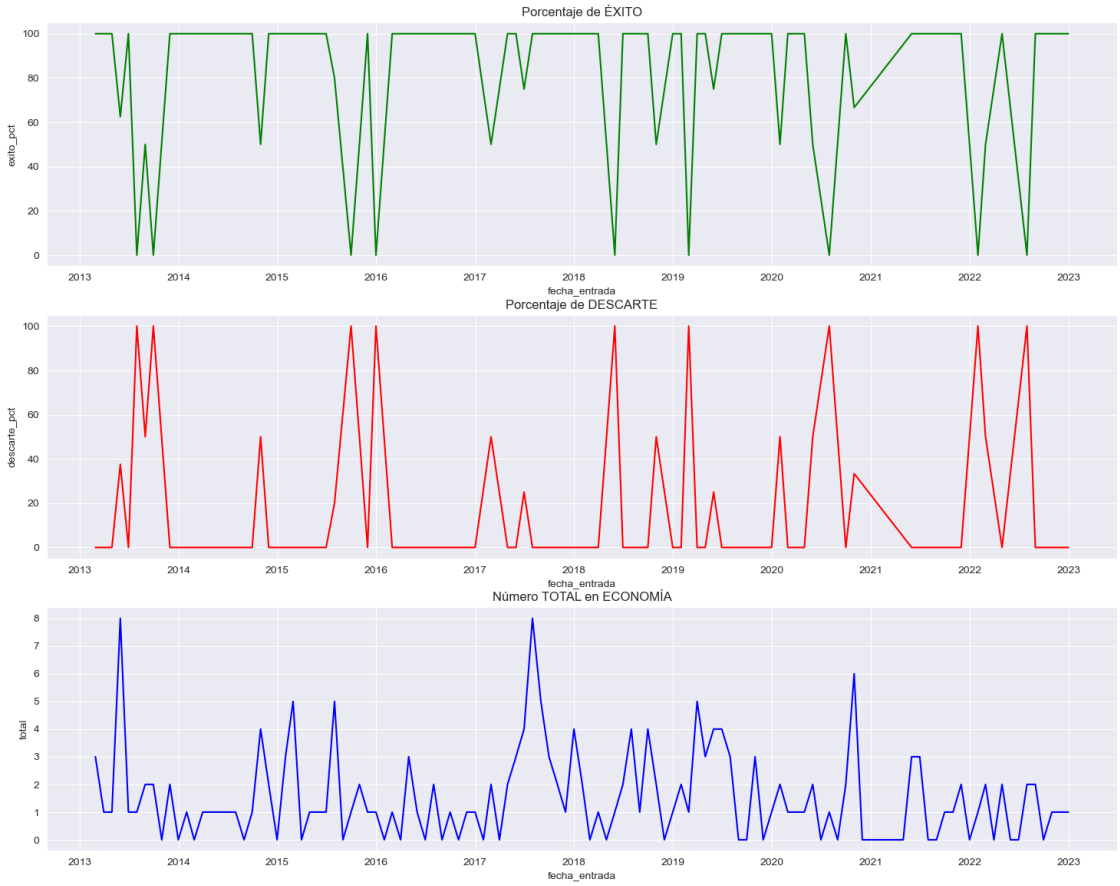
[171]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_comu, x=df_comu.index, y=df_comu.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_comu, x=df_comu.index, y=df_comu.descarte_pct,␣
 ↪color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')
```

```
sns.lineplot(ax=ax[2], data=df_comu, x=df_comu.index, y=df_comu.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en COMUNICACIÓN',␣
 ↪loc='center');
```

- **Comunicación**. Hay escasez de datos. Ha habido un empeoramiento radical en las tasas desde el último cuatrimestre de 2019.
- El número de entradas oscila en torno a 3.

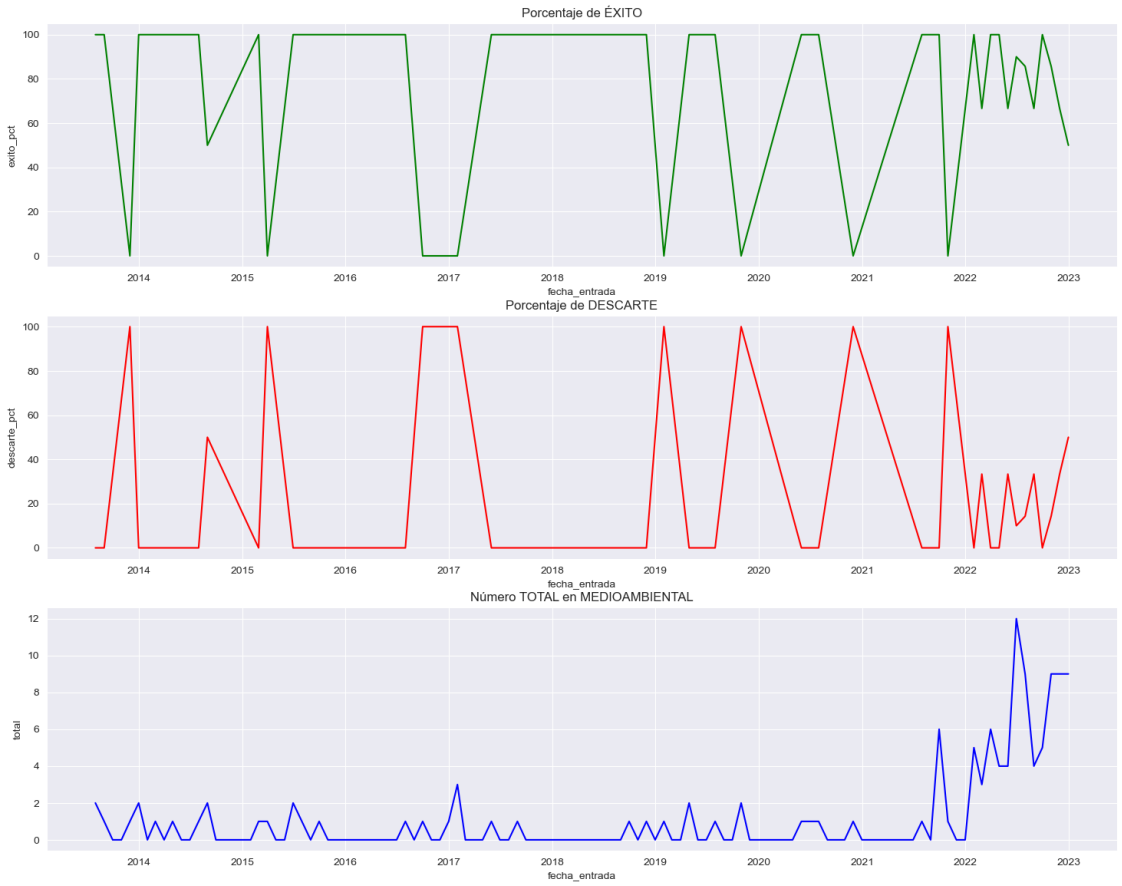
[172]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_econ, x=df_econ.index, y=df_econ.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_econ, x=df_econ.index, y=df_econ.descarte_pct,␣
 ↪color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')
```

```
sns.lineplot(ax=ax[2], data=df_econ, x=df_econ.index, y=df_econ.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en ECONOMÍA', loc='center');
```



- **Economía**. Pocos datos. Todo parece estable.

[173]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_medio, x=df_medio.index, y=df_medio.exito_pct,␣
 ↪color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_medio, x=df_medio.index, y=df_medio.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_medio, x=df_medio.index, y=df_medio.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en MEDIOAMBIENTAL',␣
 ↪loc='center');
```



- **Medioambiental**. Muy pocas entradas hasta finales de 2021 y principios de 2022: crecimiento de 2 a 9 mensuales, marcando **récord histórico**.

Caso especial: tasa de reenvío REENVÍO como CONSEJERÍA

Se ha estudiado el reenvío a través del "tema". Ahora se verá a través de la Consejería, que parece más informativo puesto que apunta al directamente al agente que gestiona la respuesta a la entrada.

- **Transportes (media = 10,4%)**. Tasa de reenvío desciende. Es la tasa más alta junto con Cultura. Transporte empieza siendo mayor, pero a partir de 2021 Cultura supera a Transportes.
- **Cultura (media = 4,7%)**. Sube la tasa ligeramente.
- Digital. Muy pocos datos.
- **Educación (media = 2,6%)**. Tasa estable. Pico puntual destacado en la segunda mitad de 2021.
- **Subvenciones-Política Social (media = 2,1%)**. Ligero descenso pero se mantiene en porcentajes muy bajos.
- Comunicación. Pocos datos y solo a partir de 2020.
- Economía. Pocos datos.
- Medioambiental. Pocos datos.

- **Subvenciones-Política Social (media = 342)**. Tiene el mayor número de reenviadas. En 2021-2022 parece tener tendencia creciente. Dado que su tasa de reenvío media es baja y no oscila mucho, parece que **el número de reenviadas crece al mismo ritmo que "terminada"**.
- **Transportes (media = 206)**. El número de reenviadas es decreciente, al igual que la tasa de reenvío.
- **Educación (media = 204)**. Tiene grandes oscilaciones. No parecen crecer a lo largo del tiempo.
- **Cultura (media = 69)**. El número de reenviadas disminuye en escalón en 2020 y se estabiliza en un nivel algo inferior al anterior, pero la tasa de reenvío parece aumentar a lo largo del tiempo, especialmente en 2016 y 2020. Esto significa que el número de reenvíos ha descendido menos que el número de "entradas terminadas" por lo que la tasa asciende. **La finalización con reenvío en Cultura es cada vez más importante respecto al total de "terminada"**.

[174]:
```
df_contte = df_compo.reset_index().groupby(['fecha_entrada', 'nueva_conse']).
 ↪agg(
    reenviada=('reenviada', 'sum'),
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('tipo', 'count'))
```

[175]:
```
df_contte = df_contte.reset_index().set_index('fecha_entrada').
 ↪groupby('nueva_conse').resample('M').agg(
    reenviada=('reenviada', 'sum'),
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('total', 'sum'))
```

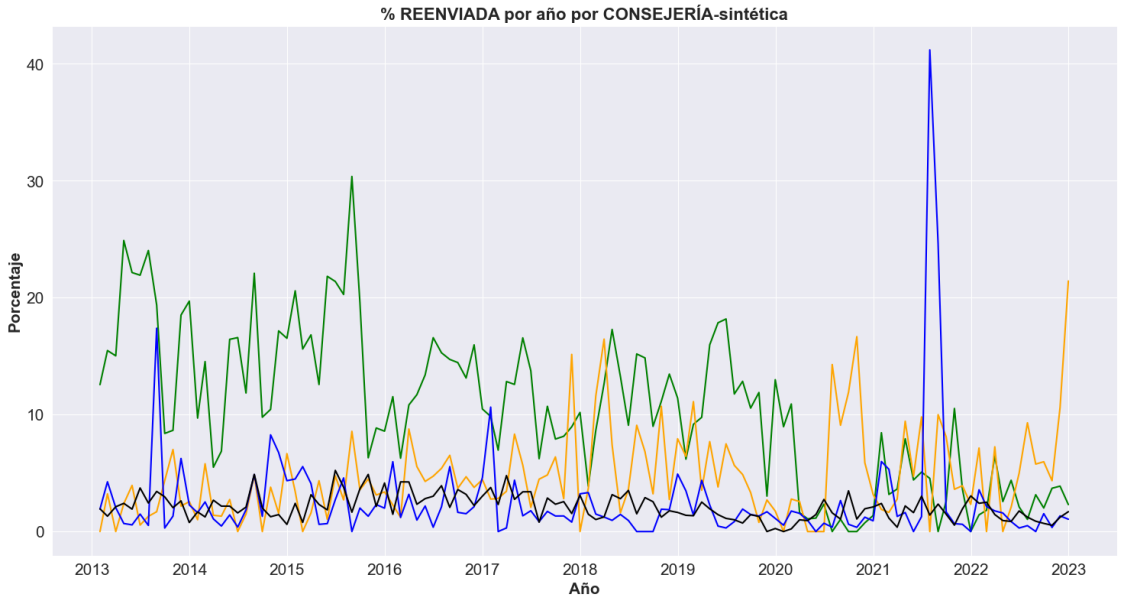
[176]:
```
df_contte = df_contte.reset_index().set_index('fecha_entrada')
```

[177]:
```
df_contte['terminada'] = df_contte.exito + df_contte.descarte
df_contte['exito_pct'] = 100*df_contte.exito / df_contte.terminada
df_contte['descarte_pct'] = 100*df_contte.descarte / df_contte.terminada
df_contte['terminada_pct'] = 100*df_contte.terminada / (df_contte.terminada +␣
 ↪df_contte.proceso)
df_contte['proceso_pct'] = 100*df_contte.proceso / (df_contte.terminada +␣
 ↪df_contte.proceso)
df_contte['exitoT_pct'] = 100*df_contte.exito / (df_contte.terminada +␣
 ↪df_contte.proceso)
df_contte['reenviada_pct'] = 100*df_contte.reenviada / df_contte.terminada
df_contte['reenviadaT_pct'] = 100*df_contte.reenviada / (df_contte.terminada +␣
 ↪df_contte.proceso)
```

[178]:
```
# Transporte
df_trans_ct = df_contte[df_contte.nueva_conse == 'Transporte']

# Cultura
df_cul_ct = df_contte[df_contte.nueva_conse == 'Cultura']

# Digitalizacion
df_digi_ct = df_contte[df_contte.nueva_conse == 'Digitalizacion']

# Educacion
df_educ_ct = df_contte[df_contte.nueva_conse == 'Educacion']

# Subvenciones
df_subv_ct = df_contte[df_contte.nueva_conse == 'Subvenciones']

# Comunicacion
df_comu_ct = df_contte[df_contte.nueva_conse == 'Comunicacion']

# Economia
df_econ_ct = df_contte[df_contte.nueva_conse == 'Economia']

# Medioambiental
df_medio_ct = df_contte[df_contte.nueva_conse == 'Medioambiental']
```

[179]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=df_trans_ct.index, y=df_trans_ct.reenviada_pct,␣
 ↪color='green')
sns.lineplot(ax=ax, x=df_cul_ct.index, y=df_cul_ct.reenviada_pct,␣
 ↪color='orange')
# sns.lineplot(ax=ax, x=df_digi_ct.index, y=df_digi_ct.reenviada_pct,␣
 ↪color='red')
sns.lineplot(ax=ax, x=df_educ_ct.index, y=df_educ_ct.reenviada_pct,␣
 ↪color='blue')
sns.lineplot(ax=ax, x=df_subv_ct.index, y=df_subv_ct.reenviada_pct,␣
 ↪color='black')
# sns.lineplot(ax=ax, x=df_comu_ct.index, y=df_comu_ct.reenviada_pct,␣
 ↪color='grey')
# sns.lineplot(ax=ax, x=df_econ_ct.index, y=df_econ_ct.reenviada_pct,␣
 ↪color='cyan')
# sns.lineplot(ax=ax, x=df_medio_ct.index, y=df_medio_ct.reenviada_pct,␣
 ↪color='brown')


ax.set_title('% REENVIADA por año por CONSEJERÍA-sintética',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))
```

```
ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15);
```

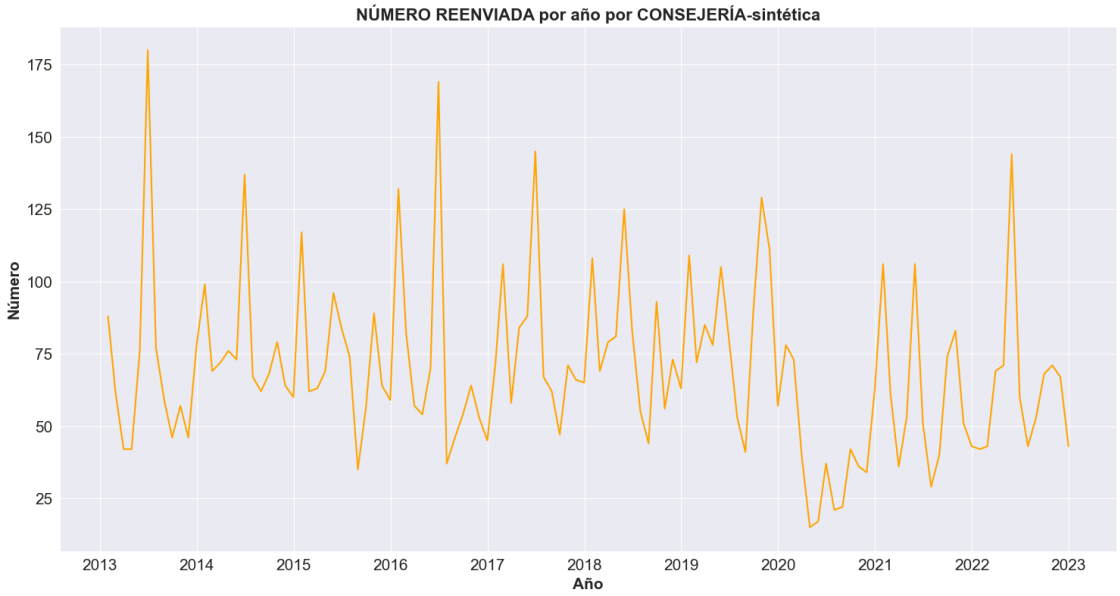


[180]:
```
print("Transportes: ", df_trans_ct.reenviada_pct.mean())
print("Cultura: ", df_cul_ct.reenviada_pct.mean())
print("Educación: ", df_educ_ct.reenviada_pct.mean())
print("Subvenciones: ", df_subv_ct.reenviada_pct.mean())
```

```
Transportes:  10.402548342809473
Cultura:  4.6879961687527665
Educación:  2.61114367967364
Subvenciones:  2.1008320628725135
```

[181]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

# sns.lineplot(ax=ax, x=df_trans_ct.index, y=df_trans_ct.total, color='green')
sns.lineplot(ax=ax, x=df_cul_ct.index, y=df_cul_ct.total, color='orange')
# sns.lineplot(ax=ax, x=df_digi_ct.index, y=df_digi_ct.total, color='red')
# sns.lineplot(ax=ax, x=df_educ_ct.index, y=df_educ_ct.total, color='blue')
# sns.lineplot(ax=ax, x=df_subv_ct.index, y=df_subv_ct.total, color='black')
# sns.lineplot(ax=ax, x=df_comu_ct.index, y=df_comu_ct.total, color='grey')
# sns.lineplot(ax=ax, x=df_econ_ct.index, y=df_econ_ct.total, color='cyan')
# sns.lineplot(ax=ax, x=df_medio_ct.index, y=df_medio_ct.total, color='brown')
```

```
ax.set_title('NÚMERO REENVIADA por año por CONSEJERÍA-sintética',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15);
```



[182]:
```
print("Transportes: ", df_trans_ct.total.mean())
print("Cultura: ", df_cul_ct.total.mean())
print("Educación: ", df_educ_ct.total.mean())
print("Subvenciones: ", df_subv_ct.total.mean())
```

```
Transportes:  206.61666666666667
Cultura:  69.7
Educación:  204.34166666666667
Subvenciones:  342.85833333333335
```

4.4.2 Estacionalidad Año

- **Transporte**. Máximo número de entradas en octubre, tras el verano. Número de entradas más bajo en mayo y junio.
- En octubre también se produce la tasa de éxito más alta.
- **Cultura**. Cultura incluye Turismo. Pico de entradas en junio.

- Mayor tasa de éxito en enero y, después, en mayo.
- **Digitalización**. Pico de entradas en mayo. Valle en marzo y agosto.
- **Educación**. Entradas máximas en junio y septiembre (final y principio del curso escolar). La tasa de éxito cae en julio y agosto.
- **Subvenciones**. Incluye Familina y Natalidad. Menor número de entradas en agosto y máxima en noviembre. Tasa de éxito más baja en agosto y septiembre.
- **Comunicación**. Incluye Justicia. Máximo en entradas en mayo y noviembre. Mayor tasa de éxito en julio y menor en diciembre.
- Tiene una gran oscilación en la tasa de éxito.
- **Economía**. Incluye Hacienda y Empleo. Mayo y julio mayores entradas (declaración de la renta).
- Tiene gran oscilación en la tasa de éxito.
- **Medioambiental**. Incluye Agricultura y Vivienda. Máximo número de entradas en junio y julio. Segunda parte del año tiene valores más altos.
- Gran oscilación en la tasa de éxito.

[183]:
```
df_conse_est = df_compo.reset_index().groupby(['mes', 'nueva_conse']).agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('tipo', 'count'))
```

[184]:
```
df_conse_est = df_conse_est.reset_index().set_index('mes')
```

[185]:
```
df_conse_est['terminada'] = df_conse_est.exito + df_conse_est.descarte
df_conse_est['exito_pct'] = 100*df_conse_est.exito / df_conse_est.terminada
df_conse_est['descarte_pct'] = 100*df_conse_est.descarte / df_conse_est.
 ↪terminada
df_conse_est['terminada_pct'] = 100*df_conse_est.terminada / (df_conse_est.
 ↪terminada + df_conse_est.proceso)
df_conse_est['proceso_pct'] = 100*df_conse_est.proceso / (df_conse_est.
 ↪terminada + df_conse_est.proceso)
df_conse_est['exitoT_pct'] = 100*df_conse_est.exito / (df_conse_est.terminada +␣
 ↪df_conse_est.proceso)
```

[186]:
```
df_conse_est
```

[186]:

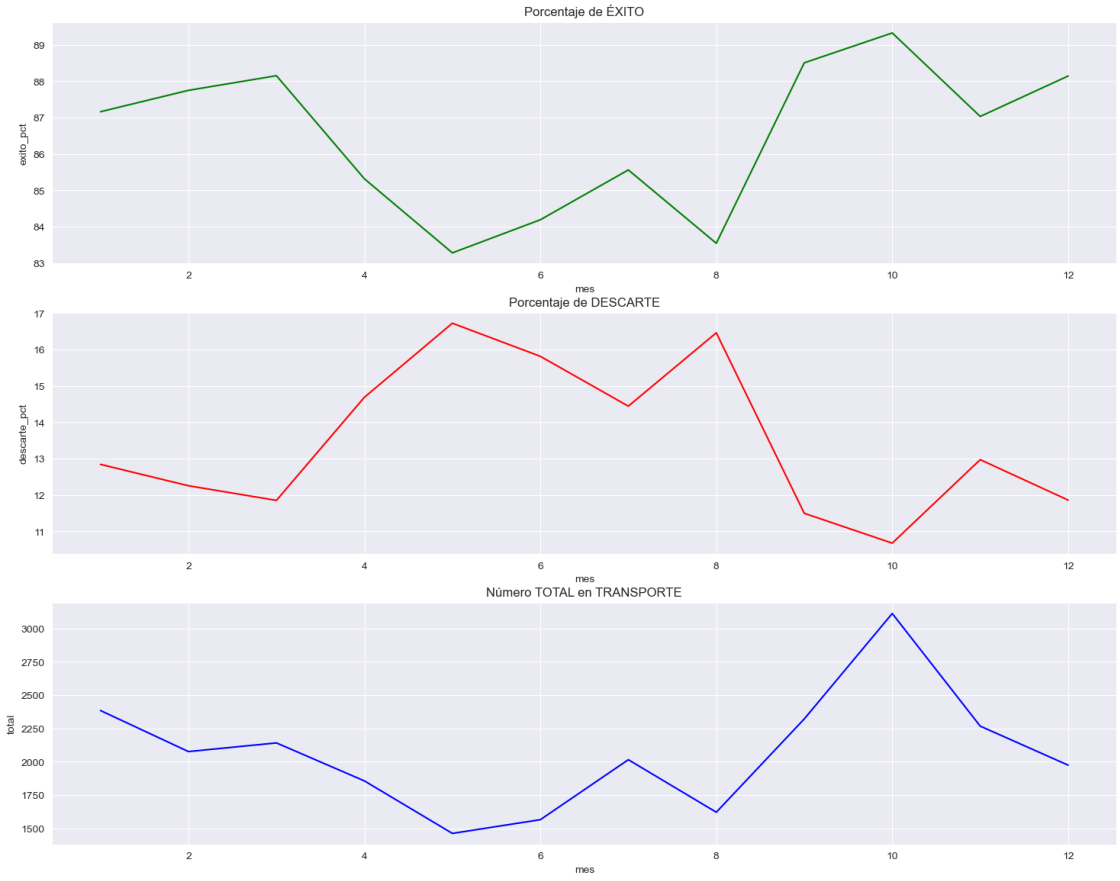
mes	nueva_conse	descarte	exito	proceso	terminada	total	exito_pct
1	Comunicacion	15	13	0	28	28	46.428571
1	Cultura	55	896	0	951	951	94.216614
1	Digitalizacion	1	74	0	75	75	98.666667

```
..              …       …    …       …         …    …         …
12   Medioambiental         3      4         5          7     12  57.142857
12     Subvenciones       346   2897       100       3243   3343  89.330866
12        Transporte       233   1732         9       1965   1974  88.142494

     descarte_pct  terminada_pct  proceso_pct  exitoT_pct
mes
1       53.571429     100.000000     0.000000   46.428571
1        5.783386     100.000000     0.000000   94.216614
1        1.333333     100.000000     0.000000   98.666667
..            …             …            …          …
12      42.857143      58.333333    41.666667   33.333333
12      10.669134      97.008675     2.991325   86.658690
12      11.857506      99.544073     0.455927   87.740628

[96 rows x 11 columns]
```

[187]:
```
# Transporte
df_trans_est = df_conse_est[df_conse_est.nueva_conse == 'Transporte']

# Cultura
df_cul_est = df_conse_est[df_conse_est.nueva_conse == 'Cultura']

# Digitalizacion
df_digi_est = df_conse_est[df_conse_est.nueva_conse == 'Digitalizacion']

# Educacion
df_educ_est = df_conse_est[df_conse_est.nueva_conse == 'Educacion']

# Subvenciones
df_subv_est = df_conse_est[df_conse_est.nueva_conse == 'Subvenciones']

# Comunicacion
df_comu_est = df_conse_est[df_conse_est.nueva_conse == 'Comunicacion']

# Economia
df_econ_est = df_conse_est[df_conse_est.nueva_conse == 'Economia']

# Medioambiental
df_medio_est = df_conse_est[df_conse_est.nueva_conse == 'Medioambiental']
```

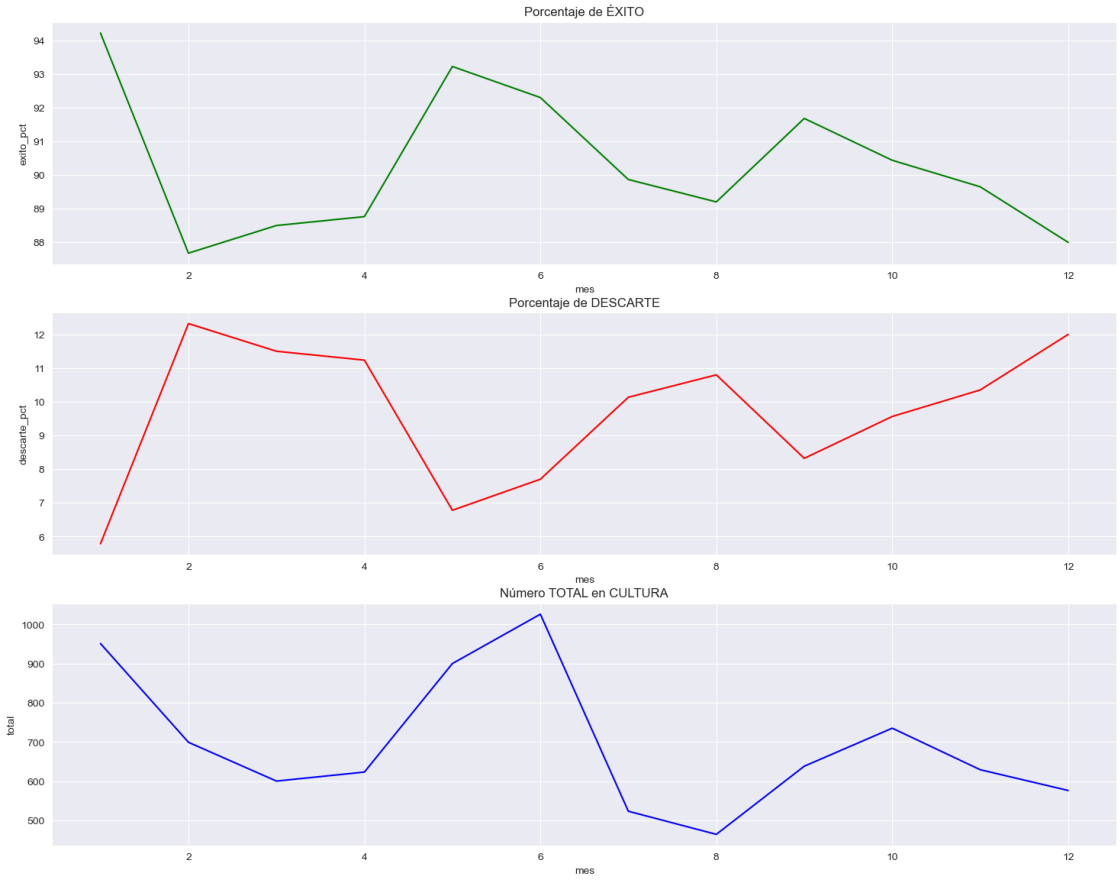
[188]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_trans_est, x=df_trans_est.index, y=df_trans_est.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')
```

```
sns.lineplot(ax=ax[1], data=df_trans_est, x=df_trans_est.index, y=df_trans_est.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_trans_est, x=df_trans_est.index, y=df_trans_est.
 ↪total, color='blue') \
                    .set_title(label='Número TOTAL en TRANSPORTE',␣
 ↪loc='center');
```

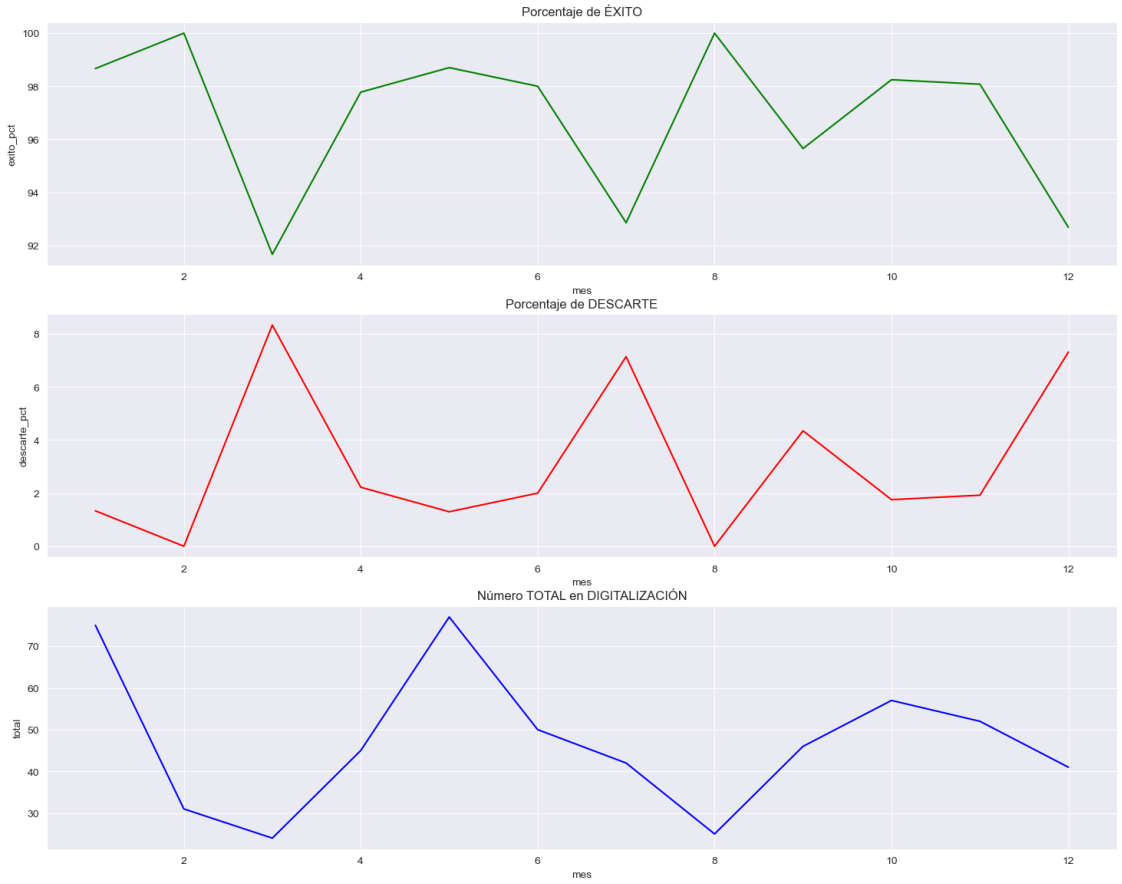


[189]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_cul_est, x=df_cul_est.index, y=df_cul_est.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_cul_est, x=df_cul_est.index, y=df_cul_est.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')
```

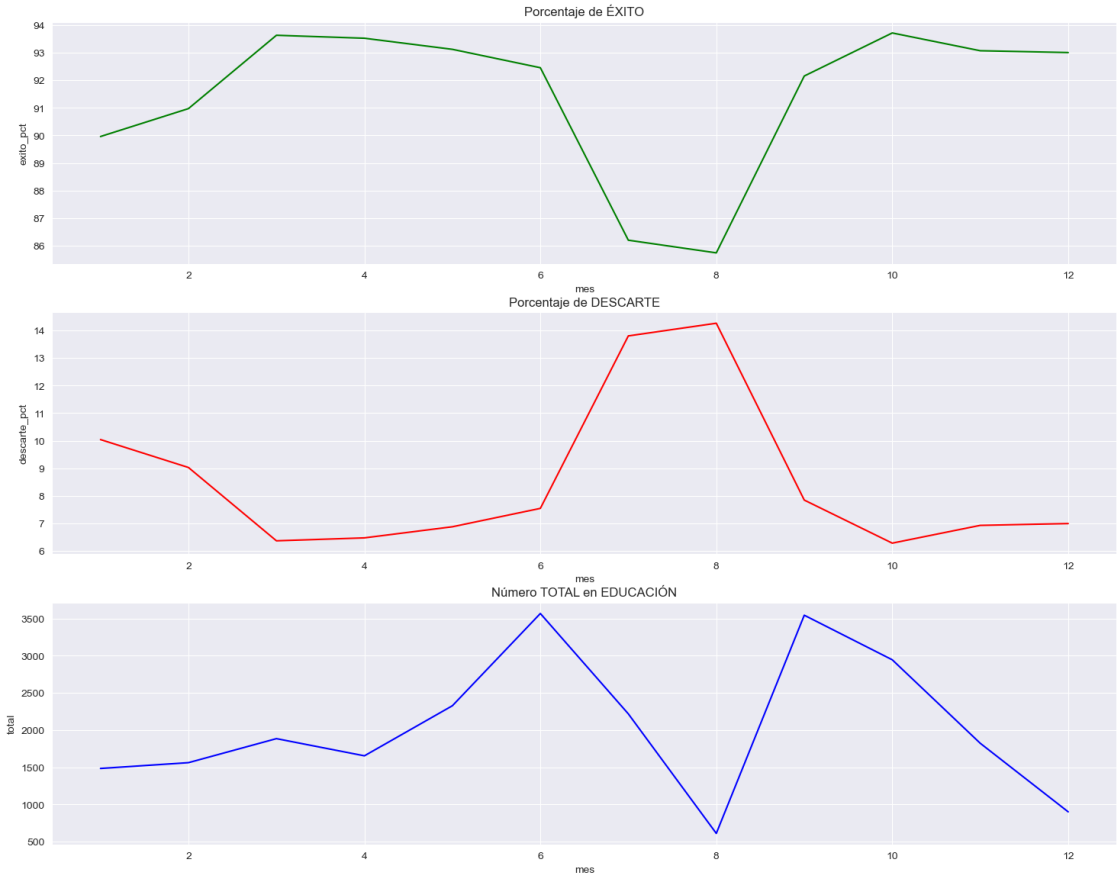
```
sns.lineplot(ax=ax[2], data=df_cul_est, x=df_cul_est.index, y=df_cul_est.total,␣
 ↪color='blue') \
                    .set_title(label='Número TOTAL en CULTURA', loc='center');
```



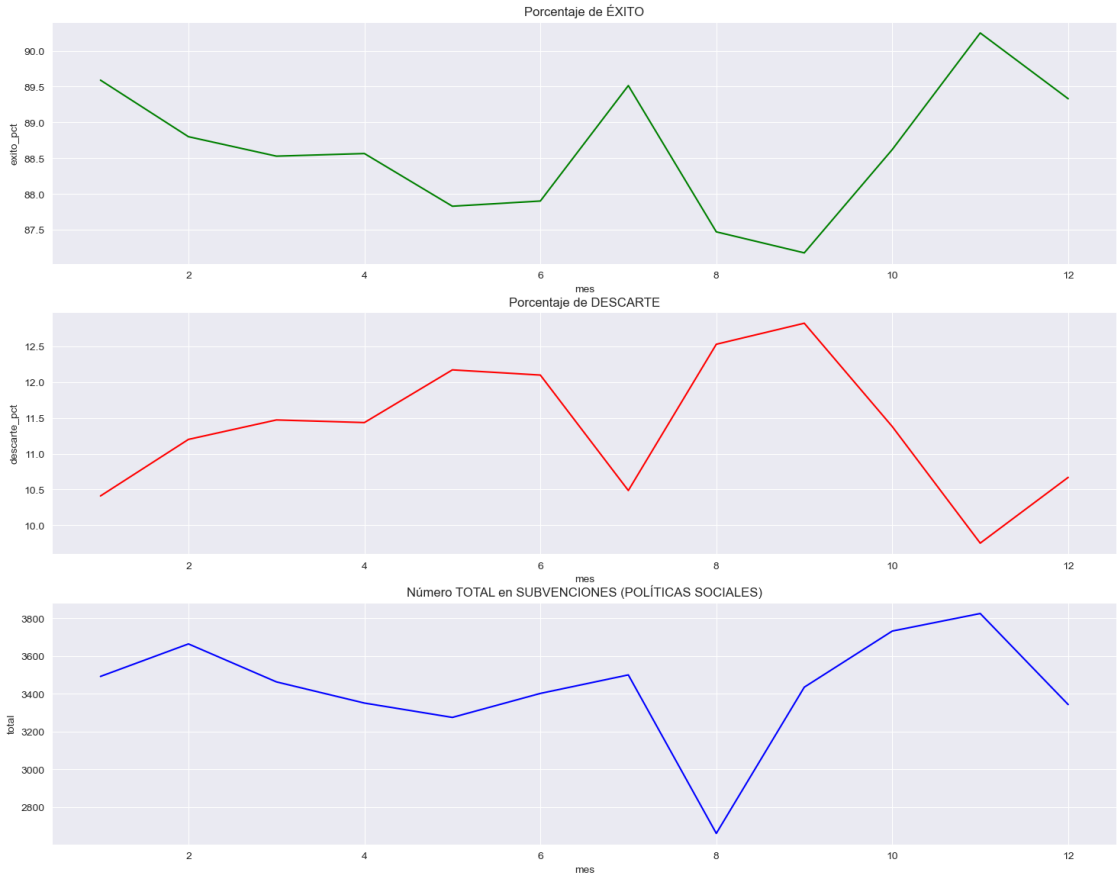
[190]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_digi_est, x=df_digi_est.index, y=df_digi_est.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_digi_est, x=df_digi_est.index, y=df_digi_est.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_digi_est, x=df_digi_est.index, y=df_digi_est.
 ↪total, color='blue') \
                    .set_title(label='Número TOTAL en DIGITALIZACIÓN',␣
 ↪loc='center');
```



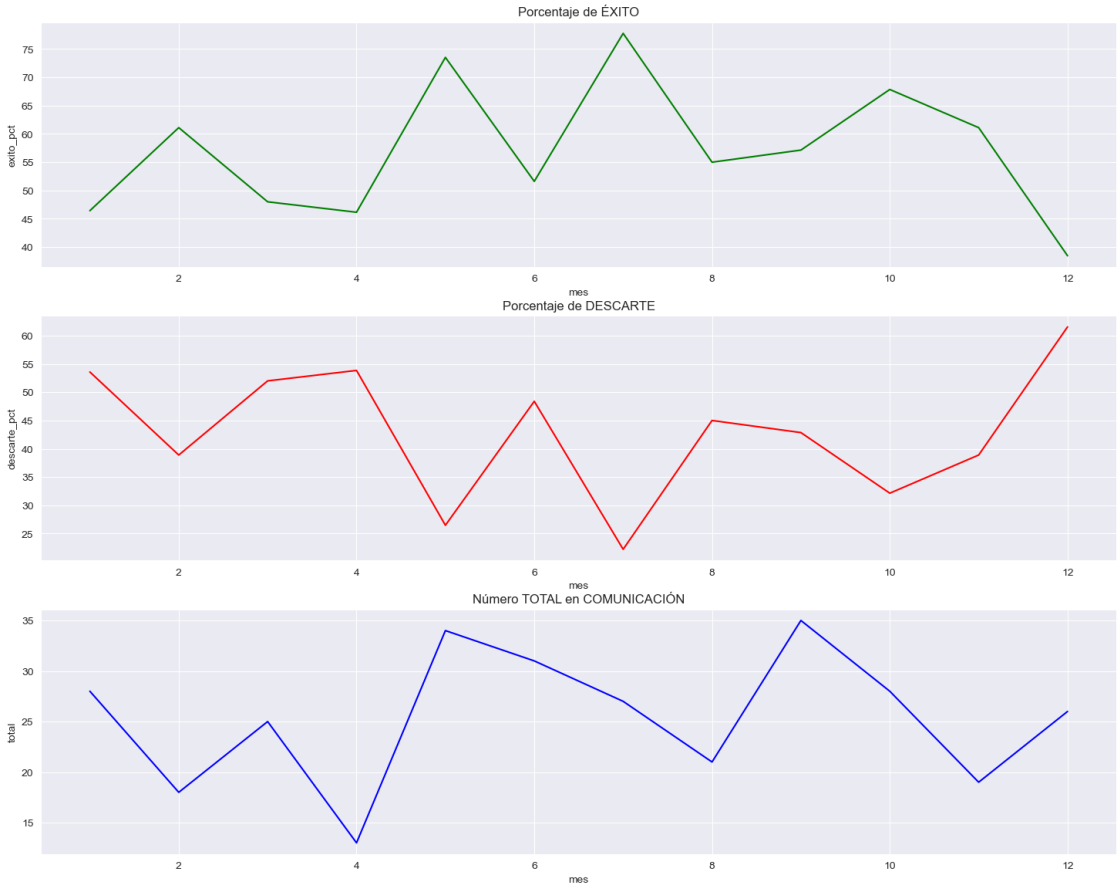
[191]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_educ_est, x=df_educ_est.index, y=df_educ_est.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_educ_est, x=df_educ_est.index, y=df_educ_est.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_educ_est, x=df_educ_est.index, y=df_educ_est.
 ↪total, color='blue') \
                    .set_title(label='Número TOTAL en EDUCACIÓN', loc='center');
```



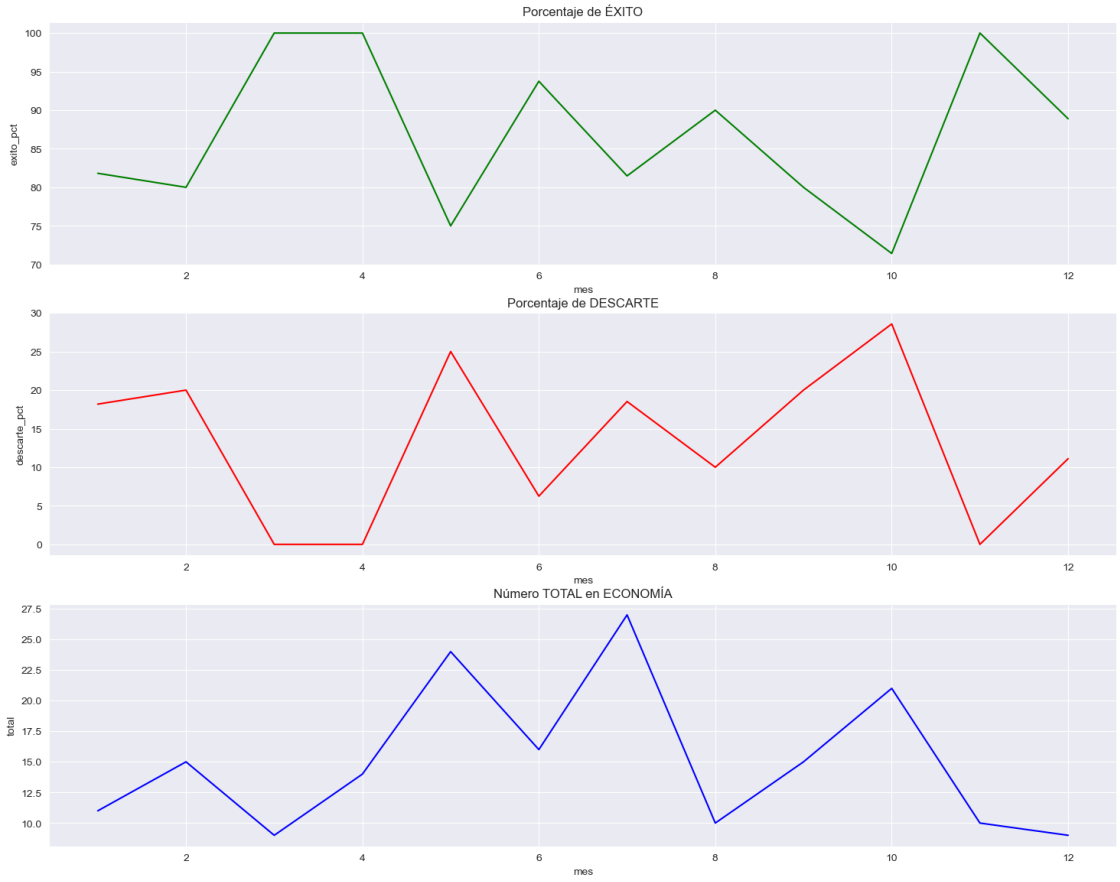
[192]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_subv_est, x=df_subv_est.index, y=df_subv_est.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_subv_est, x=df_subv_est.index, y=df_subv_est.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_subv_est, x=df_subv_est.index, y=df_subv_est.
 ↪total, color='blue') \
                    .set_title(label='Número TOTAL en SUBVENCIONES (POLÍTICAS␣
 ↪SOCIALES)', loc='center');
```



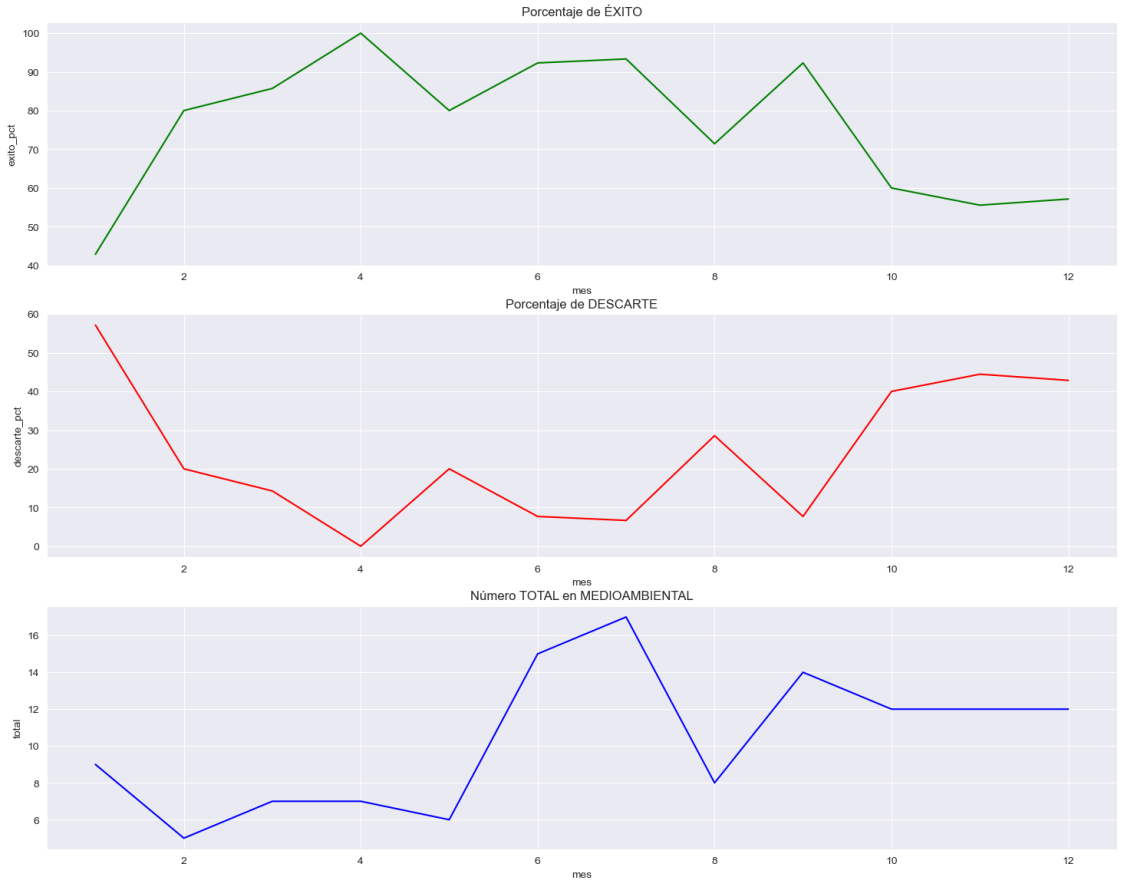
[193]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_comu_est, x=df_comu_est.index, y=df_comu_est.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_comu_est, x=df_comu_est.index, y=df_comu_est.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_comu_est, x=df_comu_est.index, y=df_comu_est.
 ↪total, color='blue') \
                    .set_title(label='Número TOTAL en COMUNICACIÓN',␣
 ↪loc='center');
```



[194]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_econ_est, x=df_econ_est.index, y=df_econ_est.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_econ_est, x=df_econ_est.index, y=df_econ_est.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_econ_est, x=df_econ_est.index, y=df_econ_est.
 ↪total, color='blue') \
                    .set_title(label='Número TOTAL en ECONOMÍA', loc='center');
```



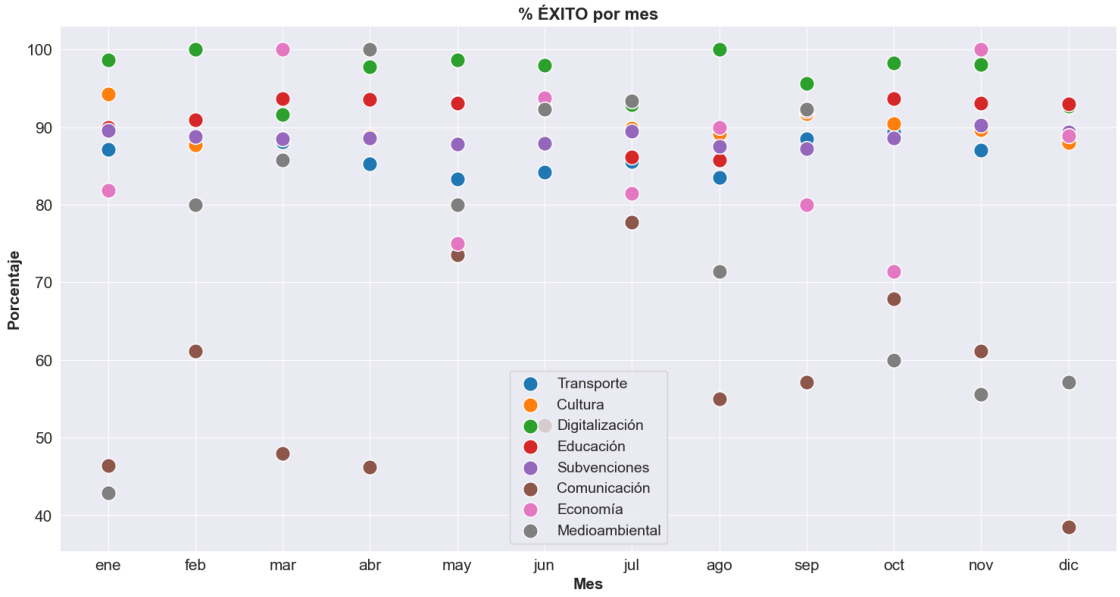
[195]:
```
f, ax = plt.subplots(3,1, figsize=(18,14))
sns.lineplot(ax=ax[0], data=df_medio_est, x=df_medio_est.index, y=df_medio_est.
 ↪exito_pct, color='green') \
                .set_title(label='Porcentaje de ÉXITO', loc='center')

sns.lineplot(ax=ax[1], data=df_medio_est, x=df_medio_est.index, y=df_medio_est.
 ↪descarte_pct, color='red') \
                    .set_title(label='Porcentaje de DESCARTE', loc='center')

sns.lineplot(ax=ax[2], data=df_medio_est, x=df_medio_est.index, y=df_medio_est.
 ↪total, color='blue') \
                    .set_title(label='Número TOTAL en MEDIOAMBIENTAL',␣
 ↪loc='center');
```



GRÁFICOS CONJUNTOS

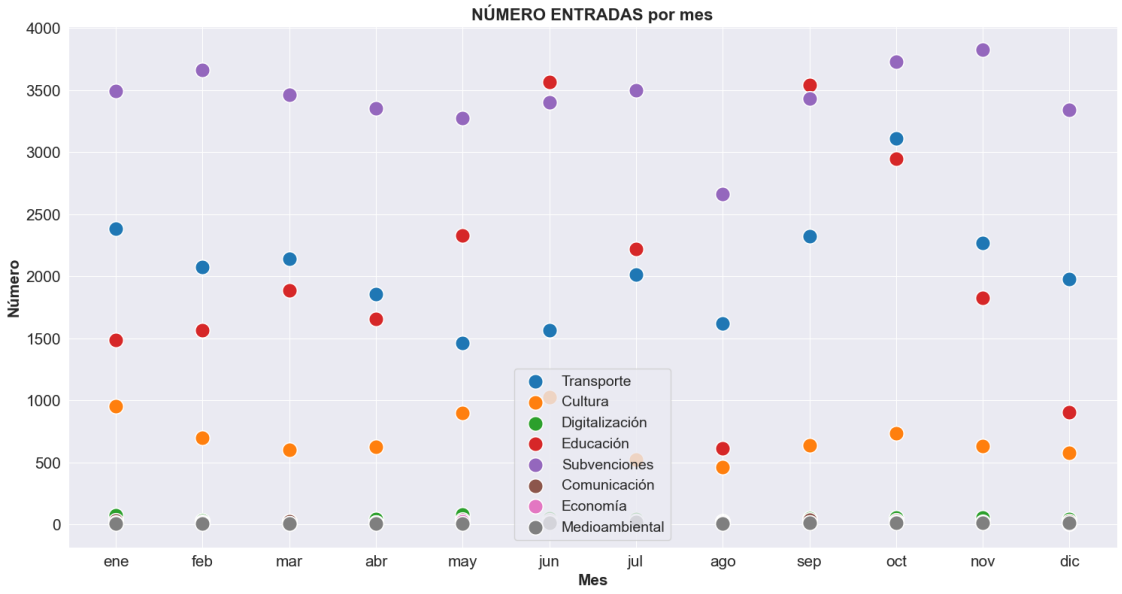
[196]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_trans_est.index, y=df_trans_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_cul_est.index, y=df_cul_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_digi_est.index, y=df_digi_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_educ_est.index, y=df_educ_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_subv_est.index, y=df_subv_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_comu_est.index, y=df_comu_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_econ_est.index, y=df_econ_est.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_medio_est.index, y=df_medio_est.exito_pct, s=200)

ax.set_title('% ÉXITO por mes',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_medio_est.index, labels=['ene', 'feb', 'mar',
```

```
                                                'abr', 'may', 'jun',
                                                'jul', 'ago', 'sep',
                                                'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='lower center', labels=['Transporte', 'Cultura',␣
 ↪'Digitalización', 'Educación',
                                      'Subvenciones', 'Comunicación',␣
 ↪'Economía', 'Medioambiental'], fontsize=14);
```



[197]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_trans_est.index, y=df_trans_est.total, s=200)
sns.scatterplot(ax=ax, x=df_cul_est.index, y=df_cul_est.total, s=200)
sns.scatterplot(ax=ax, x=df_digi_est.index, y=df_digi_est.total, s=200)
sns.scatterplot(ax=ax, x=df_educ_est.index, y=df_educ_est.total, s=200)
sns.scatterplot(ax=ax, x=df_subv_est.index, y=df_subv_est.total, s=200)
sns.scatterplot(ax=ax, x=df_comu_est.index, y=df_comu_est.total, s=200)
sns.scatterplot(ax=ax, x=df_econ_est.index, y=df_econ_est.total, s=200)
sns.scatterplot(ax=ax, x=df_medio_est.index, y=df_medio_est.total, s=200)

ax.set_title('NÚMERO ENTRADAS por mes',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))
```

```
ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_medio_est.index, labels=['ene', 'feb', 'mar',
                                                'abr', 'may', 'jun',
                                                'jul', 'ago', 'sep',
                                                'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='lower center', labels=['Transporte', 'Cultura',␣
 ↪'Digitalización', 'Educación',
                                      'Subvenciones', 'Comunicación',␣
 ↪'Economía', 'Medioambiental'], fontsize=14);
```



4.4.3 Estacionalidad Dia_Semana

- La tasa de éxito suele estar por encima del 87%, excepto en Comunicación que es del 60%.
- Transporte tiene las tasas de éxito más bajas el lunes y el viernes, Digitalización y Educación tiene las tasas de éxito más altas en fin de semana, Economía tiene la tasa de éxito el sábado igual que los días laborables, Medioambiental tiene la tasa de éxito más alta en sábado y la más baja en domingo.
- El número de entradas disminuye a lo largo de la semana siendo el máximo el lunes, excepto para Educación (máximo miércoles) y Medioambiental (máximos en martes y jueves).

[198]:
```
df_compo
```

[198]:

```
                index   tipo                   estado canal_entrada  \
fecha_entrada
2022-06-07          0  Queja  Contestada al Ciudadano      INTERNET
2022-05-27          1  Queja  Recepcionada por Unidad      INTERNET
2022-06-07          2  Queja  Contestada al Ciudadano      INTERNET
…                 …    …                        …            …
2018-02-19      99997  Queja  Contestada al Ciudadano  LLAMADAS 012
2018-03-01      99998  Queja  Contestada al Ciudadano  LLAMADAS 012
2018-03-06      99999  Queja  Contestada al Ciudadano  LLAMADAS 012

               fecha_contestacion  \
fecha_entrada
2022-06-07            2022-06-23
2022-05-27                   NaT
2022-06-07            2022-07-05
…                           …
2018-02-19            2018-02-23
2018-03-01            2018-03-12
2018-03-06            2018-03-14

                                                    consejeria  \
fecha_entrada
2022-06-07     Consejería de Medio Ambiente, Vivienda y Agric…
2022-05-27     Consejería de Medio Ambiente, Vivienda y Agric…
2022-06-07     Consejería de Medio Ambiente, Vivienda y Agric…
…                                                          …
2018-02-19                    Consorcio Regional de Transportes
2018-03-01                    Consorcio Regional de Transportes
2018-03-06                    Consorcio Regional de Transportes

                                          unidad_descripcion         tema  \
fecha_entrada
2022-06-07     D.G. de Descarbonización y Transición Energética      ENERGÍA
2022-05-27     D.G. de Descarbonización y Transición Energética      ENERGÍA
2022-06-07     D.G. de Descarbonización y Transición Energética      ENERGÍA
…                                                         …          …
2018-02-19                                                  NaN  TRANSPORTES
2018-03-01                                                  NaN  TRANSPORTES
2018-03-06                                                  NaN  TRANSPORTES

                terminada  exito  descarte  proceso   ano  mes  dia  dia_ano  \
fecha_entrada
2022-06-07              1      1         0        0  2022    6    7      158
2022-05-27              0      0         0        1  2022    5   27      147
2022-06-07              1      1         0        0  2022    6    7      158
```

```
…                    …    …        …       …   …  …  …      …
2018-02-19             1      1          0         0  2018    2   19        50
2018-03-01             1      1          0         0  2018    3    1        60
2018-03-06             1      1          0         0  2018    3    6        65

               dia_semana  reenviada     nueva_conse
fecha_entrada
2022-06-07              1          0  Medioambiental
2022-05-27              4          0  Medioambiental
2022-06-07              1          0  Medioambiental
…                     …          …              …
2018-02-19              0          0      Transporte
2018-03-01              3          0      Transporte
2018-03-06              1          0      Transporte

[99997 rows x 19 columns]
```

[199]:
```
df_conse_ds = df_compo.reset_index().groupby(['dia_semana', 'nueva_conse']).agg(
    descarte=('descarte', 'sum'),
    exito=('exito', 'sum'),
    proceso=('proceso', 'sum'),
    terminada=('terminada', 'sum'),
    total=('tipo', 'count'))
```

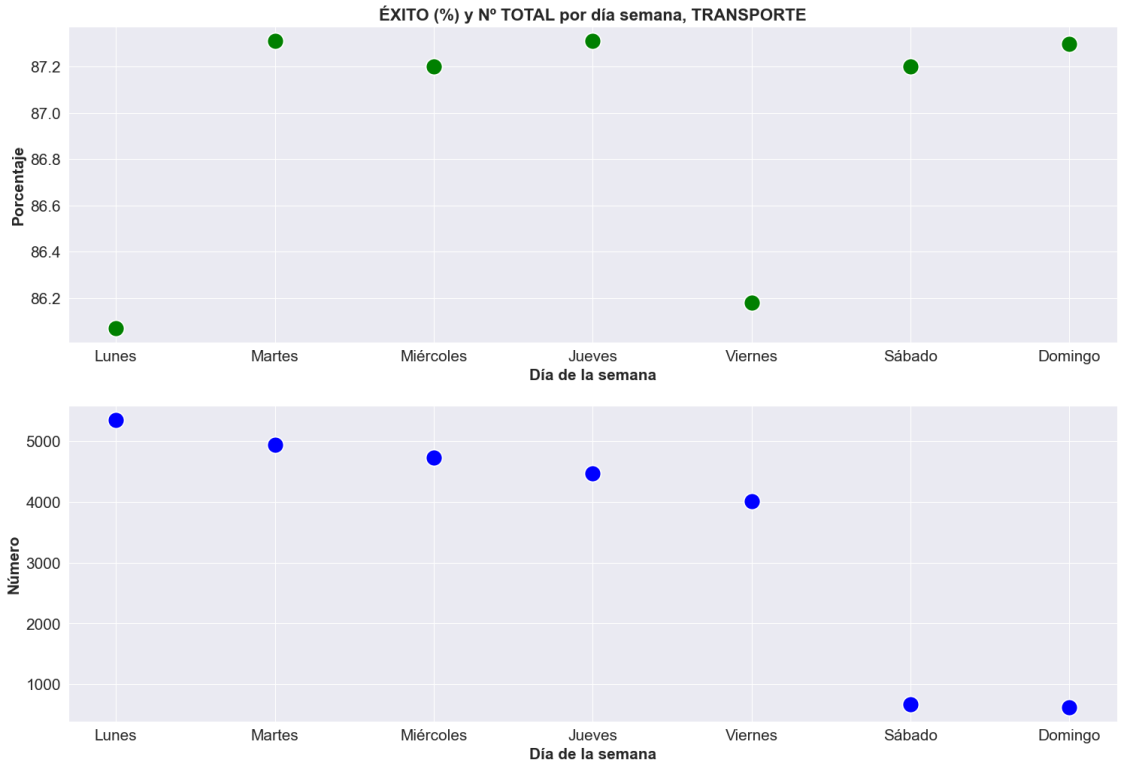
[200]:
```
df_conse_ds = df_conse_ds.reset_index().set_index('dia_semana')
```

[201]:
```
df_conse_ds['terminada'] = df_conse_ds.exito + df_conse_ds.descarte
df_conse_ds['exito_pct'] = 100*df_conse_ds.exito / df_conse_ds.terminada
df_conse_ds['descarte_pct'] = 100*df_conse_ds.descarte / df_conse_ds.terminada
df_conse_ds['terminada_pct'] = 100*df_conse_ds.terminada / (df_conse_ds.
 ↪terminada + df_conse_ds.proceso)
df_conse_ds['proceso_pct'] = 100*df_conse_ds.proceso / (df_conse_ds.terminada +␣
 ↪df_conse_ds.proceso)
df_conse_ds['exitoT_pct'] = 100*df_conse_ds.exito / (df_conse_ds.terminada +␣
 ↪df_conse_ds.proceso)
```

[202]:
```
# Transporte
df_trans_ds = df_conse_ds[df_conse_ds.nueva_conse == 'Transporte']

# Cultura
df_cul_ds = df_conse_ds[df_conse_ds.nueva_conse == 'Cultura']

# Digitalizacion
df_digi_ds = df_conse_ds[df_conse_ds.nueva_conse == 'Digitalizacion']

# Educacion
df_educ_ds = df_conse_ds[df_conse_ds.nueva_conse == 'Educacion']
```

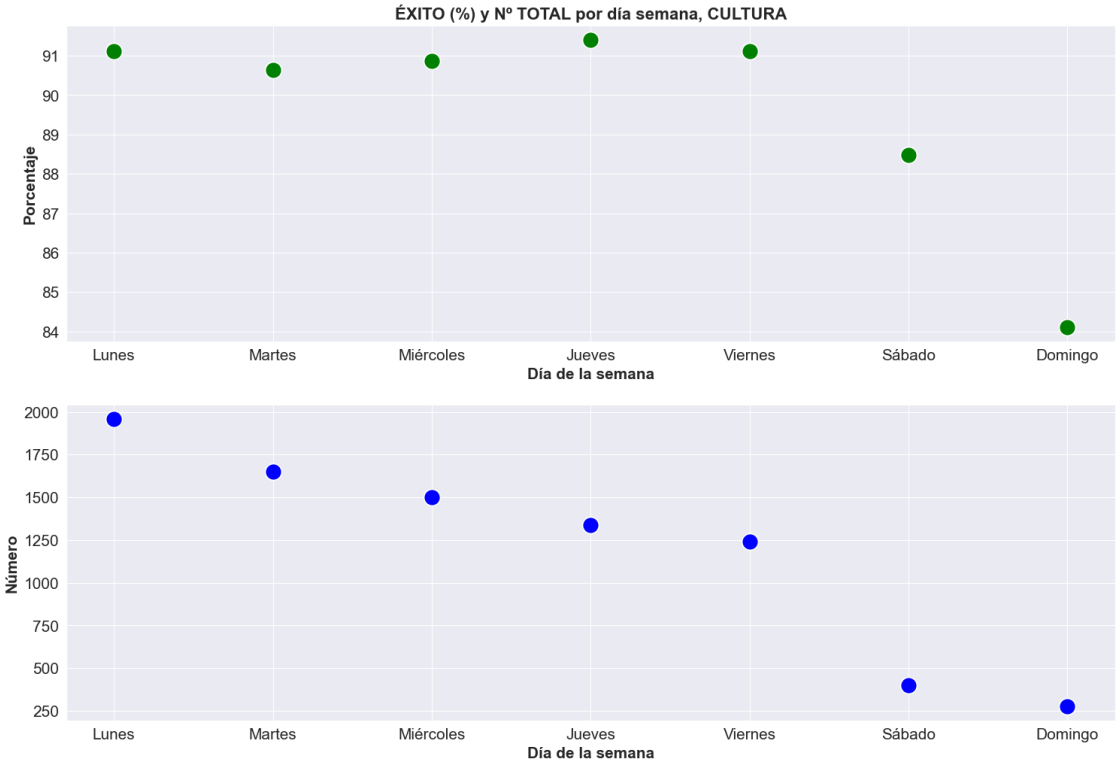
```
# Subvenciones
df_subv_ds = df_conse_ds[df_conse_ds.nueva_conse == 'Subvenciones']

# Comunicacion
df_comu_ds = df_conse_ds[df_conse_ds.nueva_conse == 'Comunicacion']

# Economia
df_econ_ds = df_conse_ds[df_conse_ds.nueva_conse == 'Economia']

# Medioambiental
df_medio_ds = df_conse_ds[df_conse_ds.nueva_conse == 'Medioambiental']
```

[203]:
```
f, ax = plt.subplots(2,1, figsize=(18,12))

sns.scatterplot(ax=ax[0], x=df_trans_ds.index, y=df_trans_ds.exito_pct,␣
 ↪color='green', s=250)
sns.scatterplot(ax=ax[1], x=df_trans_ds.index, y=df_trans_ds.total,␣
 ↪color='blue', s=250)

ax[0].set_title('ÉXITO (%) y Nº TOTAL por día semana, TRANSPORTE',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax[0].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax[1].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax[0].set_xticks(ticks=df_trans_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.
ax[1].set_xticks(ticks=df_trans_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.

ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)

ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[1].tick_params(axis='y', which='major', labelsize=15);
```



[204]:
```
f, ax = plt.subplots(2,1, figsize=(18,12))

sns.scatterplot(ax=ax[0], x=df_cul_ds.index, y=df_cul_ds.exito_pct,␣
 ↪color='green', s=250)
sns.scatterplot(ax=ax[1], x=df_cul_ds.index, y=df_cul_ds.total, color='blue',␣
 ↪s=250)

ax[0].set_title('ÉXITO (%) y Nº TOTAL por día semana, CULTURA',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax[0].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax[1].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax[0].set_xticks(ticks=df_cul_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.
ax[1].set_xticks(ticks=df_cul_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.

ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
```

```
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)

ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[1].tick_params(axis='y', which='major', labelsize=15);
```

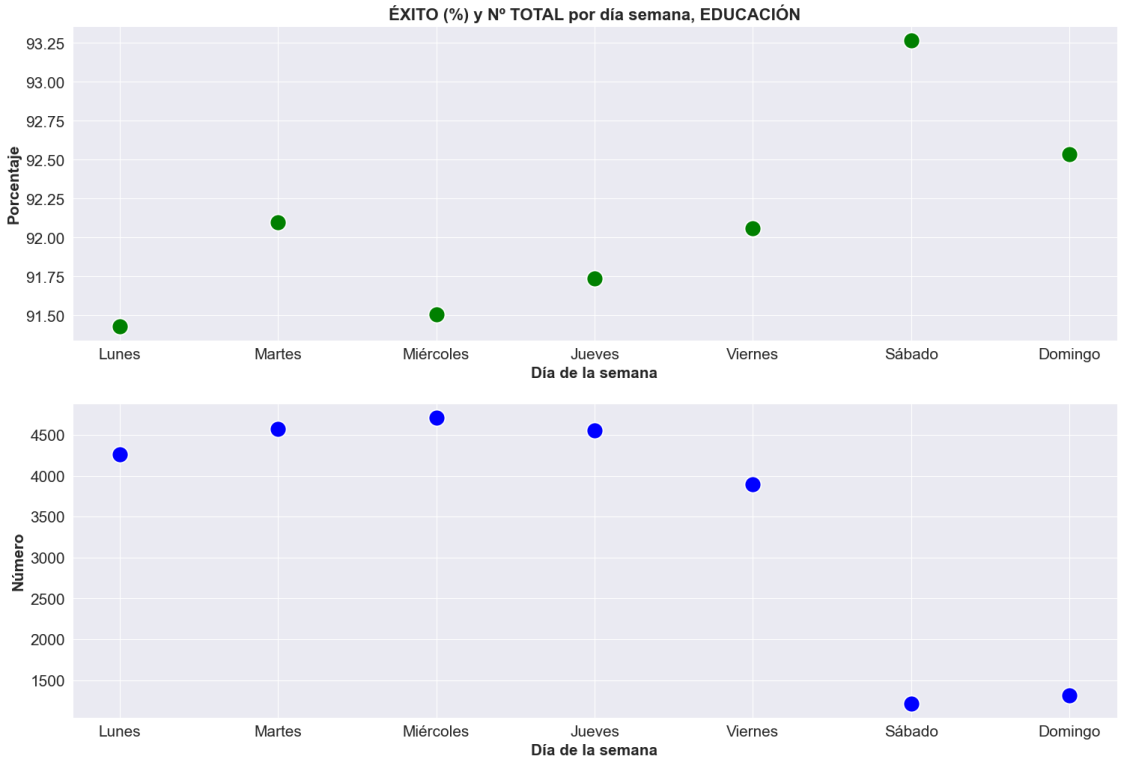


[205]:
```
f, ax = plt.subplots(2,1, figsize=(18,12))

sns.scatterplot(ax=ax[0], x=df_digi_ds.index, y=df_digi_ds.exito_pct,␣
 ↪color='green', s=250)
sns.scatterplot(ax=ax[1], x=df_digi_ds.index, y=df_digi_ds.total, color='blue',␣
 ↪s=250)

ax[0].set_title('ÉXITO (%) y Nº TOTAL por día semana, DIGITALIZACIÓN',
            fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax[0].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax[1].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))
```

```
ax[0].set_xticks(ticks=df_digi_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.
ax[1].set_xticks(ticks=df_digi_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.

ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)

ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[1].tick_params(axis='y', which='major', labelsize=15);
```

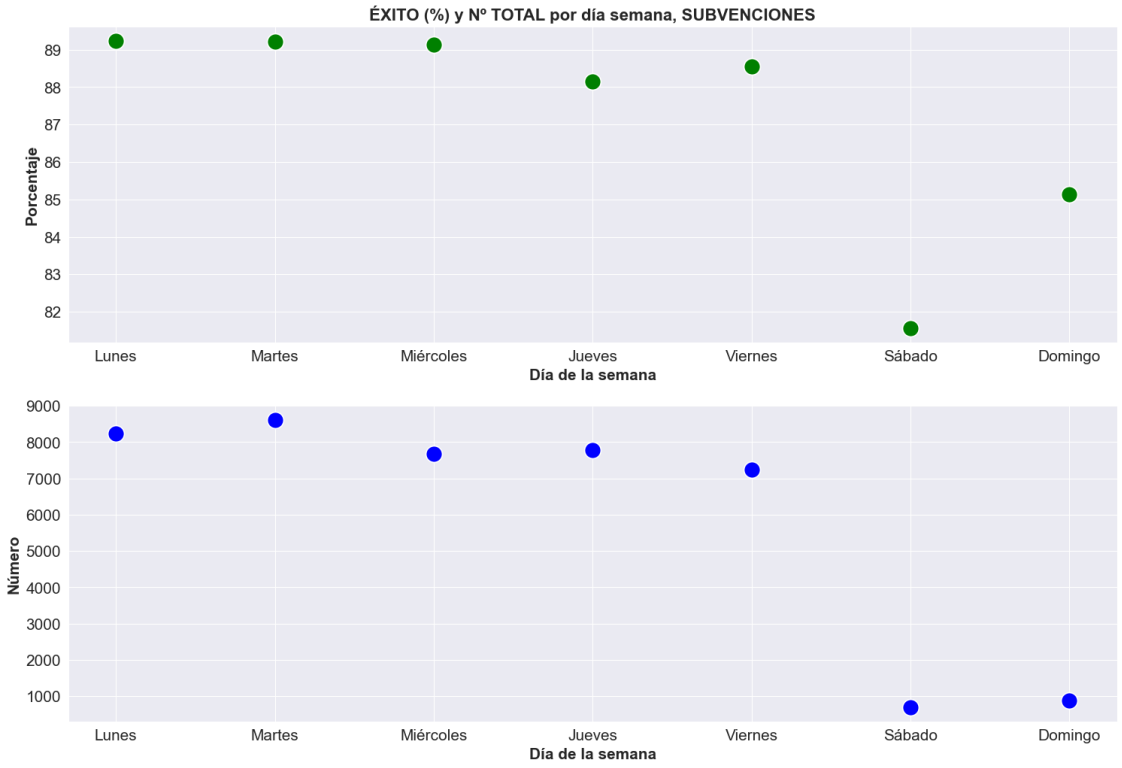
[206]:
```
f, ax = plt.subplots(2,1, figsize=(18,12))

sns.scatterplot(ax=ax[0], x=df_educ_ds.index, y=df_educ_ds.exito_pct,␣
 ↪color='green', s=250)
sns.scatterplot(ax=ax[1], x=df_educ_ds.index, y=df_educ_ds.total, color='blue',␣
 ↪s=250)

ax[0].set_title('ÉXITO (%) y Nº TOTAL por día semana, EDUCACIÓN',
            fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax[0].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
```

```
ax[0].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax[1].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax[0].set_xticks(ticks=df_educ_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.
ax[1].set_xticks(ticks=df_educ_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.

ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)

ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[1].tick_params(axis='y', which='major', labelsize=15);
```



[207]:
```
f, ax = plt.subplots(2,1, figsize=(18,12))

sns.scatterplot(ax=ax[0], x=df_subv_ds.index, y=df_subv_ds.exito_pct,␣
 ↪color='green', s=250)
sns.scatterplot(ax=ax[1], x=df_subv_ds.index, y=df_subv_ds.total, color='blue',␣
 ↪s=250)
```

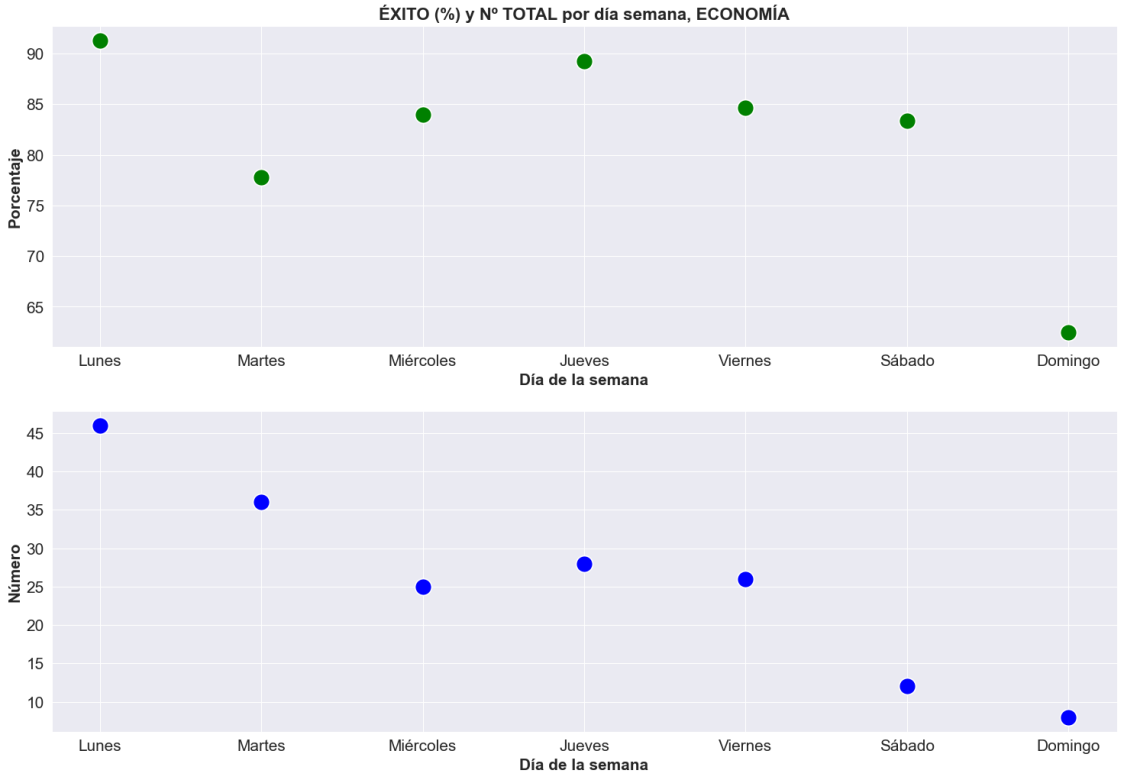
```
ax[0].set_title('ÉXITO (%) y Nº TOTAL por día semana, SUBVENCIONES',
            fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax[0].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax[1].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax[0].set_xticks(ticks=df_subv_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.
ax[1].set_xticks(ticks=df_subv_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.

ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)

ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[1].tick_params(axis='y', which='major', labelsize=15);
```



[208]:
```
f, ax = plt.subplots(2,1, figsize=(18,12))

sns.scatterplot(ax=ax[0], x=df_comu_ds.index, y=df_comu_ds.exito_pct,␣
 ↪color='green', s=250)
sns.scatterplot(ax=ax[1], x=df_comu_ds.index, y=df_comu_ds.total, color='blue',␣
 ↪s=250)

ax[0].set_title('ÉXITO (%) y Nº TOTAL por día semana, COMUNICACIÓN',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax[0].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax[1].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax[0].set_xticks(ticks=df_comu_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.
ax[1].set_xticks(ticks=df_comu_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.

ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)

ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[1].tick_params(axis='y', which='major', labelsize=15);
```

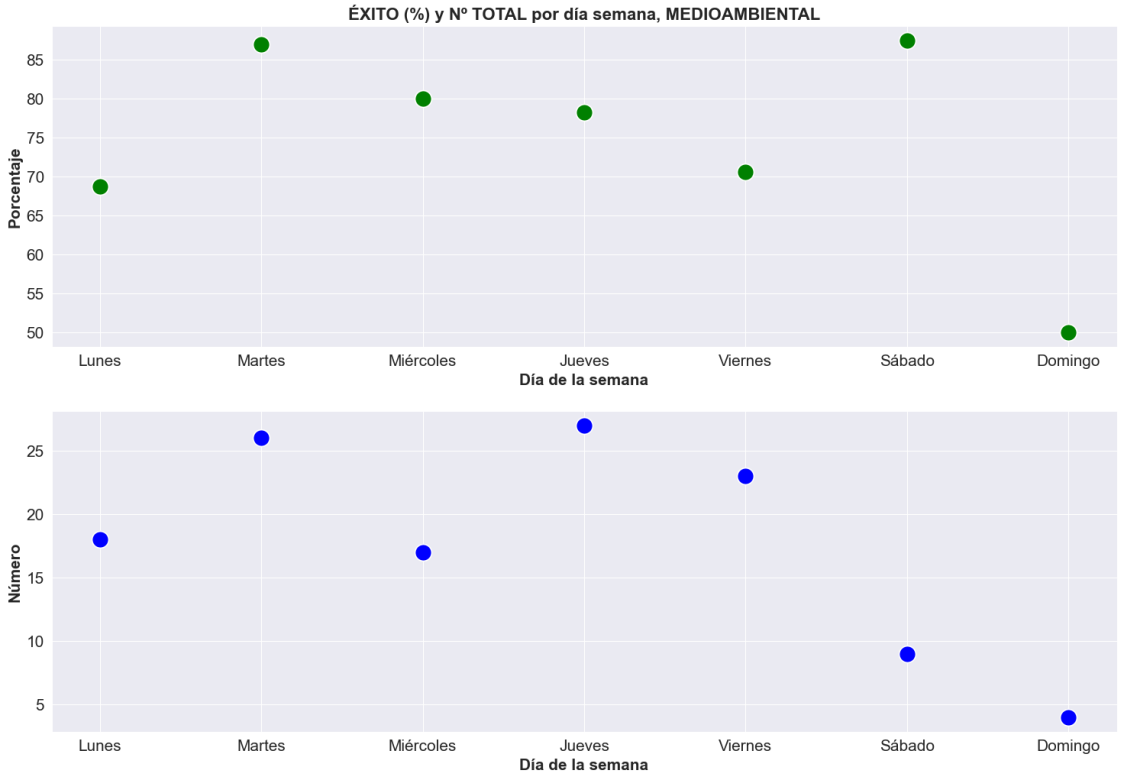
[209]:
```
f, ax = plt.subplots(2,1, figsize=(18,12))

sns.scatterplot(ax=ax[0], x=df_econ_ds.index, y=df_econ_ds.exito_pct,␣
 ↪color='green', s=250)
sns.scatterplot(ax=ax[1], x=df_econ_ds.index, y=df_econ_ds.total, color='blue',␣
 ↪s=250)

ax[0].set_title('ÉXITO (%) y Nº TOTAL por día semana, ECONOMÍA',
            fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax[0].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax[1].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax[0].set_xticks(ticks=df_econ_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.
ax[1].set_xticks(ticks=df_econ_ds.index, labels=etiquetas)  # Solo para poner␣
 ↪las etiquetas al eje x.

ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
```

```
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)

ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[1].tick_params(axis='y', which='major', labelsize=15);
```



[210]:
```
f, ax = plt.subplots(2,1, figsize=(18,12))

sns.scatterplot(ax=ax[0], x=df_medio_ds.index, y=df_medio_ds.exito_pct,␣
 ↪color='green', s=250)
sns.scatterplot(ax=ax[1], x=df_medio_ds.index, y=df_medio_ds.total,␣
 ↪color='blue', s=250)

ax[0].set_title('ÉXITO (%) y Nº TOTAL por día semana, MEDIOAMBIENTAL',
            fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax[0].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax[1].set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))
```

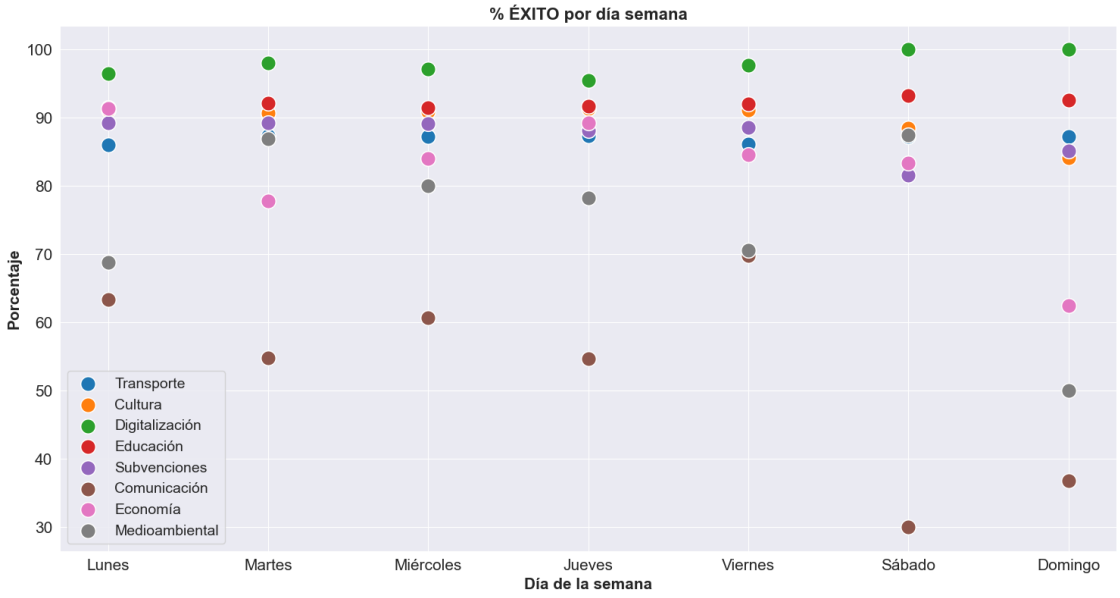
```
ax[0].set_xticks(ticks=df_medio_ds.index, labels=etiquetas)  # Solo para poner
 ↪las etiquetas al eje x.
ax[1].set_xticks(ticks=df_medio_ds.index, labels=etiquetas)  # Solo para poner
 ↪las etiquetas al eje x.

ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)

ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[1].tick_params(axis='y', which='major', labelsize=15);
```



GRÁFICOS CONJUNTO RESUMEN

[211]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_trans_ds.index, y=df_trans_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_cul_ds.index, y=df_cul_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_digi_ds.index, y=df_digi_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_educ_ds.index, y=df_educ_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_subv_ds.index, y=df_subv_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_comu_ds.index, y=df_comu_ds.exito_pct, s=200)
sns.scatterplot(ax=ax, x=df_econ_ds.index, y=df_econ_ds.exito_pct, s=200)
```

```
sns.scatterplot(ax=ax, x=df_medio_ds.index, y=df_medio_ds.exito_pct, s=200)

ax.set_title('% ÉXITO por día semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Porcentaje', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_medio_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='lower left', labels=['Transporte', 'Cultura', 'Digitalización',␣
 ↪'Educación',
                                    'Subvenciones', 'Comunicación',␣
 ↪'Economía', 'Medioambiental'], fontsize=14);
```



[212]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=df_trans_ds.index, y=df_trans_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_cul_ds.index, y=df_cul_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_digi_ds.index, y=df_digi_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_educ_ds.index, y=df_educ_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_subv_ds.index, y=df_subv_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_comu_ds.index, y=df_comu_ds.total, s=200)
```

```
sns.scatterplot(ax=ax, x=df_econ_ds.index, y=df_econ_ds.total, s=200)
sns.scatterplot(ax=ax, x=df_medio_ds.index, y=df_medio_ds.total, s=200)

ax.set_title('% ÉXITO por día semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día de la semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=df_medio_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Transporte', 'Cultura', 'Digitalización',␣
 ↪'Educación',
                                     'Subvenciones', 'Comunicación',␣
 ↪'Economía', 'Medioambiental'], fontsize=14);
```

5 Análisis tiempo de respuesta

Aquí hay que calcular el tiempo de respuesta (fecha_contestación - fecha_entrada) y repetir el análisis temporal y no temporal.

[213]:
```
# Copia df para df de tiempo respuesta
df_tr = df_compo.reset_index().copy()

# Eliminar NaT en columnas fecha_entrada y fecha_contestacion
df_tr.drop(df_tr[df_tr.fecha_entrada.isna()].index, inplace=True)
df_tr.drop(df_tr[df_tr.fecha_contestacion.isna()].index, inplace=True)

# Tiempo de respuesta en formato timedelta
df_tr['tiempo_respuesta'] = df_tr.fecha_contestacion - df_tr.fecha_entrada

# Tiempo de respuesta en número entero
df_tr['tiempo_respuesta'] = df_tr.tiempo_respuesta.dt.days
```

[214]:
```
# Fecha de entrada como índice del dataframe duración
df_tr = df_tr.set_index('fecha_entrada')
```

5.1 Tiempo de respuesta a lo largo de los años

5.1.1 Díario

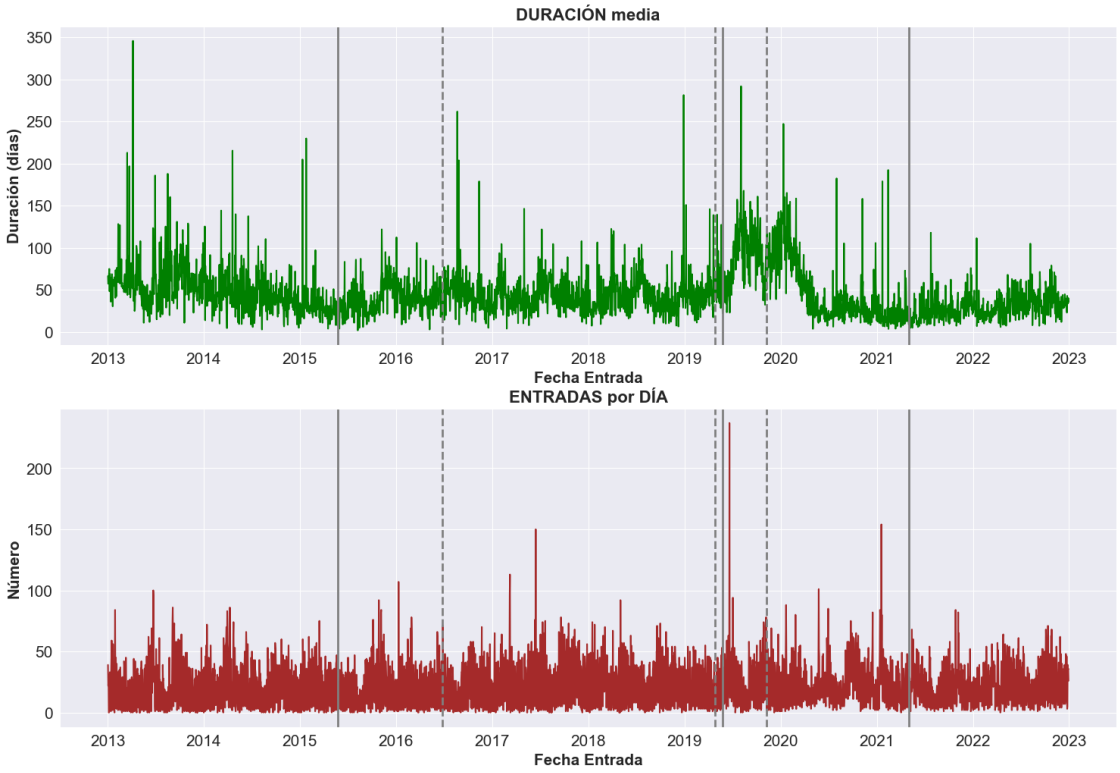
- A mitad de 2019 hay un aumento anormal en la duración media, y a partir de 2020 la duración media cae incluso por debajo de la anterior a 2019.
- En el número de entradas no hay existe un cambio brusco semejante.

[215]:
```
# Elecciones Generales y En la CAM.
e2021_may_cam = pd.to_datetime('2021-05-04')
e2019_abr = pd.to_datetime('2019-04-28')
e2019_nov = pd.to_datetime('2019-11-10')
e2019_may_cam = pd.to_datetime('2019-05-26')
e2016_jun = pd.to_datetime('2016-06-26')
e2015_may_cam = pd.to_datetime('2015-05-24')
```

[216]:
```
# Dataframe con métricas DIARÍAS de eficiencia del proceso incluyendo DURACIÓN
dfd = df_tr.resample('D').agg(terminada=('terminada', 'sum'),
                                        exito=('exito', 'sum'),
                                        descarte=('descarte', 'sum'),
                                        proceso=('proceso', 'sum'),
                                        reenviada=('reenviada', 'sum'),
                                        total=('tipo', 'count'),
                                        duracion=('tiempo_respuesta', 'mean'))
```

[217]:
```
f, ax = plt.subplots(2,1, figsize=(18,12))

sns.lineplot(ax=ax[0], data=dfd, x=dfd.index, y=dfd.duracion, color='green')
sns.lineplot(ax=ax[1], data=dfd, x=dfd.index, y=dfd.total, color='brown')

ax[0].set_title('DURACIÓN media', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
```

```
ax[0].set_xlabel("Fecha Entrada", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Duración (días)', fontsize=15, fontdict=dict(weight='bold'))
ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[0].tick_params(axis='y', which='major', labelsize=15)

ax[0].axvline(x=e2019_abr, color='grey', lw=2, ls='--')
ax[0].axvline(x=e2019_may_cam, color='grey', lw=2, ls='-')
ax[0].axvline(x=e2019_nov, color='grey', lw=2, ls='--')
ax[0].axvline(x=e2016_jun, color='grey', lw=2, ls='--')
ax[0].axvline(x=e2015_may_cam, color='grey', lw=2, ls='-')
ax[0].axvline(x=e2021_may_cam, color='grey', lw=2, ls='-')

ax[1].set_title('ENTRADAS por DÍA', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[1].set_xlabel("Fecha Entrada", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[1].tick_params(axis='y', which='major', labelsize=15)

ax[1].axvline(x=e2019_abr, color='grey', lw=2, ls='--')
ax[1].axvline(x=e2019_may_cam, color='grey', lw=2, ls='-')
ax[1].axvline(x=e2019_nov, color='grey', lw=2, ls='--')
ax[1].axvline(x=e2016_jun, color='grey', lw=2, ls='--')
ax[1].axvline(x=e2015_may_cam, color='grey', lw=2, ls='-')
ax[1].axvline(x=e2021_may_cam, color='grey', lw=2, ls='-');
```



5.1.2 Mensual

- La duración media (medida mensualmente) es 45 días.
- Desde mitad del 2019 hasta principios del 2020 la duración media se eleva anormalmente. Algo que también se observaba en la periodicidad diaria.
- El cambio puede deberse a un comportamiento estacional o a las elecciones (línea continua para autonómicas; discontinua para generales).

[218]:
```
# Dataframe con métricas MENSUALES de eficiencia del proceso incluyendo DURACIÓN
dfm = df_tr.resample('M').agg(terminada=('terminada', 'sum'),
                                        exito=('exito', 'sum'),
                                        descarte=('descarte', 'sum'),
                                        proceso=('proceso', 'sum'),
                                        reenviada=('reenviada', 'sum'),
                                        total=('tipo', 'count'),
                                        duracion=('tiempo_respuesta', 'mean'))
```

[219]:
```
f, ax = plt.subplots(2,1, figsize=(18,16))

sns.lineplot(ax=ax[0], data=dfm, x=dfm.index, y=dfm.duracion, color='green')
sns.lineplot(ax=ax[1], data=dfm, x=dfm.index, y=dfm.total, color='brown')

ax[0].set_title('DURACIÓN media', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[0].set_xlabel("Fecha Entrada", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Duración (días)', fontsize=15, fontdict=dict(weight='bold'))
ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[0].tick_params(axis='y', which='major', labelsize=15)

ax[0].axvline(x=e2019_abr, color='grey', lw=2, ls='--')
ax[0].axvline(x=e2019_may_cam, color='grey', lw=2, ls='-')
ax[0].axvline(x=e2019_nov, color='grey', lw=2, ls='--')
ax[0].axvline(x=e2016_jun, color='grey', lw=2, ls='--')
ax[0].axvline(x=e2015_may_cam, color='grey', lw=2, ls='-')
ax[0].axvline(x=e2021_may_cam, color='grey', lw=2, ls='-')

ax[1].set_title('ENTRADAS por MES', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[1].set_xlabel("Fecha Entrada", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[1].tick_params(axis='y', which='major', labelsize=15)

ax[1].axvline(x=e2019_abr, color='grey', lw=2, ls='--')
```

```
ax[1].axvline(x=e2019_may_cam, color='grey', lw=2, ls='-')
ax[1].axvline(x=e2019_nov, color='grey', lw=2, ls='--')
ax[1].axvline(x=e2016_jun, color='grey', lw=2, ls='--')
ax[1].axvline(x=e2015_may_cam, color='grey', lw=2, ls='-')
ax[1].axvline(x=e2021_may_cam, color='grey', lw=2, ls='-');
```

5.2 Estacionalidad

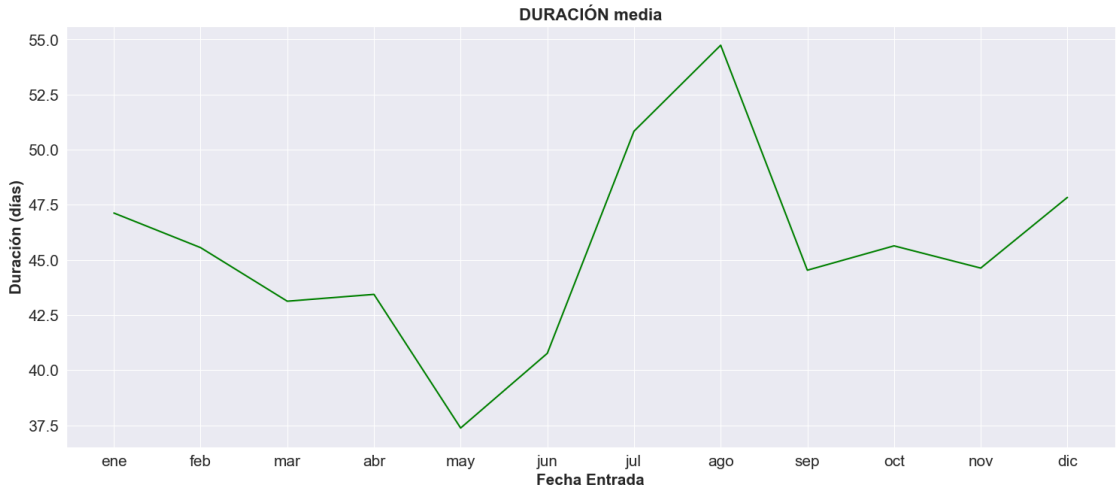
5.2.1 Año

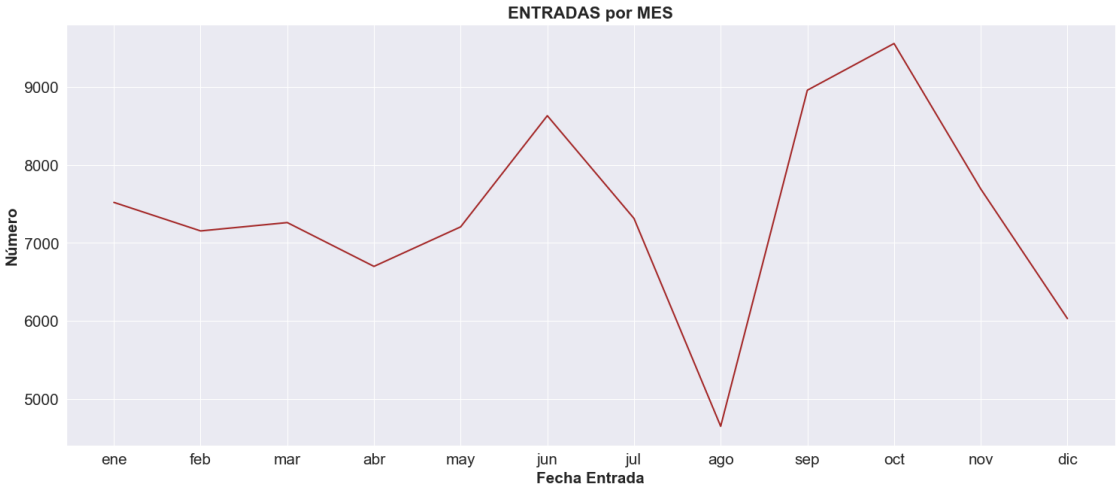
- La duración oscila a lo largo del año entre 37,5 días (para los expediente que entran en mayo) y 55 días (para los expedientes que entran en agosto).
- La duración mejora hasta mayo a pesar de que el número de entradas se mantiene estable.
- Por otro lado, la duración empeora en junio, pero sobre todo en julio y agosto (verano), a pesar de que el número de entradas se reduce casi a la mitad.
- A partir de septiembre, la duración se mantiene estable entorno a los 46 días.
- El cambio que se observaba en el gráfico de evolución histórica de la duración parece explicarse

por este **comportamiento estacional en el mes de mayo**.

[220]:
```
# Dataframe estacional a intervalos de mes
dfm_e = df_tr.reset_index().groupby(['mes']).agg(terminada=('terminada', 'sum'),
                                                 exito=('exito', 'sum'),
                                                 descarte=('descarte', 'sum'),
                                                 proceso=('proceso', 'sum'),
                                                 reenviada=('reenviada', 'sum'),
                                                 total=('tipo', 'count'),
                                                 duracion=('tiempo_respuesta',␣
 ↪'mean'))
```

[221]:
```
f, ax = plt.subplots(2,1, figsize=(18,16))

sns.lineplot(ax=ax[0], data=dfm_e, x=dfm_e.index, y=dfm_e.duracion,␣
 ↪color='green')
sns.lineplot(ax=ax[1], data=dfm_e, x=dfm_e.index, y=dfm_e.total, color='brown')

ax[0].set_title('DURACIÓN media', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[0].set_xlabel("Fecha Entrada", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Duración (días)', fontsize=15, fontdict=dict(weight='bold'))
ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[0].set_xticks(ticks=dfm_e.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo para␣
 ↪poner las etiquetas al eje x.

ax[1].set_title('ENTRADAS por MES', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[1].set_xlabel("Fecha Entrada", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[1].tick_params(axis='y', which='major', labelsize=15)
ax[1].set_xticks(ticks=dfm_e.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic']);

# plt.savefig("multimedia/tiempo_respuesta_estacional_anio.png")
```





5.2.2 Día Semana

- La duración de los expedientes que entran en fin de semana (sábado y domingo) es la más baja, ya que se supone que son contestados inmediatamente comienza la semana.
- El máximo de duración se encuentra en el jueves, a pesar de que el número de entradas a lo largo de la semana va disminuyendo. Parece que hay un efecto saturación a medida que entran expedientes a lo largo de la semana, sin embargo, la oscilación es solo de 44 a 46,5 días de lunes a viernes, mucho menor que la oscilación mensual.

[222]:
```
# Dataframe estacional a intervalos de mes
dfds_e = df_tr.reset_index().groupby(['dia_semana']).
 ↪agg(terminada=('terminada', 'sum'),
                                                    exito=('exito', 'sum'),
                                                    descarte=('descarte',␣
 ↪'sum'),
```

```
                                                    proceso=('proceso',␣
↪'sum'),
                                                   ␣
↪reenviada=('reenviada', 'sum'),
                                                    total=('tipo',␣
↪'count'),
                                                   ␣
↪duracion=('tiempo_respuesta', 'mean'))
```

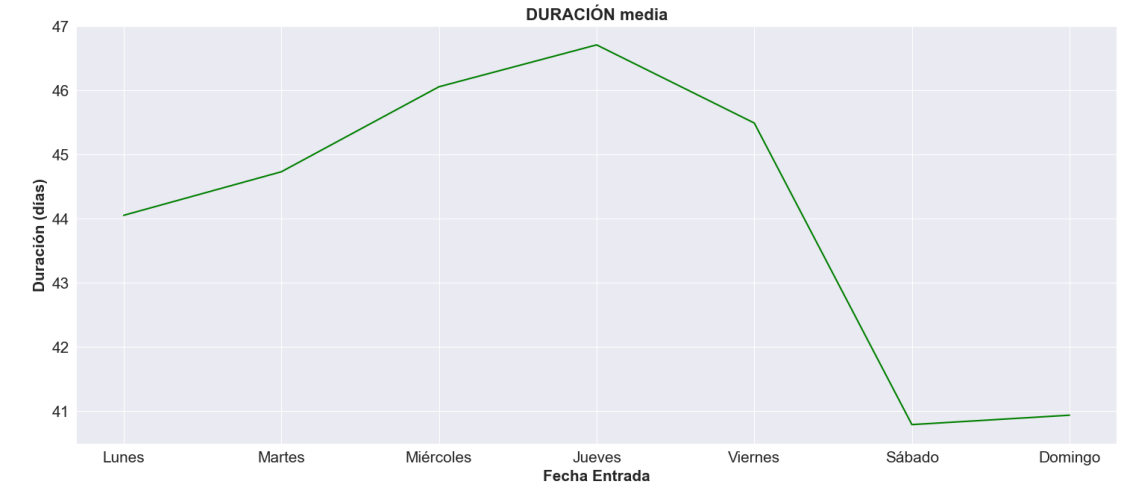
[223]:
```
dfds_e
```

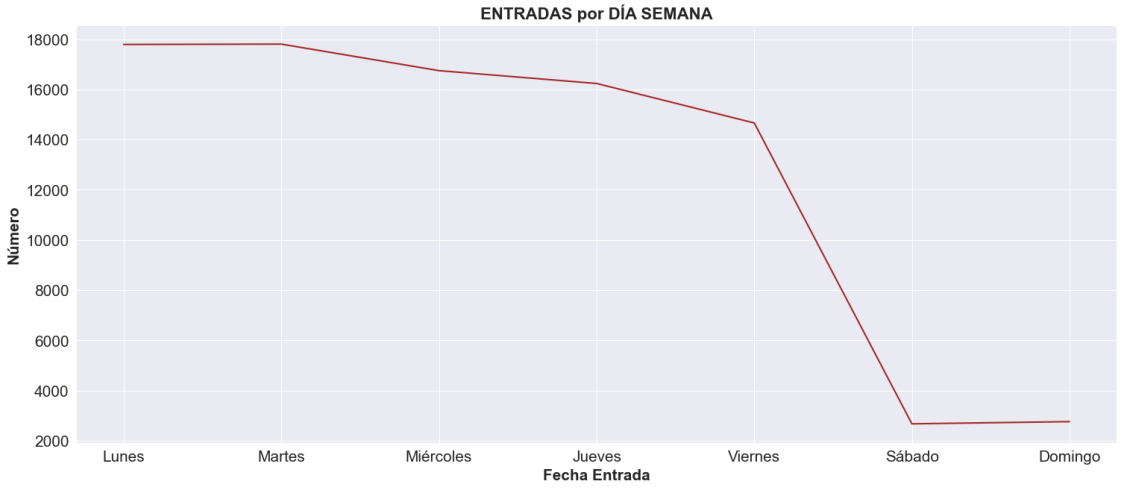
[223]:

dia_semana	terminada	exito	descarte	proceso	reenviada	total	duracion
0	17789	17786	3	0	2	17789	44.051830
1	17802	17797	5	0	4	17802	44.730648
2	16745	16736	9	0	5	16745	46.056256
3	16234	16229	5	0	4	16234	46.709252
4	14661	14658	3	0	3	14661	45.491167
5	2678	2678	0	0	0	2678	40.788648
6	2765	2763	2	0	1	2765	40.935262

[224]:
```
f, ax = plt.subplots(2,1, figsize=(18,16))

sns.lineplot(ax=ax[0], data=dfds_e, x=dfds_e.index, y=dfds_e.duracion,␣
↪color='green')
sns.lineplot(ax=ax[1], data=dfds_e, x=dfds_e.index, y=dfds_e.total,␣
↪color='brown')

ax[0].set_title('DURACIÓN media', fontsize=16, loc='center',␣
↪fontdict=dict(weight='bold'))
ax[0].set_xlabel("Fecha Entrada", fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Duración (días)', fontsize=15, fontdict=dict(weight='bold'))
ax[0].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[0].set_xticks(ticks=dfds_e.index, labels=['Lunes', 'Martes', 'Miércoles',␣
↪'Jueves',
                                             'Viernes', 'Sábado', 'Domingo'])

ax[1].set_title('ENTRADAS por DÍA SEMANA', fontsize=16, loc='center',␣
↪fontdict=dict(weight='bold'))
ax[1].set_xlabel("Fecha Entrada", fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax[1].tick_params(axis='y', which='major', labelsize=15)
ax[1].set_xticks(ticks=dfds_e.index, labels=['Lunes', 'Martes', 'Miércoles',␣
↪'Jueves',
```

```
                                        'Viernes', 'Sábado', 'Domingo']);
```





5.3 Variables categóricas - Sin Temporalidad

- **TIPO**. El tiempo medio de respuesta de Agradecimiento (28) es más bajo que el de Queja (45) o Sugerencia (40).
- **TEMA**. Políticas Sociales y Transportes son más lentos en responder (53 y casi 52, respectivamente). Energía, Educación y Cultura están entre 28 y 34 (34, 31 y 28, respectivamente).
- **CANAL DE ENTRADA**. Examinando las 3 principales (Llamadas 012, Presencial e Internet), vemos que los 2 primeros canales tienen tiempos de respueta muy parecidos (48 y 47), mientras que Internet es algo inferior (casi 42).
- **CONSEJERÍA**. Horquilla amplia entre el máximo (Consejería Medio Ambiente, 86) y el mínimo (Consejería de Educación, 25).
- **CONSEJERÍAS AGRUPADAS - NUEVA CONSE**. Hay 4 grupos:
- Medioambiental, 78.

- Economía, Subvenciones (Políticas Sociales) y Transporte, casi 57, 53 y 52 días, respectivamente.
- Comunicación, 41.
- Educación, Cultura y Digitalización, 31, 28 y 28.

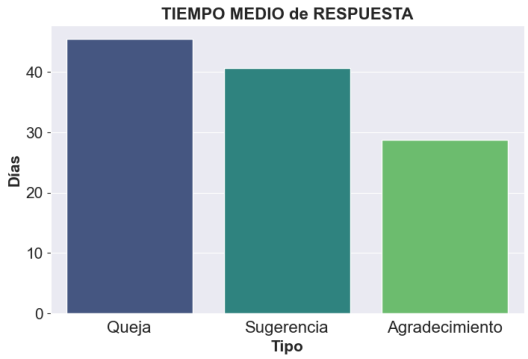
Hay bastante correspondencia entre TEMA y NUEVA CONSEJERÍA: Políticas Sociales, Transporte y Educación, Cultura.

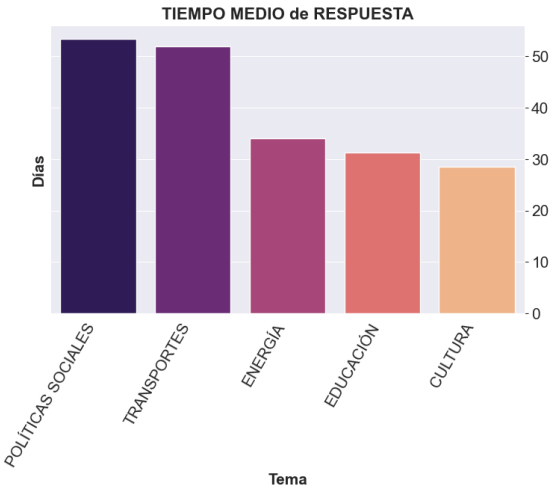
[225]:
```
dftr = df_tr.reset_index()
```

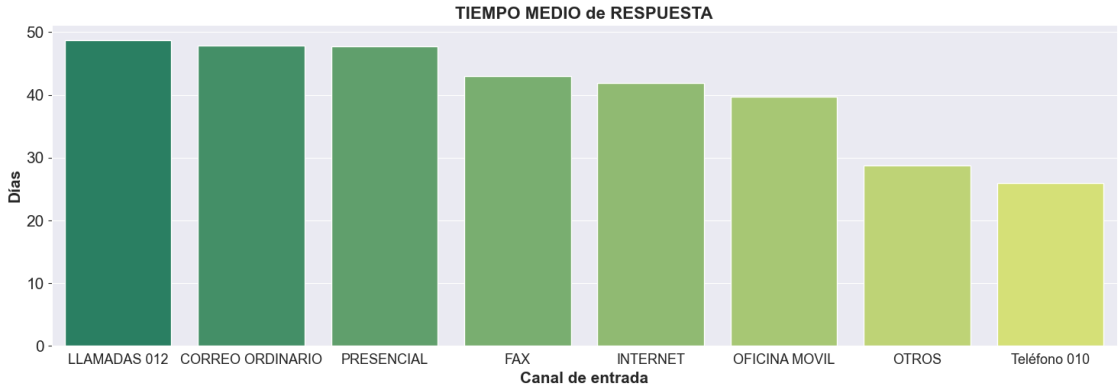
[226]:
```
# Gráficos Tiempo de respuesta por variables categóricas

f, ax = plt.subplots(1, 2, figsize=(18,5))

# TIPO
sns.barplot(ax=ax[0], x=dftr.groupby('tipo')['tiempo_respuesta'].mean().index,
            y=dftr.groupby('tipo')['tiempo_respuesta'].mean().values,
            palette='viridis',
            order=dftr.groupby('tipo')['tiempo_respuesta'].mean().
 ↪sort_values(ascending=False).index)

ax[0].set_title('TIEMPO MEDIO de RESPUESTA', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[0].set_xlabel('Tipo', loc=None, fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))
ax[0].tick_params(axis='x', which='major', labelsize=16, labelrotation=0)

ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[0].yaxis.tick_left()  # Donde se ponen las marcas de eje

# TEMA
sns.barplot(ax=ax[1], x=dftr.groupby('tema')['tiempo_respuesta'].mean().index,
             y=dftr.groupby('tema')['tiempo_respuesta'].mean().values,
             palette='magma',
             order=dftr.groupby('tema')['tiempo_respuesta'].mean().
 ↪sort_values(ascending=False).index)

ax[1].set_title('TIEMPO MEDIO de RESPUESTA', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[1].set_xlabel('Tema', fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))
ax[1].tick_params(axis='x', which='major', labelsize=15, labelrotation=60)
ax[1].set_xticks(ax[1].get_xticks(), ax[1].get_xticklabels(), ha='right')
ax[1].tick_params(axis='y', which='major', labelsize=15)
ax[1].yaxis.tick_right();  # Donde se ponen las marcas de eje

###
```

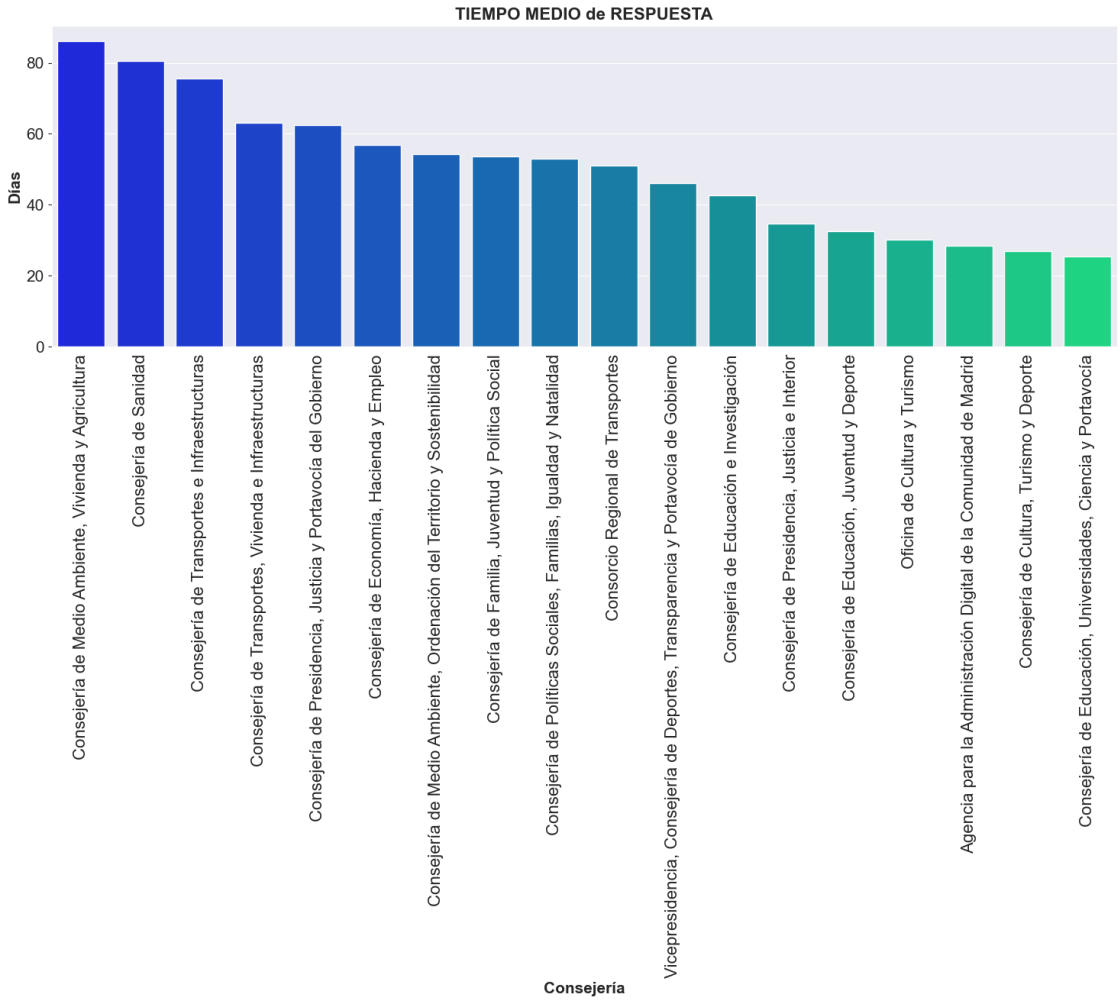
```
f, ax = plt.subplots(2, 1, figsize=(18,12))

# CANAL_ENTRADA
sns.barplot(ax=ax[0], x=dftr.groupby('canal_entrada')['tiempo_respuesta'].
 ↪mean().index,
            y=dftr.groupby('canal_entrada')['tiempo_respuesta'].mean().values,
            palette='summer',
            order=dftr.groupby('canal_entrada')['tiempo_respuesta'].mean().
 ↪sort_values(ascending=False).index)

ax[0].set_title('TIEMPO MEDIO de RESPUESTA', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[0].set_xlabel('Canal de entrada', fontsize=15, fontdict=dict(weight='bold'))
ax[0].set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))
ax[0].tick_params(axis='x', which='major', labelsize=13, labelrotation=0)

ax[0].tick_params(axis='y', which='major', labelsize=15)
ax[0].yaxis.tick_left()  # Donde se ponen las marcas de eje

# CONSEJERIA
sns.barplot(ax=ax[1], x=dftr.groupby('consejeria')['tiempo_respuesta'].mean().
 ↪index,
            y=dftr.groupby('consejeria')['tiempo_respuesta'].mean().values,
            palette='winter',
            order=dftr.groupby('consejeria')['tiempo_respuesta'].mean().
 ↪sort_values(ascending=False).index)

ax[1].set_title('TIEMPO MEDIO de RESPUESTA', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax[1].set_xlabel('Consejería', fontsize=15, fontdict=dict(weight='bold'))
ax[1].set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))
ax[1].tick_params(axis='x', which='major', labelsize=16, labelrotation=90)
ax[1].set_xticks(ax[1].get_xticks(), ax[1].get_xticklabels(), ha='center')
ax[1].tick_params(axis='y', which='major', labelsize=15)
ax[1].yaxis.tick_left();  # Donde se ponen las marcas de eje

###

f, ax = plt.subplots(1, 1, figsize=(18,6))

# CONSEJERIA AGRUPADA
sns.barplot(ax=ax, x=dftr.groupby('nueva_conse')['tiempo_respuesta'].mean().
 ↪index,
            y=dftr.groupby('nueva_conse')['tiempo_respuesta'].mean().values,
            palette='winter',
```

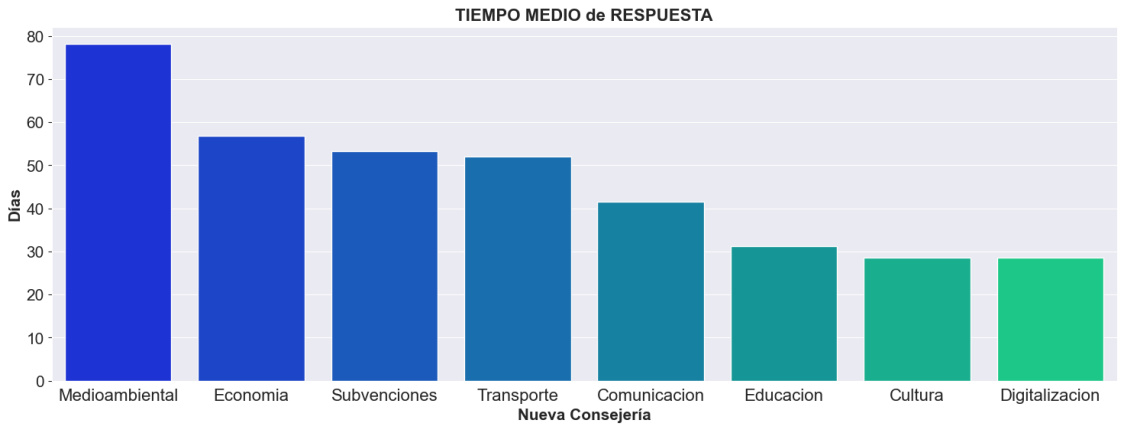
```
            order=dftr.groupby('nueva_conse')['tiempo_respuesta'].mean().
 ↪sort_values(ascending=False).index)

ax.set_title('TIEMPO MEDIO de RESPUESTA', fontsize=16, loc='center',␣
 ↪fontdict=dict(weight='bold'))
ax.set_xlabel('Nueva Consejería', fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))
ax.tick_params(axis='x', which='major', labelsize=16, labelrotation=0)
ax.set_xticks(ax.get_xticks(), ax.get_xticklabels(), ha='center')
ax.tick_params(axis='y', which='major', labelsize=15);
ax.yaxis.tick_left()  # Donde se ponen las marcas de eje
```






TIEMPO MEDIO de RESPUESTA
Días
50
40
30
20
10
0
LLAMADAS 012
CORREO ORDINARIO
PRESENCIAL
FAX
INTERNET
OFICINA MOVIL
OTROS
Teléfono 010
Canal de entrada


TIEMPO MEDIO de RESPUESTA
Días
80
60
40
20
0
Consejería de Medio Ambiente, Vivienda y Agricultura
Consejería de Sanidad
Consejería de Transportes e Infraestructuras
Consejería de Transportes, Vivienda e Infraestructuras
Consejería de Presidencia, Justicia y Portavocía del Gobierno
Consejería de Economía, Hacienda y Empleo
Consejería de Medio Ambiente, Ordenación del Territorio y Sostenibilidad
Consejería de Familia, Juventud y Política Social
Consejería de Políticas Sociales, Familias, Igualdad y Natalidad
Consorcio Regional de Transportes
Vicepresidencia, Consejería de Deportes, Transparencia y Portavocía de Gobierno
Consejería de Educación e Investigación
Consejería de Presidencia, Justicia e Interior
Consejería de Educación, Juventud y Deporte
Oficina de Cultura y Turismo
Agencia para la Administración Digital de la Comunidad de Madrid
Consejería de Cultura, Turismo y Deporte
Consejería de Educación, Universidades, Ciencia y Portavocía
Consejería



[227]:
```
dftr.groupby('nueva_conse')['tiempo_respuesta'].mean().
↪sort_values(ascending=False)
```

[227]:
```
nueva_conse
Medioambiental    78.234568
Economia          56.888889
Subvenciones      53.263930
Transporte        52.032209
Comunicacion      41.605714
Educacion         31.163749
Cultura           28.531353
Digitalizacion    28.513661
Name: tiempo_respuesta, dtype: float64
```

[228]:
```
dftr.groupby('consejeria')['tiempo_respuesta'].mean().
↪sort_values(ascending=False)
```

[228]:
```
consejeria
Consejería de Medio Ambiente, Vivienda y Agricultura
86.114754
Consejería de Sanidad
80.444444
Consejería de Transportes e Infraestructuras
75.550607
Consejería de Transportes, Vivienda e Infraestructuras
63.004593
Consejería de Presidencia, Justicia y Portavocía del Gobierno
62.350000
Consejería de Economía, Hacienda y Empleo
56.888889
Consejería de Medio Ambiente, Ordenación del Territorio y Sostenibilidad
54.200000
```

```
Consejería de Familia, Juventud y Política Social
53.583555
Consejería de Políticas Sociales, Familias, Igualdad y Natalidad
53.066209
Consorcio Regional de Transportes
51.056590
Vicepresidencia, Consejería de Deportes, Transparencia y Portavocía de Gobierno
46.000000
Consejería de Educación e Investigación
42.741269
Consejería de Presidencia, Justicia e Interior
34.706349
Consejería de Educación, Juventud y Deporte
32.571515
Oficina de Cultura y Turismo
30.082227
Agencia para la Administración Digital de la Comunidad de Madrid
28.513661
Consejería de Cultura, Turismo y Deporte
26.965787
Consejería de Educación, Universidades, Ciencia y Portavocía
25.472536
Name: tiempo_respuesta, dtype: float64
```

[229]:
```
dftr.groupby('canal_entrada')['tiempo_respuesta'].mean().
  ↪sort_values(ascending=False)
```

[229]:
```
canal_entrada
LLAMADAS 012        48.692586
CORREO ORDINARIO    47.830952
PRESENCIAL          47.768833
FAX                 43.010309
INTERNET            41.910141
OFICINA MOVIL       39.750000
OTROS               28.688172
Teléfono 010        26.000000
Name: tiempo_respuesta, dtype: float64
```

[230]:
```
dftr.groupby('tema')['tiempo_respuesta'].mean().sort_values(ascending=False)
```

[230]:
```
tema
POLÍTICAS SOCIALES    53.352582
TRANSPORTES           51.992049
ENERGÍA               34.021739
EDUCACIÓN             31.352126
CULTURA               28.530126
Name: tiempo_respuesta, dtype: float64
```

[231]:
```
dftr.groupby('tipo')['tiempo_respuesta'].mean().sort_values(ascending=False)
```

[231]:
```
tipo
Queja             45.432996
Sugerencia        40.653613
Agradecimiento    28.695652
Name: tiempo_respuesta, dtype: float64
```

5.4 Variables categóricas - Temporalidad

5.4.1 Tipo

Mensual

- **Queja**. El tiempo medio de respuesta es ligeramente decreciente a lo largo del tiempo, aunque las entradas se mantienen estables.
- El tiempo de respuesta se eleva notablemente en la segunda mitad de 2019 y hasta el primer tercio de 2020.
- **Sugerencia**. Comportamiento similar a Queja, pero con un volumen de entradas menor.
- **Agradecimiento**. El número de datos es muy bajo (solo 13 datos -meses-). El volumen de entradas es muy bajo (menor a 10). Esto puede explicar la diferencia en el tiempo de respuesta medio entre Agradecimiento y Queja o Sugerencia. Solo hay Agradecimiento a partir de 2022.

[232]:
```
dftr_tipo = dftr.reset_index().groupby(['fecha_entrada', 'tipo']).
↪agg(tr_medio=('tiempo_respuesta', 'mean'),
                                                                    ␣
↪total=('tipo', 'count'))
```

[233]:
```
dftr_tipo = dftr_tipo.reset_index().set_index('fecha_entrada').groupby('tipo').
↪resample('M') \
                                          .agg(tr_medio=('tr_medio', 'mean'),␣
↪total=('total', 'sum'))
```

[234]:
```
dftr_tipo = dftr_tipo.reset_index().set_index('fecha_entrada')
```

[235]:
```
dftr_tipo
```

[235]:

fecha_entrada	tipo	tr_medio	total
2021-12-31	Agradecimiento	192.000000	1
2022-01-31	Agradecimiento	NaN	0
2022-02-28	Agradecimiento	NaN	0
…	…	…	…
2022-10-31	Sugerencia	38.883333	51
2022-11-30	Sugerencia	20.507246	48
2022-12-31	Sugerencia	22.416667	36

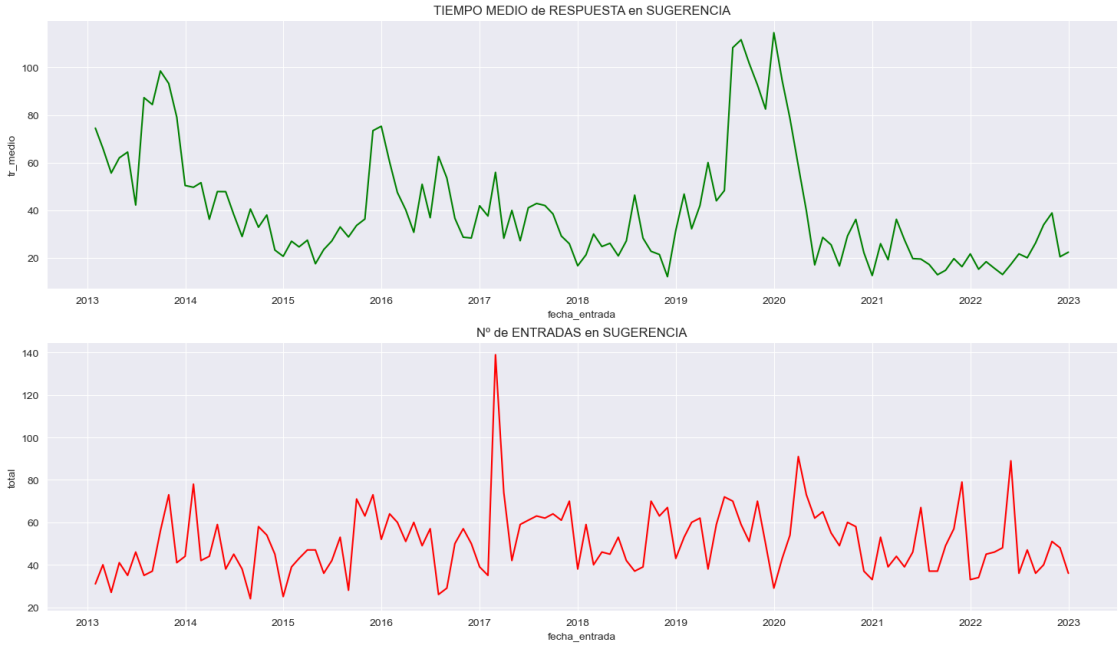
```
[253 rows x 3 columns]
```

[236]:
```
# Queja
dftr_q_mes = dftr_tipo[dftr_tipo.tipo == 'Queja']

# Sugerencia
dftr_s_mes = dftr_tipo[dftr_tipo.tipo == 'Sugerencia']

# Agradecimiento
dftr_a_mes = dftr_tipo[dftr_tipo.tipo == 'Agradecimiento']
```

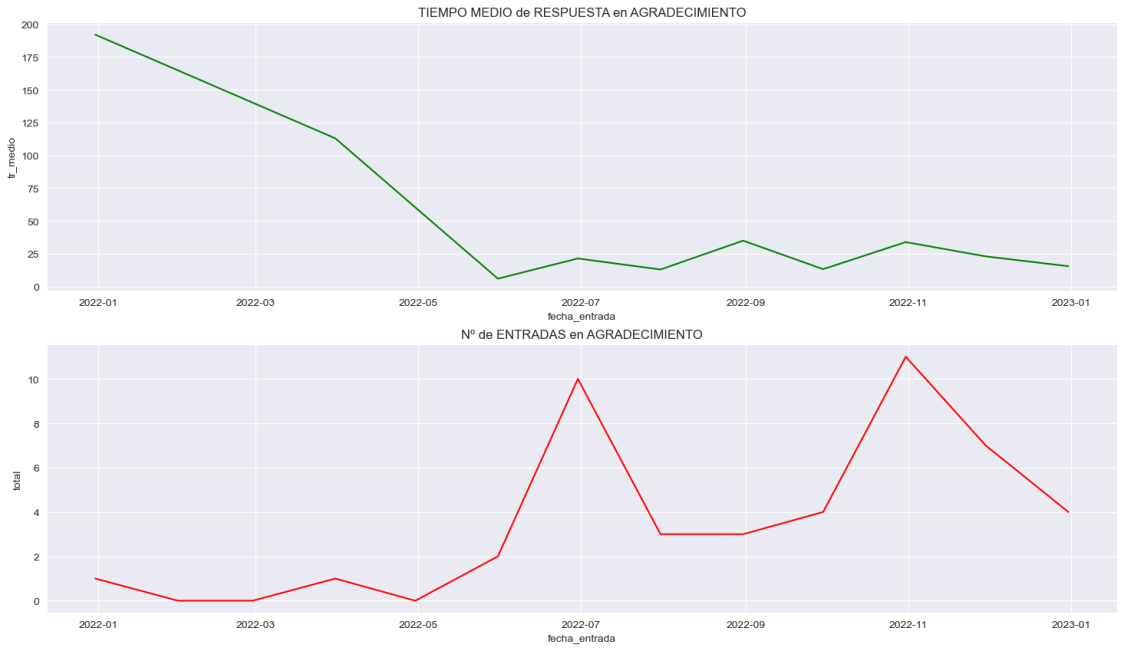
[237]:
```
f, ax = plt.subplots(2,1, figsize=(18,10))

sns.lineplot(ax=ax[0], data=dftr_q_mes, x=dftr_q_mes.index, y=dftr_q_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en QUEJA',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=dftr_q_mes, x=dftr_q_mes.index, y=dftr_q_mes.total,␣
 ↪color='red') \
                    .set_title(label='Nº de ENTRADAS en QUEJA', loc='center');
```

[238]:
```
f, ax = plt.subplots(2,1, figsize=(18,10))
```

```
sns.lineplot(ax=ax[0], data=dftr_s_mes, x=dftr_s_mes.index, y=dftr_s_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en SUGERENCIA',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=dftr_s_mes, x=dftr_s_mes.index, y=dftr_s_mes.total,␣
 ↪color='red') \
                    .set_title(label='Nº de ENTRADAS en SUGERENCIA',␣
 ↪loc='center');
```



[239]:
```
f, ax = plt.subplots(2,1, figsize=(18,10))

sns.lineplot(ax=ax[0], data=dftr_a_mes, x=dftr_a_mes.index, y=dftr_a_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en AGRADECIMIENTO',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=dftr_a_mes, x=dftr_a_mes.index, y=dftr_a_mes.total,␣
 ↪color='red') \
                    .set_title(label='Nº de ENTRADAS en AGRADECIMIENTO',␣
 ↪loc='center');
```



Estacionalidad Año

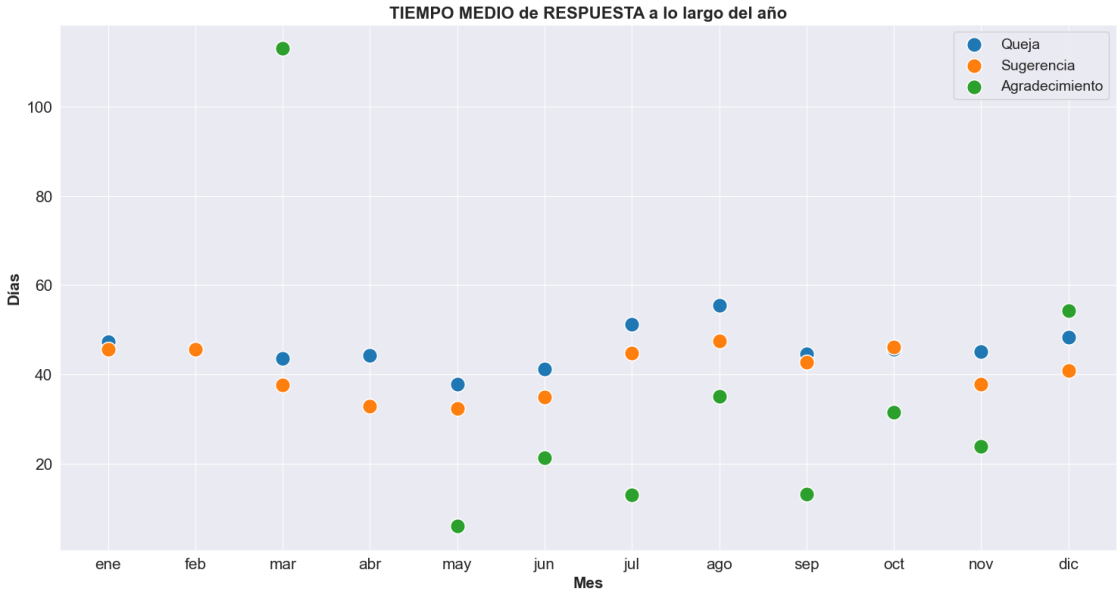
- **Queja**. El tiempo de respuesta es algo mayor a Sugerencia y Agradecimiento durante casi todo el año, excepto en febrero, octurbre y diciembre.
- **Sugerencia**. Oscila igual que Queja, casi siempre con valores inferiores a esta en el tiempo de respuesta.
- **Agradecimiento**. Destaca el pico máximo en marzo. También tiene el mayor tiempo de respuesta en diciembre. El resto del año está en números inferiores a los de que Queja y Sugerencia.
- Hay un gran diferencia en el número de entradas de Queja, por un lado, y Sugerenca y Agradecimiento por otro.
- El **máximo se produce en los tres casos en octubre**.
- Queja oscila a lo largo del año, con el mínimo claramente en agosto, y luego diciembre.
- Sugerencia también oscila, pero en un orden menor que Queja. Su mínimo está en diciembre y luego en agosto.
- Agradecimiento obtiene el mínimo en marzo.

[240]:
```
dftr_tipo_a = dftr.groupby(['mes', 'tipo']).agg(
    tr_medio=('tiempo_respuesta', 'mean'),
    total=('tipo', 'count')).reset_index()
```

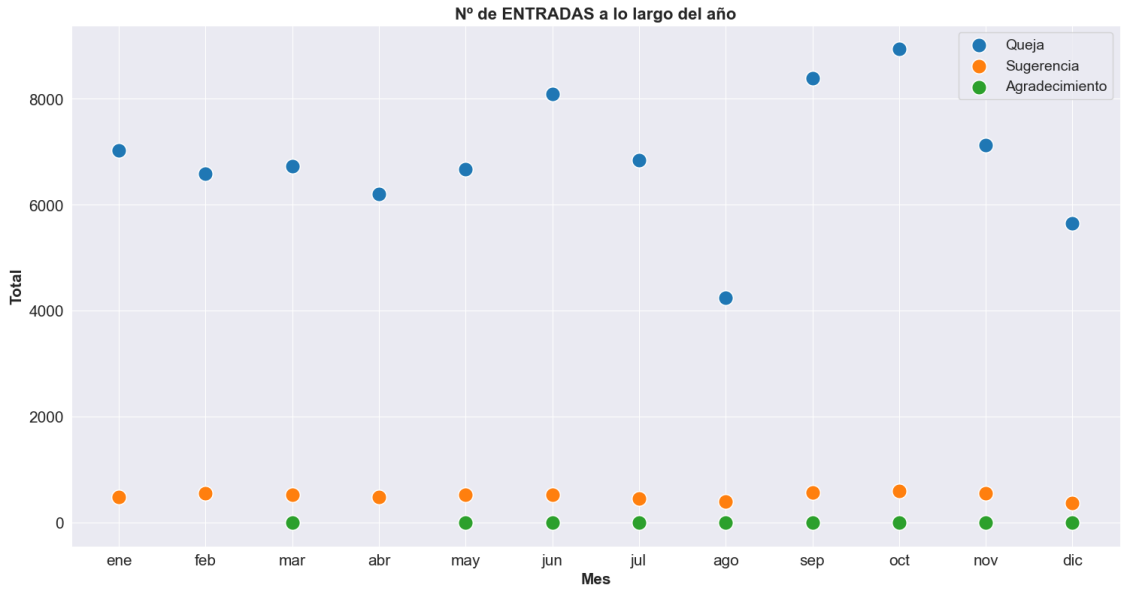
[241]:
```
dftr_tipo_a = dftr_tipo_a.set_index('mes')
```

[242]:
```
# Queja
dftr_q_a = dftr_tipo_a[dftr_tipo_a.tipo == 'Queja']

# Sugerencia
dftr_s_a = dftr_tipo_a[dftr_tipo_a.tipo == 'Sugerencia']

# Agradecimiento
dftr_a_a = dftr_tipo_a[dftr_tipo_a.tipo == 'Agradecimiento']
```

[243]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_q_a.index, y=dftr_q_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_s_a.index, y=dftr_s_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_a_a.index, y=dftr_a_a.tr_medio, s=200)

ax.set_title('TIEMPO MEDIO de RESPUESTA a lo largo del año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_a.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Queja', 'Sugerencia', 'Agradecimiento'],␣
 ↪fontsize=14);
```



[244]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_q_a.index, y=dftr_q_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_s_a.index, y=dftr_s_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_a_a.index, y=dftr_a_a.total, s=200)

ax.set_title('Nº de ENTRADAS a lo largo del año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Total', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_a.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Queja', 'Sugerencia', 'Agradecimiento'],␣
 ↪fontsize=14);
```



Estacionalidad Día Semana

- El mayor número de días en responder ocurre los días miércoles y jueves.
- Hay desacoplamiento entre Agradecimiento y el resto, excepto para el día martes.
- Las quejas siempre tardan más en responderse, excepto el día miércoles.
- Queja y Sugerencia tienen un comportamiento similar, disminuyen ligeramente hasta el viernes.
- Agradecimiento. El número de entradas disminuye del lunes al miércoles (mínimo) y vuelve a subir hasta el viernes.

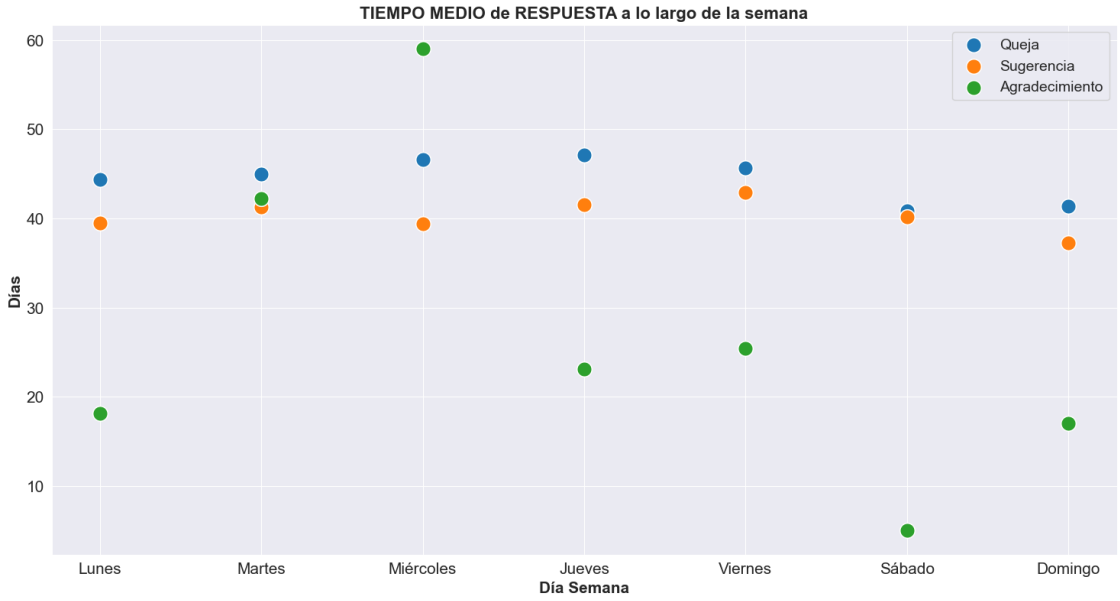
[245]:
```
dftr_tipo_ds = dftr.groupby(['dia_semana', 'tipo']).agg(
    tr_medio=('tiempo_respuesta', 'mean'),
    total=('tipo', 'count')).reset_index()
```

[246]:
```
dftr_tipo_ds = dftr_tipo_ds.set_index('dia_semana')
```

[247]:
```
# Queja
dftr_q_ds = dftr_tipo_ds[dftr_tipo_ds.tipo == 'Queja']

# Sugerencia
dftr_s_ds = dftr_tipo_ds[dftr_tipo_ds.tipo == 'Sugerencia']

# Agradecimiento
dftr_a_ds = dftr_tipo_ds[dftr_tipo_ds.tipo == 'Agradecimiento']
```

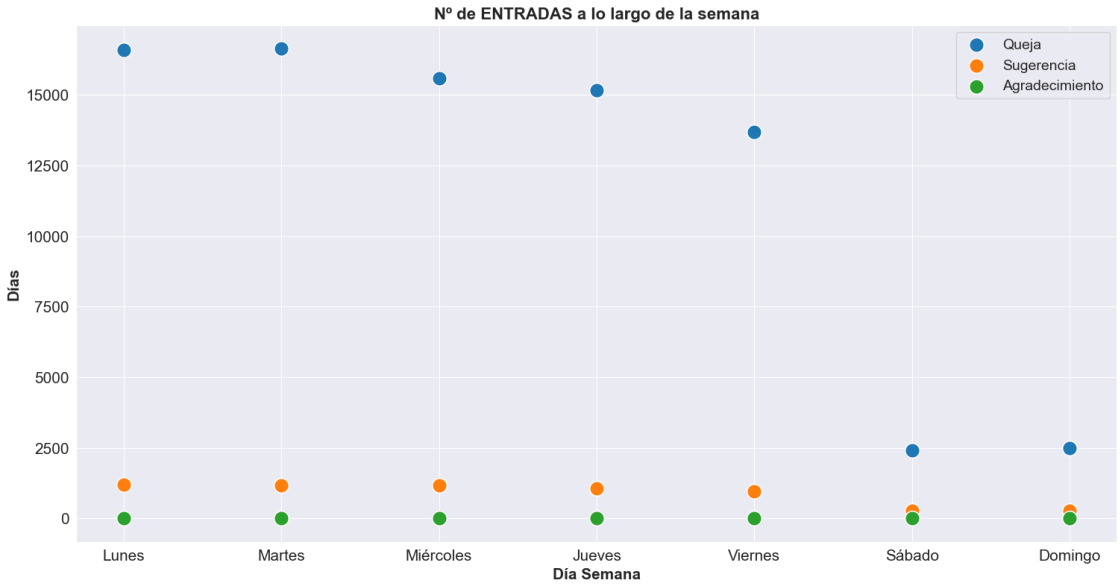
[248]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))
```

```
sns.scatterplot(ax=ax, x=dftr_q_ds.index, y=dftr_q_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_s_ds.index, y=dftr_s_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_a_ds.index, y=dftr_a_ds.tr_medio, s=200)

ax.set_title('TIEMPO MEDIO de RESPUESTA a lo largo de la semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día Semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Queja', 'Sugerencia', 'Agradecimiento'],␣
 ↪fontsize=14);
```



[249]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_q_ds.index, y=dftr_q_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_s_ds.index, y=dftr_s_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_a_ds.index, y=dftr_a_ds.total, s=200)

ax.set_title('Nº de ENTRADAS a lo largo de la semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))
```

```
ax.set_xlabel("Día Semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_ds.index, labels=etiquetas)  # Solo para poner las␣
↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Queja', 'Sugerencia', 'Agradecimiento'],␣
↪fontsize=14);
```



5.4.2 Tema

Mensual

[250]:
```
dftr_tema = dftr.reset_index().groupby(['fecha_entrada', 'tema']).
↪agg(tr_medio=('tiempo_respuesta', 'mean'),
                                                                    ␣
↪total=('tema', 'count'))
```

[251]:
```
dftr_tema = dftr_tema.reset_index().set_index('fecha_entrada').groupby('tema').
↪resample('M') \
                                        .agg(tr_medio=('tr_medio', 'mean'),␣
↪total=('total', 'sum'))
```

[252]:
```
dftr_tema = dftr_tema.reset_index().set_index('fecha_entrada')
```

[253]:
```
dftr_tema
```

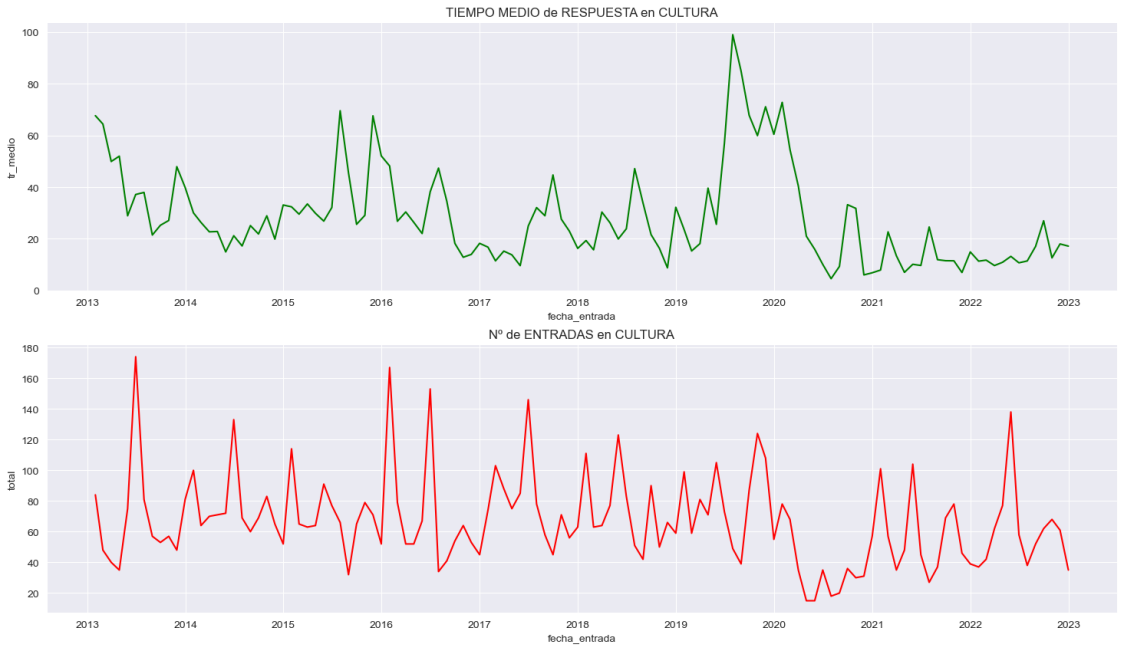
[253]:

fecha_entrada	tema	tr_medio	total
2013-01-31	CULTURA	67.634524	84
2013-02-28	CULTURA	64.396569	48
2013-03-31	CULTURA	49.906863	40
…	…	…	…
2022-10-31	TRANSPORTES	19.432999	127
2022-11-30	TRANSPORTES	19.102564	98
2022-12-31	TRANSPORTES	16.849667	126

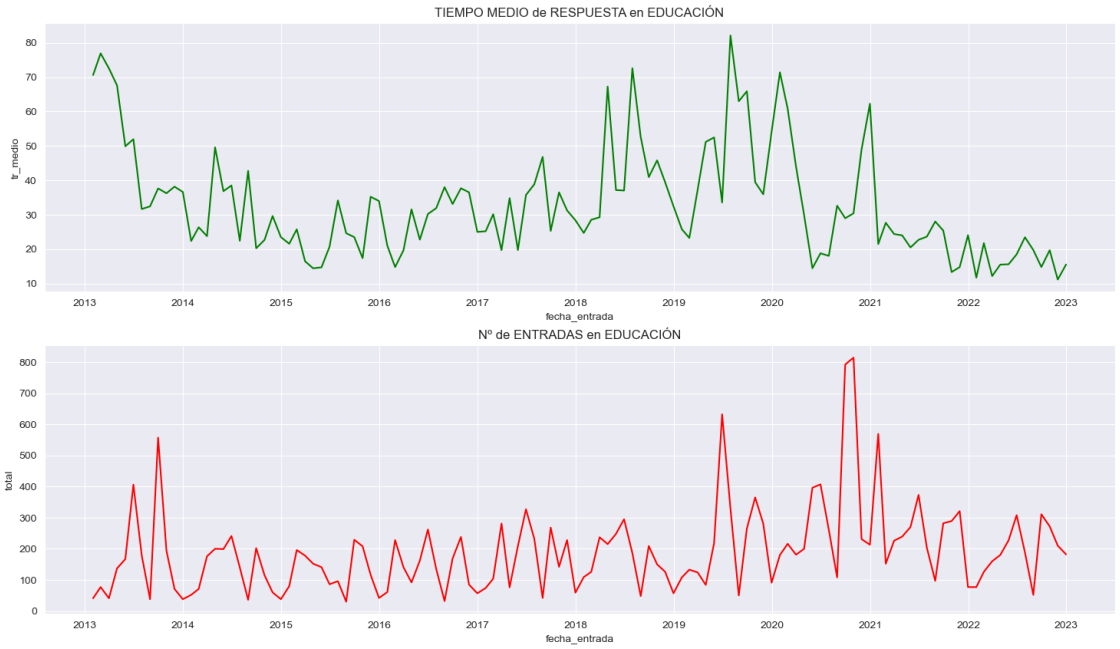
[492 rows x 3 columns]

[254]:
```
# Cultura
dftr_cu_mes = dftr_tema[dftr_tema.tema == 'CULTURA']

# Educación
dftr_ed_mes = dftr_tema[dftr_tema.tema == 'EDUCACIÓN']

# Energía
dftr_en_mes = dftr_tema[dftr_tema.tema == 'ENERGÍA']

# Políticas Sociales
dftr_ps_mes = dftr_tema[dftr_tema.tema == 'POLÍTICAS SOCIALES']

# Transportes
dftr_tr_mes = dftr_tema[dftr_tema.tema == 'TRANSPORTES']
```

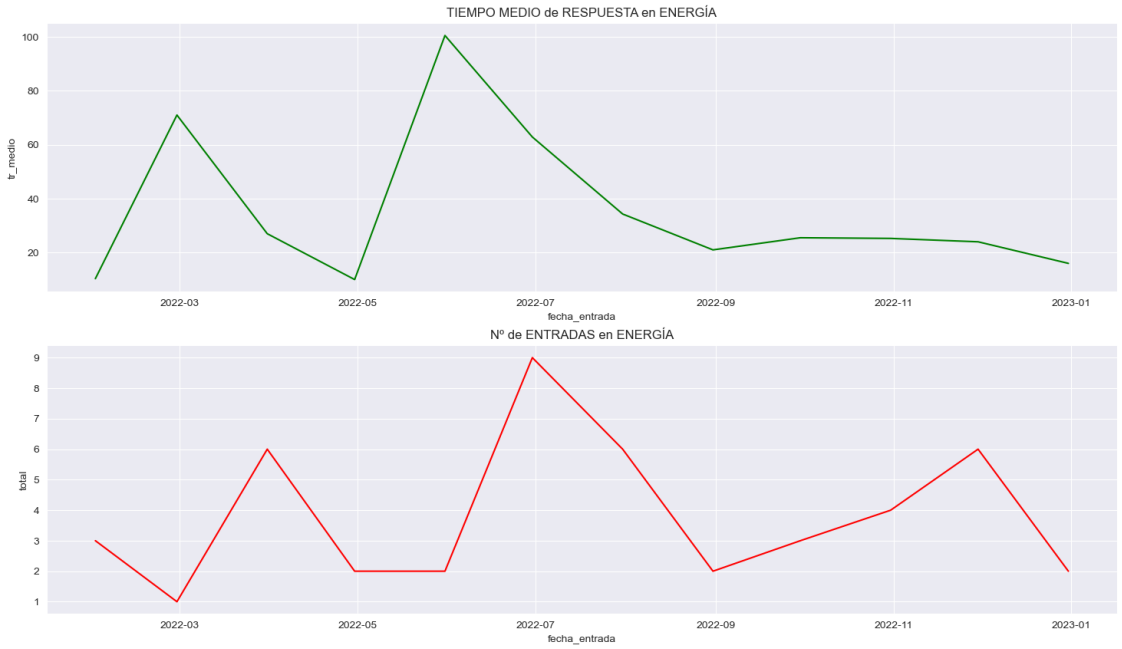
[255]:
```
f, ax = plt.subplots(2,1, figsize=(18,10))

sns.lineplot(ax=ax[0], data=dftr_cu_mes, x=dftr_cu_mes.index, y=dftr_cu_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en CULTURA',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=dftr_cu_mes, x=dftr_cu_mes.index, y=dftr_cu_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en CULTURA', loc='center');
```



[256]:
```
f, ax = plt.subplots(2,1, figsize=(18,10))

sns.lineplot(ax=ax[0], data=dftr_ed_mes, x=dftr_ed_mes.index, y=dftr_ed_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en EDUCACIÓN',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=dftr_ed_mes, x=dftr_ed_mes.index, y=dftr_ed_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en EDUCACIÓN',␣
 ↪loc='center');
```



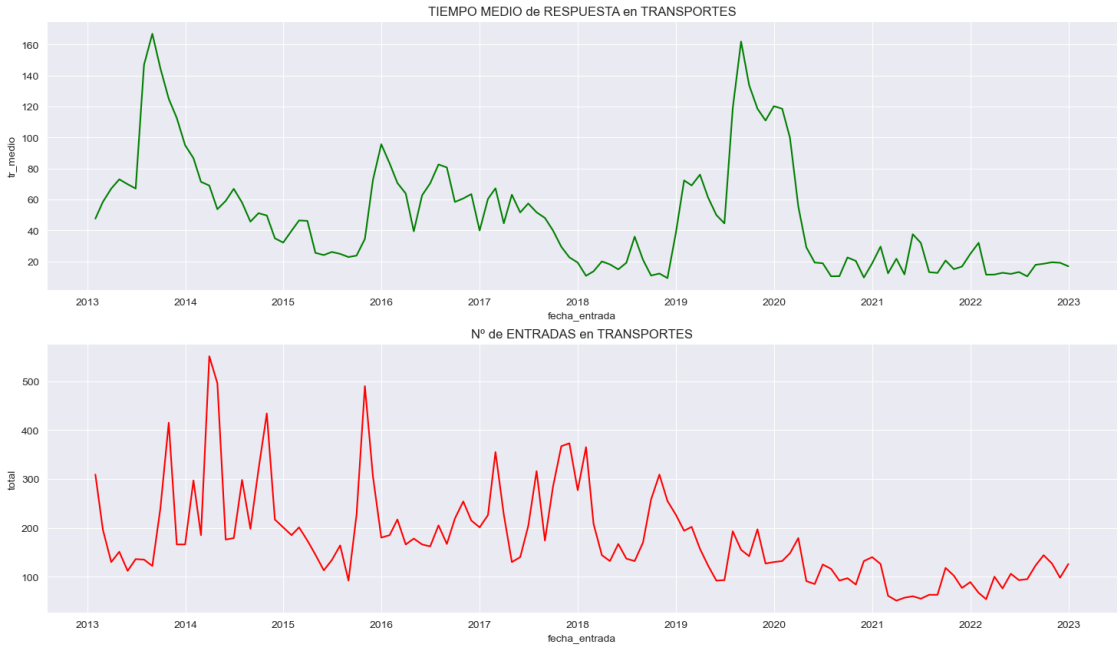
[257]:
```
f, ax = plt.subplots(2,1, figsize=(18,10))

sns.lineplot(ax=ax[0], data=dftr_en_mes, x=dftr_en_mes.index, y=dftr_en_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en ENERGÍA',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=dftr_en_mes, x=dftr_en_mes.index, y=dftr_en_mes.
 ↪total, color='red') \
                  .set_title(label='Nº de ENTRADAS en ENERGÍA', loc='center');
```



[258]:
```
f, ax = plt.subplots(2,1, figsize=(18,10))

sns.lineplot(ax=ax[0], data=dftr_ps_mes, x=dftr_ps_mes.index, y=dftr_ps_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en POLÍTICAS␣
 ↪SOCIALES', loc='center')

sns.lineplot(ax=ax[1], data=dftr_ps_mes, x=dftr_ps_mes.index, y=dftr_ps_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en POLÍTICAS SOCIALES',␣
 ↪loc='center');
```

- **Cultura**. El tiempo de respuesta disminuye a lo largo del tiempo. El número de entradas desciende a lo largo del tiempo. El tiempo de respuesta se dispara desde la segunda mitad de 2019 hasta inicios de 2020.
- **Educación**. El tiempo medio de respuesta disminuye hasta 2016, a partir de ahí aumenta hasta mediados de 2019 y luego vuelve a caer a sus niveles más bajos (15 días). El número de entradas es ligeramente creciente.
- **Energía**. Pocos datos. La duración media es igual de alta que en otros tema (100 días), a pesar de que el número de entradas es muy inferior (5 vs 60, aproximadamente). Los datos empiezan en 2022. **No se entiende esta desproporción entre carga de trabajo y demora en la respuesta**.
- **Políticas Sociales**. El número de entradas crece, pero el tiempo de respuesta medio no se altera (40-50). Es el que tiene más entradas solo después de educación. En torno a la segunda mitad de 2019 sufre igualmente un notable aumento en el tiempo de respuesta (140-160) alcanzando su máximo.
- **Transportes**. Hay igualmente un aumento del tiempo de respuesta en la segunda mitad de 2019, pero sin alcanzar el máximo que llegó en la segunda mitad de 2013. Desde mitad del 2020 el tiempo medio se ha reducido y permanece estable alrededor de 15-20 días, la segunda más baja tras educación. El número de entradas se ha desplomado desde 2019, pasando de unas 300 entradas a poco más de 100.
- La **parálisis de la segunda mitad de 2019** se deba, tal vez, a las **elecciones** generales (abril y noviembre) y autonómicas (mayo) de ese año.

[259]:
```
f, ax = plt.subplots(2,1, figsize=(18,10))

sns.lineplot(ax=ax[0], data=dftr_tr_mes, x=dftr_tr_mes.index, y=dftr_tr_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en TRANSPORTES',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=dftr_tr_mes, x=dftr_tr_mes.index, y=dftr_tr_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en TRANSPORTES',␣
 ↪loc='center');
```

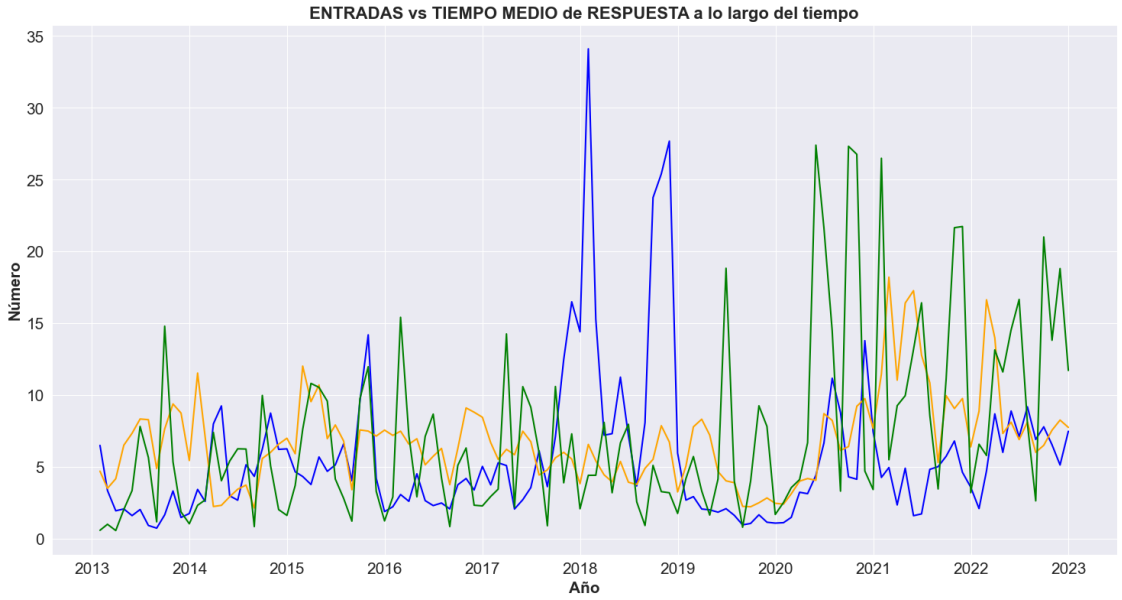


Comparación ratio ENTRADAS / TIEMPO_MEDIO_RESPUESTA

[260]:
```
dftr_tr_mes = dftr_tr_mes.copy()  # Copia para evitar warning.
dftr_tr_mes.loc[:, 'tot_tr_medio'] = dftr_tr_mes.total / dftr_tr_mes.tr_medio

dftr_ps_mes = dftr_ps_mes.copy()  # Copia para evitar warning.
dftr_ps_mes.loc[:, 'tot_tr_medio'] = dftr_ps_mes.total / dftr_ps_mes.tr_medio

dftr_ed_mes = dftr_ed_mes.copy()  # Copia para evitar warning.
dftr_ed_mes.loc[:, 'tot_tr_medio'] = dftr_ed_mes.total / dftr_ed_mes.tr_medio
```

[261]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))
```

```
sns.lineplot(ax=ax, x=dftr_tr_mes.index, y=dftr_tr_mes.tot_tr_medio,␣
 ↪color='blue')
sns.lineplot(ax=ax, x=dftr_ps_mes.index, y=dftr_ps_mes.tot_tr_medio,␣
 ↪color='orange')
sns.lineplot(ax=ax, x=dftr_ed_mes.index, y=dftr_ed_mes.tot_tr_medio,␣
 ↪color='green')

ax.set_title('ENTRADAS vs TIEMPO MEDIO de RESPUESTA a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15);
```



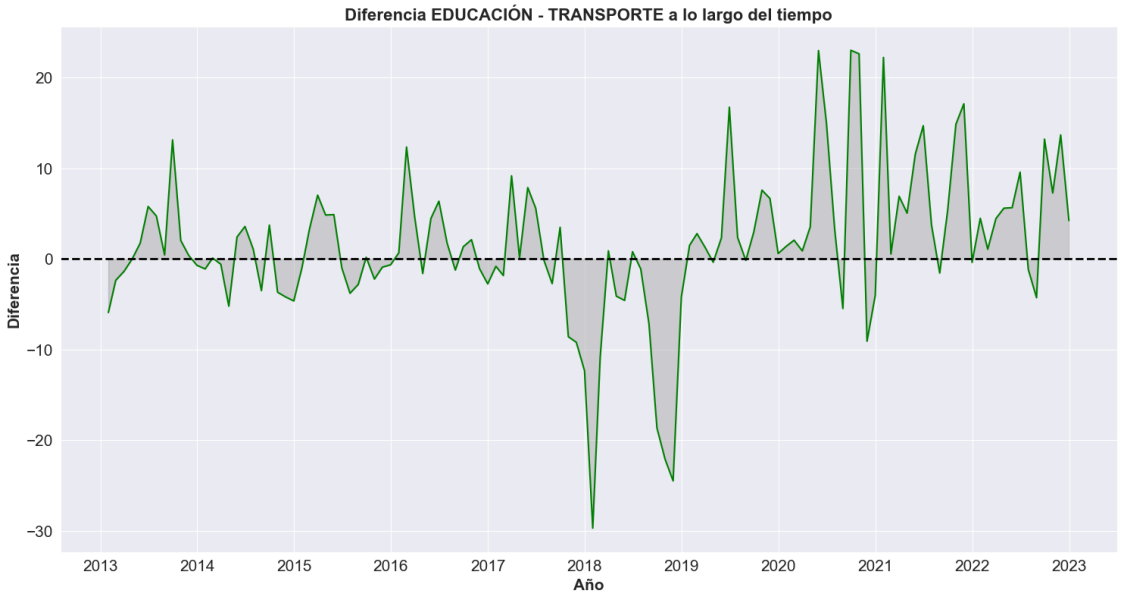
Diferencia de ratios y Diferencia acumulada

- Durante 2018 es claramente mejor Transportes, a partir de ese año Educación es mejor.
- Igual que con Educación, Transportes es mejor que Políticas Sociales durante 2018, pero la superioridad de Políticas Sociales sobre Transporte es casi absoluta durante el resto del tiempo.
- La diferencia acumulada muestra la predominancia de Educación sobre Transportes, y aún más de Políticas Sociales sobre Transportes.
- La comparación entre Políticas Sociales y Educación, como contraste, muestra que hasta 2019 Políticas Sociales fue acumulando ventaja sobre Educación, pero a partir de ese año la mejora

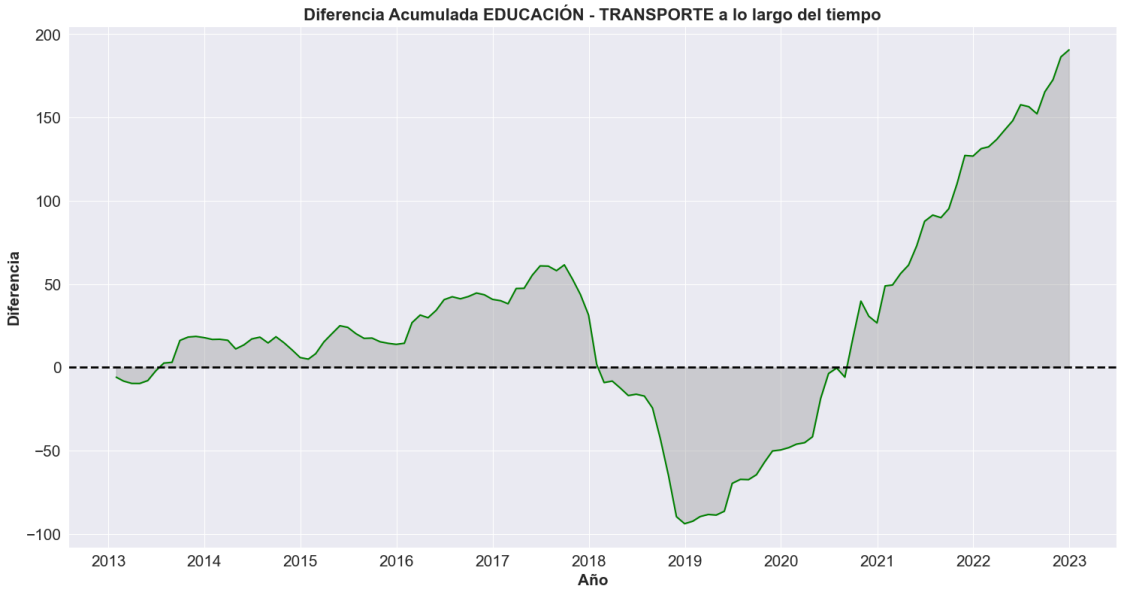
de Educación ha sido rápida y a finales de 2022 se muestra claramente más eficiente.

[262]:
```
# Diferencia Educación vs Transportes
edtr = dftr_ed_mes.tot_tr_medio - dftr_tr_mes.tot_tr_medio
edtr = edtr.to_frame()

# Acumulación de diferencia
edtr['dif_acumulada'] = edtr.tot_tr_medio.cumsum()
```

[263]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=edtr.index, y=edtr.tot_tr_medio, color='green')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia EDUCACIÓN - TRANSPORTE a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```

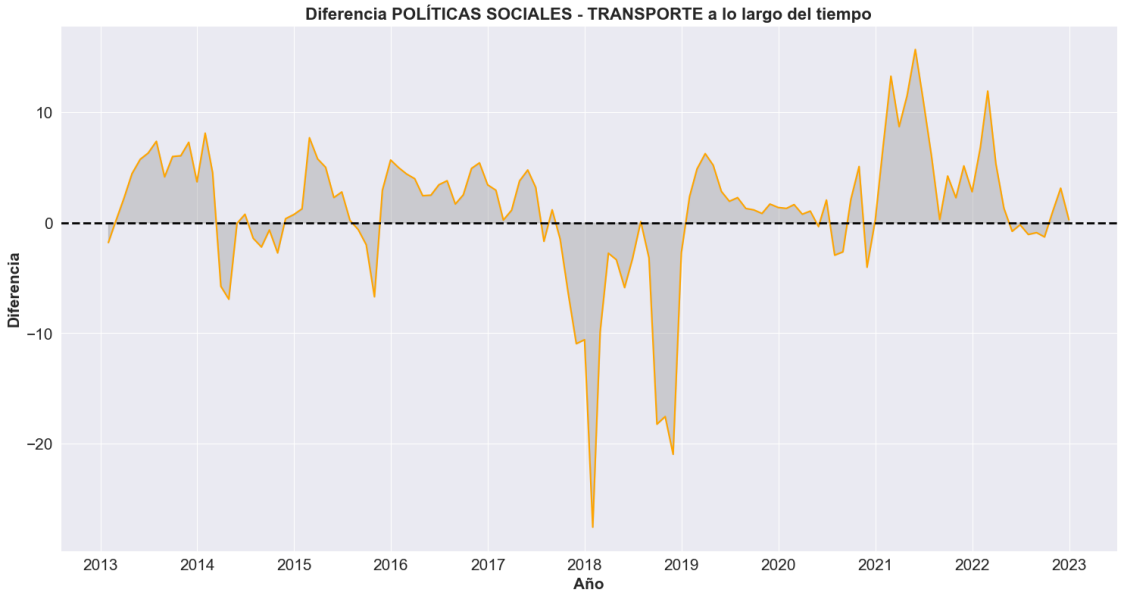


[264]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=edtr.index, y=edtr.dif_acumulada, color='green')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada EDUCACIÓN - TRANSPORTE a lo largo del␣
 ↪tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```

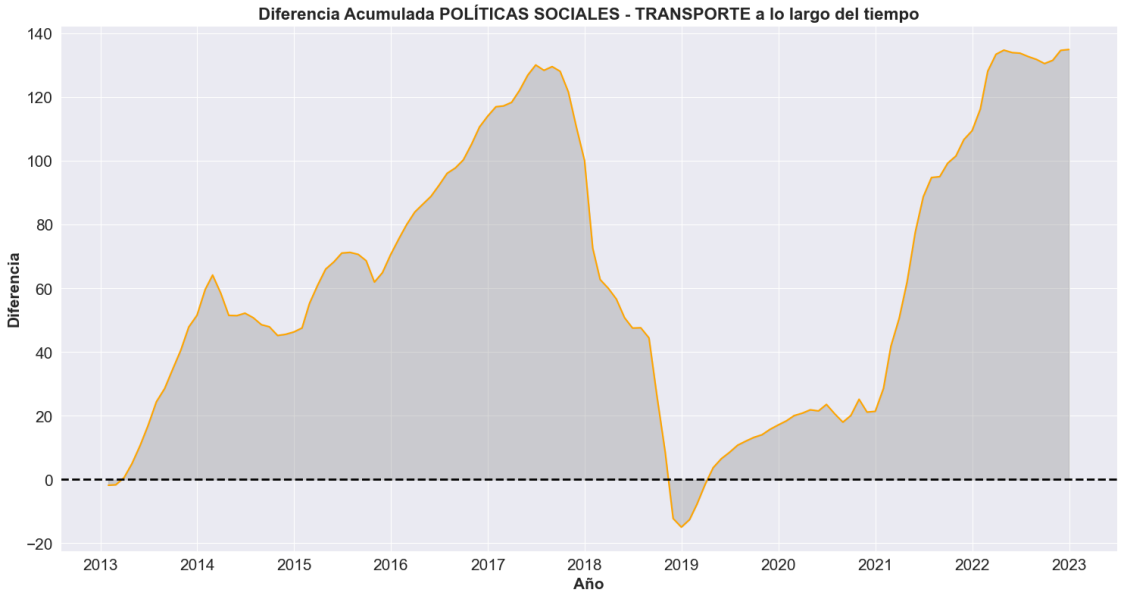


[265]:
```
# Diferencia Políticas Sociales vs Transportes
pstr = dftr_ps_mes.tot_tr_medio - dftr_tr_mes.tot_tr_medio
pstr = pstr.to_frame()

# Acumulación de diferencia
pstr['dif_acumulada'] = pstr.tot_tr_medio.cumsum()
```

[266]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=pstr.index, y=pstr.tot_tr_medio, color='orange')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia POLÍTICAS SOCIALES - TRANSPORTE a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



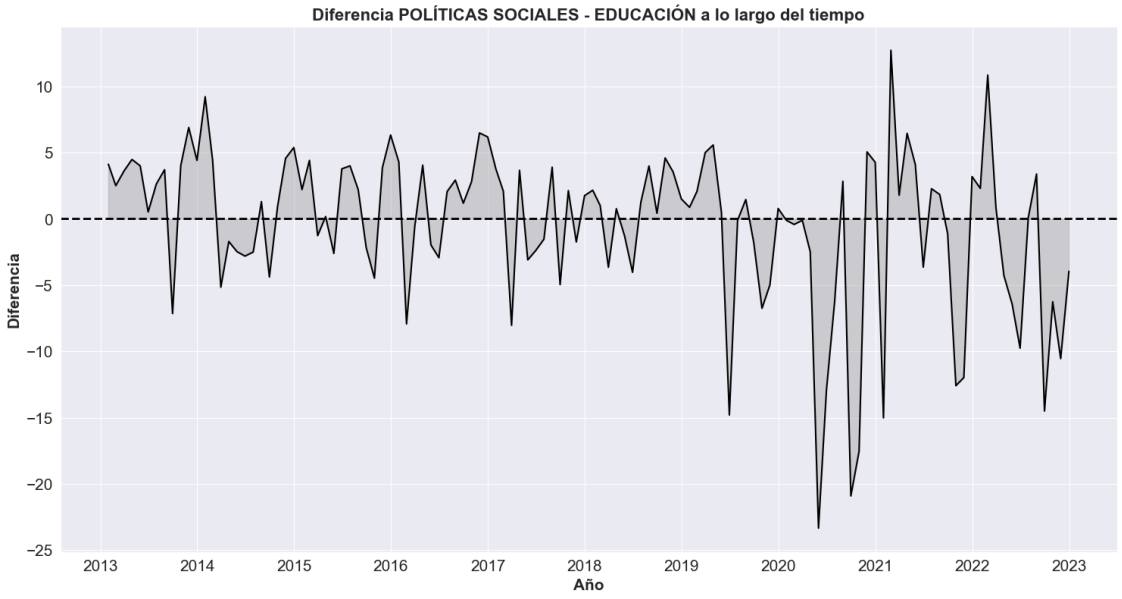
[267]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=pstr.index, y=pstr.dif_acumulada, color='orange')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada POLÍTICAS SOCIALES - TRANSPORTE a lo largo␣
 ↪del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



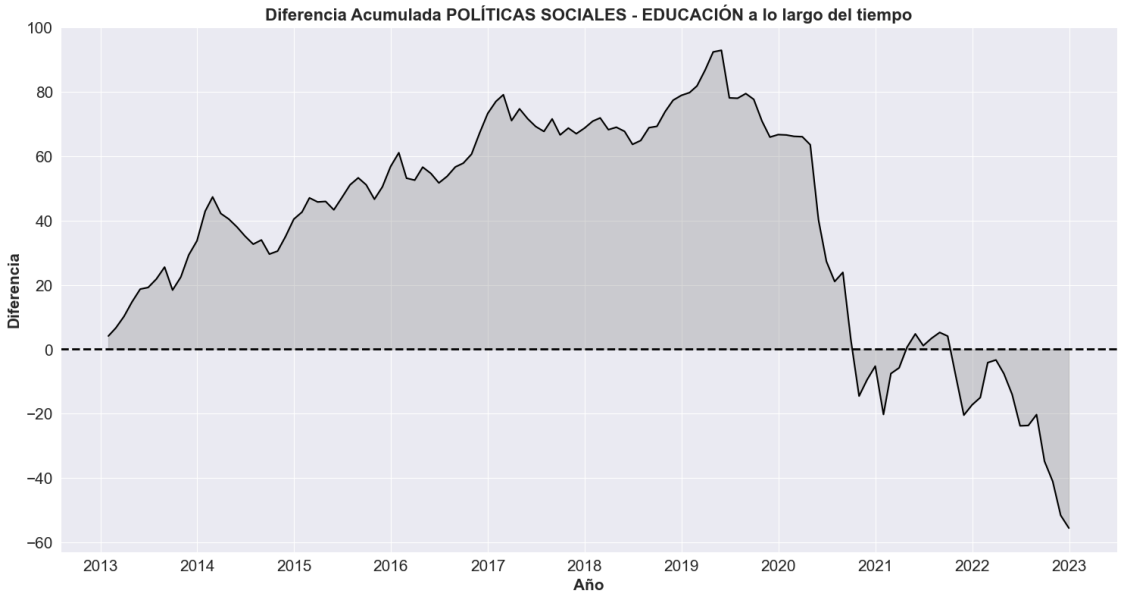
[268]:
```
# Diferencia Políticas Sociales vs Educación
psed = dftr_ps_mes.tot_tr_medio - dftr_ed_mes.tot_tr_medio
```

```
psed = psed.to_frame()

# Acumulación de diferencia
psed['dif_acumulada'] = psed.tot_tr_medio.cumsum()
```

[269]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=psed.index, y=psed.tot_tr_medio, color='black')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia POLÍTICAS SOCIALES - EDUCACIÓN a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



[270]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=psed.index, y=psed.dif_acumulada, color='black')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada POLÍTICAS SOCIALES - EDUCACIÓN a lo largo␣
 ↪del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



Estacionalidad Año

- **Energía**. Tiene el máximo tiempo de respuesta en mayo aunque el número de entradas es el más bajo para todos los meses.

- **Cultura**. Es el segundo con menos entradas en todos los meses, pero en enero, julio y noviembre es el tercer tiempo de respuesta más alto.
- **Educación**. Tiene un número de entradas generalmente mayor a **Transportes**, pero tiene un tiempo de respuesta menor, lo que muestra que es más eficiene comparativamente.
- **Políticas Sociales**. Tiene un tiempo de respuesta similar a Transportes, pero con un número de entradas que es el mayor de todos.
- **Transportes**. Se muestra más ineficiente en comparación con Políticas Sociales y Educación.

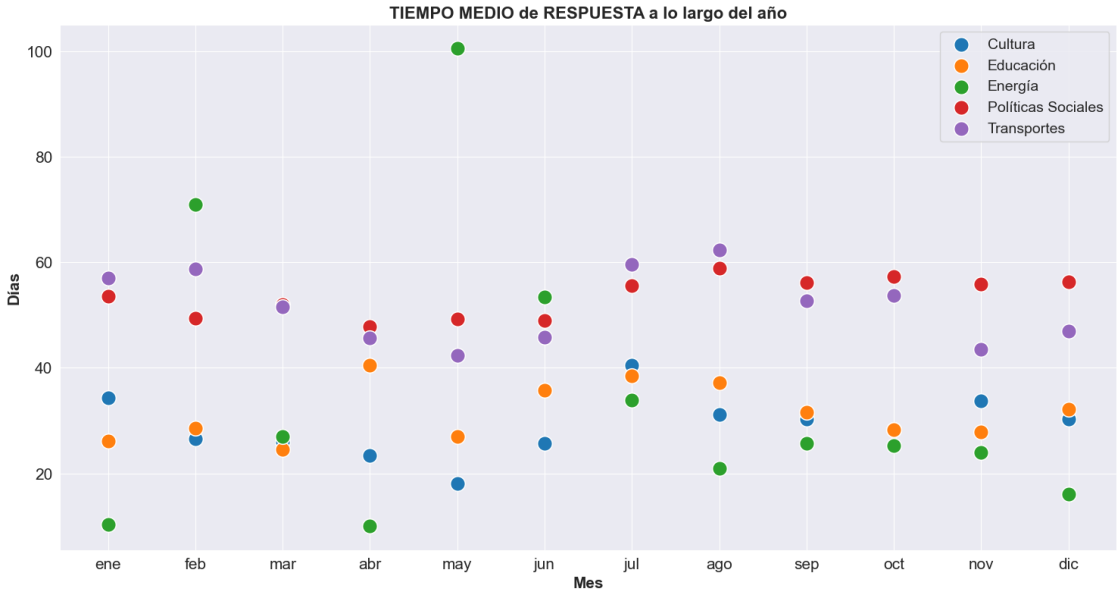
[271]:
```
dftr_tema_a = dftr.groupby(['mes', 'tema']).agg(
    tr_medio=('tiempo_respuesta', 'mean'),
    total=('tema', 'count')).reset_index()
```

[272]:
```
dftr_tema_a = dftr_tema_a.set_index('mes')
```

[273]:
```
# Cultura
dftr_cu_a = dftr_tema_a[dftr_tema_a.tema == 'CULTURA']

# Educación
dftr_ed_a = dftr_tema_a[dftr_tema_a.tema == 'EDUCACIÓN']

# Energía
dftr_en_a = dftr_tema_a[dftr_tema_a.tema == 'ENERGÍA']

# Políticas Sociales
dftr_ps_a = dftr_tema_a[dftr_tema_a.tema == 'POLÍTICAS SOCIALES']

# Transportes
dftr_tr_a = dftr_tema_a[dftr_tema_a.tema == 'TRANSPORTES']
```

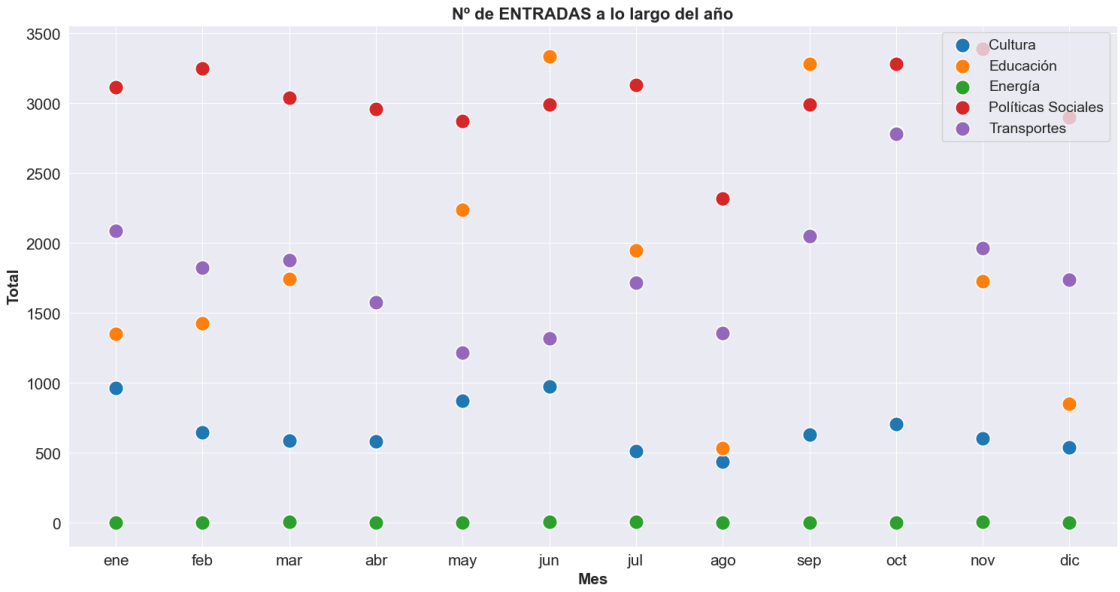
[274]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_cu_a.index, y=dftr_cu_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_ed_a.index, y=dftr_ed_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_en_a.index, y=dftr_en_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_ps_a.index, y=dftr_ps_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_tr_a.index, y=dftr_tr_a.tr_medio, s=200)

ax.set_title('TIEMPO MEDIO de RESPUESTA a lo largo del año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_a.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
```

```
                                                'oct', 'nov', 'dic'])  # Solo␣
↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Cultura',
                                     'Educación',
                                     'Energía',
                                     'Políticas Sociales',
                                     'Transportes'], fontsize=14);
```



[275]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_cu_a.index, y=dftr_cu_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_ed_a.index, y=dftr_ed_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_en_a.index, y=dftr_en_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_ps_a.index, y=dftr_ps_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_tr_a.index, y=dftr_tr_a.total, s=200)

ax.set_title('Nº de ENTRADAS a lo largo del año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Total', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_a.index, labels=['ene', 'feb', 'mar',
```

```
                                                'abr', 'may', 'jun',
                                                'jul', 'ago', 'sep',
                                                'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Cultura',
                                     'Educación',
                                     'Energía',
                                     'Políticas Sociales',
                                     'Transportes'], fontsize=14);
```

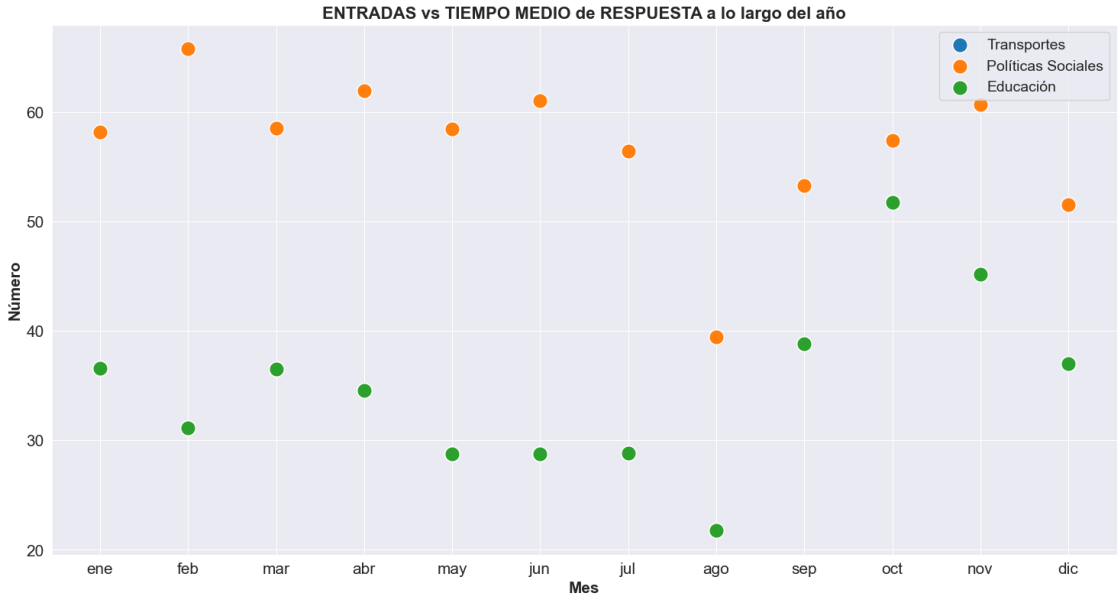


Comparación año (ver ratio abajo): Transportes vs Educación; Educación vs. Políticas Sociales

- Se observa como los meses realmente productivos de **Educación** son mayo y junio, y septiembre y octubre. Es decir, antes y después de las vacaciones de verano.
- **Transportes** tiene todo el año menor rendimiento, excepto el mes de agosto en el que Educación es peor.
- Políticas Sociales (naranja) es más estable a lo largo del tiempo.
- Transportes (azul) tiene un pico de rendimiento entre 2017 y 2018, y en la segunda mitad de 2019.
- Educación (verde) tiene varios picos de rendimiento a finales de 2020 y luego rinde muy bien hasta finales de 2022. Educación mejora a lo largo del tiempo.

[276]:
```
# Ratio de ENTRADAS / TIEMPO DE RESPUESTA MEDIO (más indica mejor desempeño)

dftr_tr_a = dftr_tr_a.copy()  # Copia para evitar warning.
dftr_tr_a.loc[:, 'tot_tr_medio'] = dftr_tr_a.total / dftr_tr_a.tr_medio

dftr_ps_a = dftr_ps_a.copy()  # Copia para evitar warning.
dftr_ps_a.loc[:, 'tot_tr_medio'] = dftr_ps_a.total / dftr_ps_a.tr_medio

dftr_ed_a = dftr_tr_a.copy()  # Copia para evitar warning.
dftr_ed_a.loc[:, 'tot_tr_medio'] = dftr_ed_a.total / dftr_ed_a.tr_medio
```

[277]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_tr_a.index, y=dftr_tr_a.tot_tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_ps_a.index, y=dftr_ps_a.tot_tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_ed_a.index, y=dftr_ed_a.tot_tr_medio, s=200)

ax.set_title('ENTRADAS vs TIEMPO MEDIO de RESPUESTA a lo largo del año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_a.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo␣
 ↪para poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Transportes',
                                     'Políticas Sociales',
                                     'Educación'], fontsize=14);
```



Estacionalidad Día Semana

- Los valores de todas las series son estables a lo largo de la semana.
- Descartando el valor anómalo de Energía el lunes, hay 2 grupos: Transportes y Políticas Sociales con tiempos medios de respuesta casi siempre por encima de 50 días, y Cultura, Educación y Energía con tiempos medios de respuesta casi siempre por debajo de 35 días.
- El número de entradas disminuye durante el fin de semana aunque el tiempo medio de respuesta no parece alterarse.

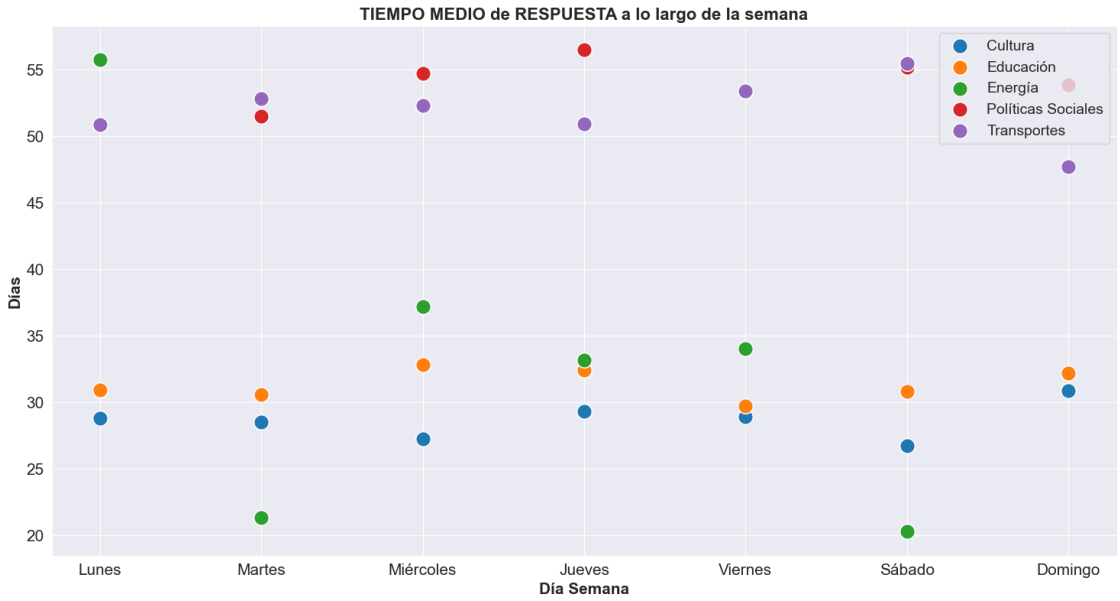
[278]:
```
dftr_tema_ds = dftr.groupby(['dia_semana', 'tema']).agg(
    tr_medio=('tiempo_respuesta', 'mean'),
    total=('tema', 'count')).reset_index()
```

[279]:
```
dftr_tema_ds = dftr_tema_ds.set_index('dia_semana')
```

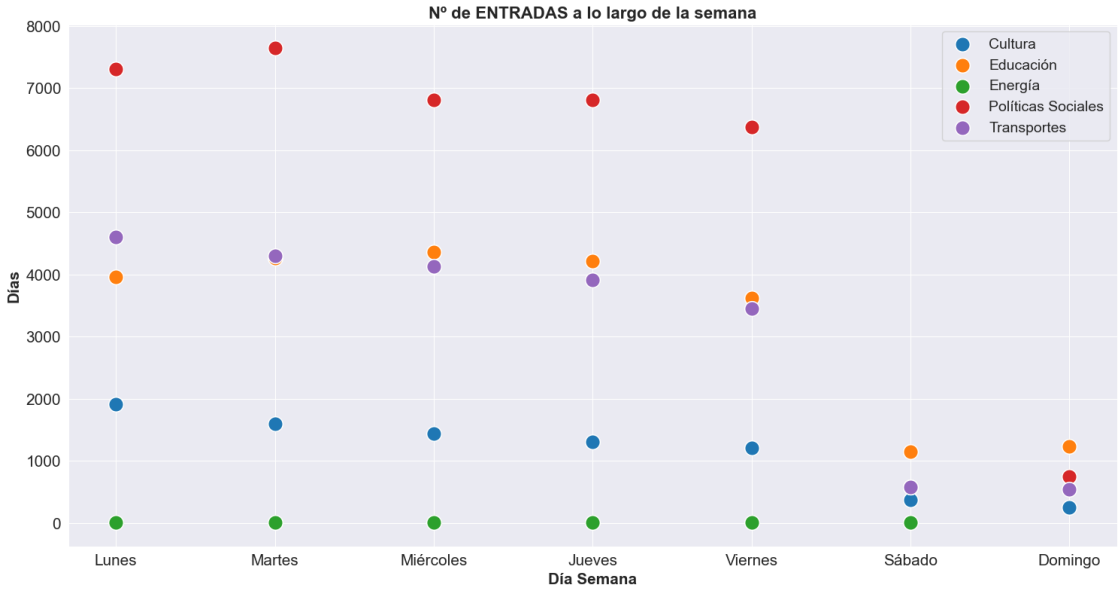
[280]:
```
# Cultura
dftr_cu_ds = dftr_tema_ds[dftr_tema_ds.tema == 'CULTURA']

# Educación
dftr_ed_ds = dftr_tema_ds[dftr_tema_ds.tema == 'EDUCACIÓN']

# Energía
dftr_en_ds = dftr_tema_ds[dftr_tema_ds.tema == 'ENERGÍA']

# Políticas Sociales
dftr_ps_ds = dftr_tema_ds[dftr_tema_ds.tema == 'POLÍTICAS SOCIALES']
```

```
# Transportes
dftr_tr_ds = dftr_tema_ds[dftr_tema_ds.tema == 'TRANSPORTES']
```

[281]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_cu_ds.index, y=dftr_cu_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_ed_ds.index, y=dftr_ed_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_en_ds.index, y=dftr_en_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_ps_ds.index, y=dftr_ps_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_tr_ds.index, y=dftr_tr_ds.tr_medio, s=200)

ax.set_title('TIEMPO MEDIO de RESPUESTA a lo largo de la semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día Semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Cultura',
                                     'Educación',
                                     'Energía',
                                     'Políticas Sociales',
                                     'Transportes'], fontsize=14);
```



[282]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_cu_ds.index, y=dftr_cu_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_ed_ds.index, y=dftr_ed_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_en_ds.index, y=dftr_en_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_ps_ds.index, y=dftr_ps_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_tr_ds.index, y=dftr_tr_ds.total, s=200)

ax.set_title('Nº de ENTRADAS a lo largo de la semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día Semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Cultura',
                                     'Educación',
                                     'Energía',
                                     'Políticas Sociales',
                                     'Transportes'], fontsize=14);
```



5.4.3 Canal

Mensual Internet

- El tiempo de respuesta se eleva durante 2019 y principios de 2020 (fenómenos endógeno: dos elecciones generales y unas elecciones autonómicas en Madrid): paralización de la administración. A partir de mitad de 2020 el tiempo de respuesta se estabiliza en valores inferiores a los de pre-pandemia.
- El número de entradas no se ve afectado por la pandemia y el ritmo creciente continua aunque con un pico de entradas a finales de 2021. A partir de entonces el ritmo se estabiliza en su ritmo creciente.

Llamadas 012

- El tiempo de respuesta mejora desde 2013 hasta finales de 2015, cuando vuelve a empeorar. Esta mejora de la eficiencia también se ve en Internet, pero aquí de forma más acusada. Tenemos igualmente el parón de 2019 en el tiempo de respuesta y la misma estabilización en valores más bajos a los de pre-pandemia.
- A diferencia de Internet, el número de las Llamdas 012 desciende a lo largo del tiempo, lo cual podría explicar su mejora de eficiencia en el tiempo de respuesta a lo largo del tiempo, pero no valdría la misma explicación para Internet. El tiempo de respuesta en los últimos años es aproximadamente 30 en Internet y el volumen de entradas de 500 (proporción 16,7), mientras que en Llamadas 012 el tiempo de respuesta es 20 y el número de entradas casi 100 (proporción 5): Internet es más de 3 veces mejor que Llamadas 012.

Presencial

- El tiempo de respuesta es el más estable a lo largo del tiempo y solo se altera significativamente en 2019 (elecciones lo empeoran) y 2020 (pandemia lo mejora). **La mejora se debe seguramente a la caída en picado del número de entradas durante la pandemia, aunque el empeoramiento durante el año electoral de 2019 no se explica porque exista un mayor volumen de entradas sino por la parálisis administrativa de las elecciones**. Tras la pandemia el tiempo de respuesta vuelve a aumentar.
- El número de entradas es muy estable alrededor de las 250 entradas. Con la pandemia cayó a cero y luego se ha recuperado casi en su totalidad a niveles pre-pandemia.

[283]:
```
dftr_canal = dftr.reset_index().groupby(['fecha_entrada', 'canal_entrada']).
↪agg(tr_medio=('tiempo_respuesta', 'mean'),
                                                                    ␣
↪total=('canal_entrada', 'count'))
```

[284]:
```
dftr_canal = dftr_canal.reset_index().set_index('fecha_entrada').
↪groupby('canal_entrada').resample('M') \
                                          .agg(tr_medio=('tr_medio', 'mean'),␣
↪total=('total', 'sum'))
```

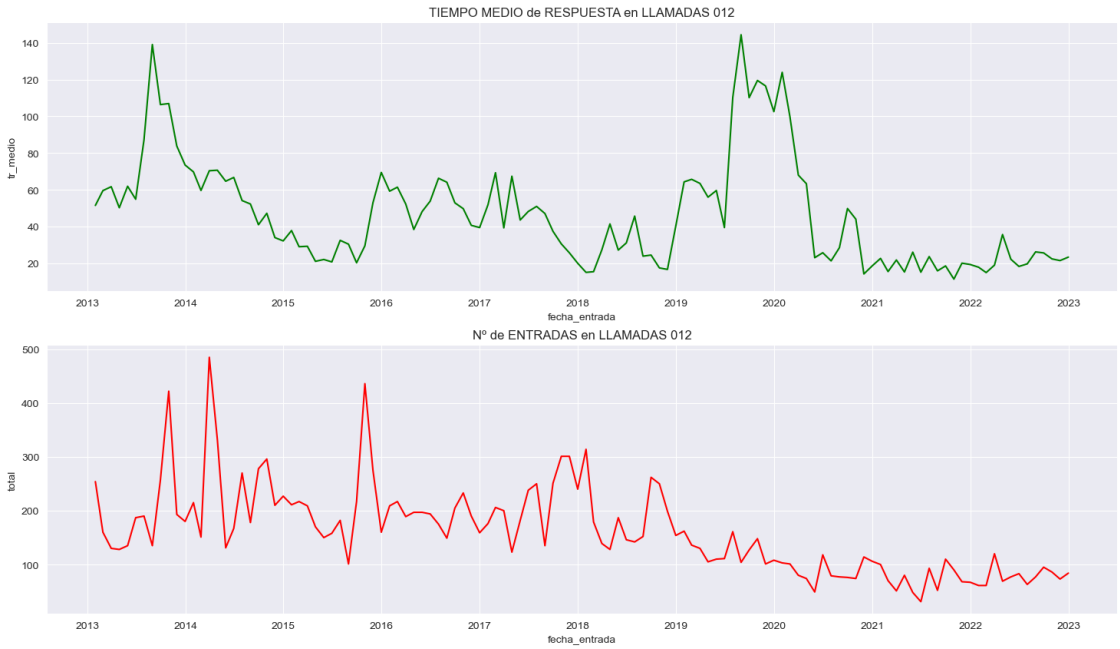
[285]:
```
dftr_canal = dftr_canal.reset_index().set_index('fecha_entrada')
```

[286]:
```
# Internet
dftr_in_mes = dftr_canal[dftr_canal.canal_entrada == 'INTERNET']

# Llamadas 012
dftr_ll_mes = dftr_canal[dftr_canal.canal_entrada == 'LLAMADAS 012']

# Presencial
dftr_pr_mes = dftr_canal[dftr_canal.canal_entrada == 'PRESENCIAL']
```

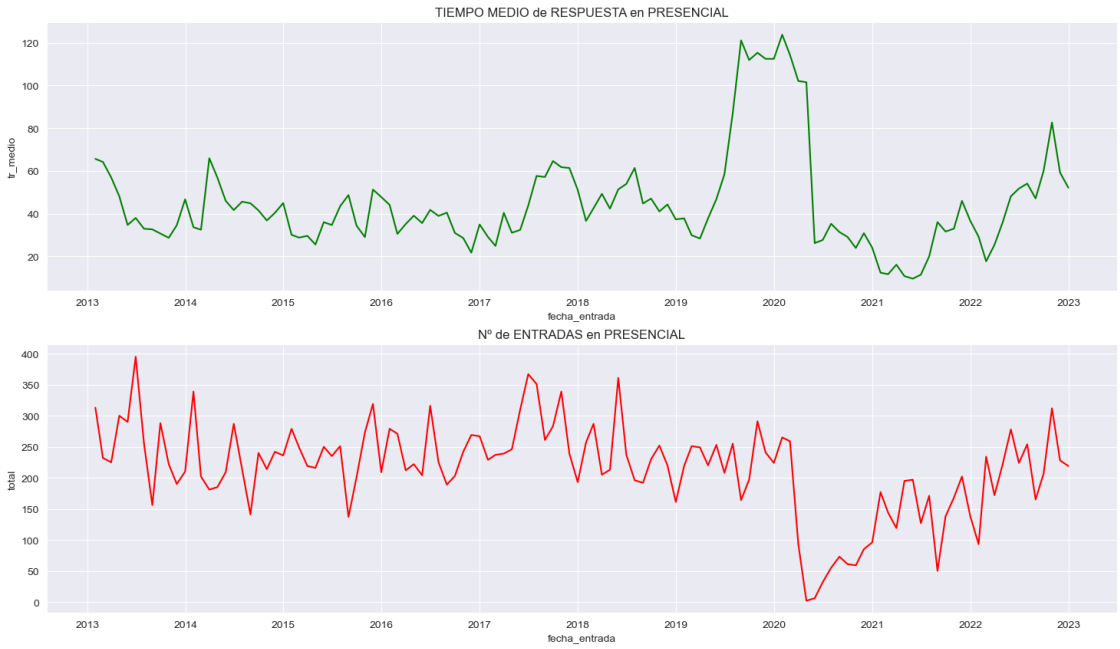
[287]:
```
f, ax = plt.subplots(2,1, figsize=(18,10))

sns.lineplot(ax=ax[0], data=dftr_in_mes, x=dftr_in_mes.index, y=dftr_in_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en INTERNET',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=dftr_in_mes, x=dftr_in_mes.index, y=dftr_in_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en INTERNET',␣
 ↪loc='center');
```

[288]:
```
f, ax = plt.subplots(2,1, figsize=(18,10))

sns.lineplot(ax=ax[0], data=dftr_ll_mes, x=dftr_ll_mes.index, y=dftr_ll_mes.
 ↪tr_medio, color='green') \
```

```
                .set_title(label='TIEMPO MEDIO de RESPUESTA en LLAMADAS 012',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=dftr_ll_mes, x=dftr_ll_mes.index, y=dftr_ll_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en LLAMADAS 012',␣
 ↪loc='center');
```



[289]:
```
f, ax = plt.subplots(2,1, figsize=(18,10))

sns.lineplot(ax=ax[0], data=dftr_pr_mes, x=dftr_pr_mes.index, y=dftr_pr_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en PRESENCIAL',␣
 ↪loc='center')

sns.lineplot(ax=ax[1], data=dftr_pr_mes, x=dftr_pr_mes.index, y=dftr_pr_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en PRESENCIAL',␣
 ↪loc='center');
```



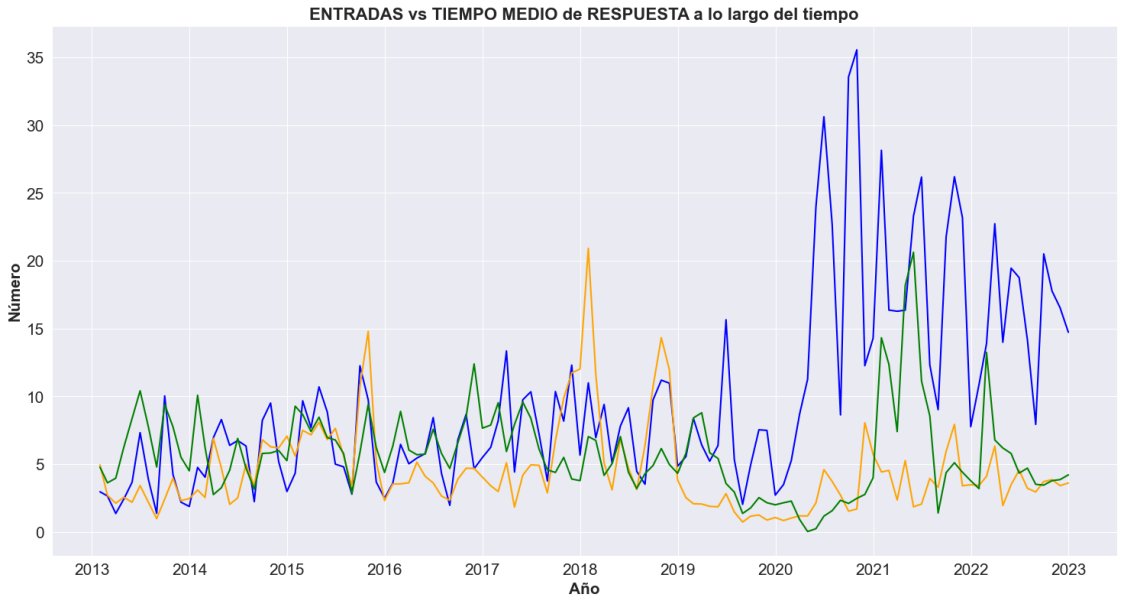
Comparación ratio ENTRADAS / TIEMPO_MEDIO_RESPUESTA

- Internet (azul) es claramente más eficiente en "entradas / tiempo de respuesta" a medida que pasa el tiempo. Presencial (verde) y Llamadas 012 (naranja) son más similares entre sí.

[290]:
```
# Ratio de ENTRADAS / TIEMPO DE RESPUESTA MEDIO (más indica mejor desempeño)
dftr_in_mes = dftr_in_mes.copy()
dftr_in_mes['tot_tr_medio'] = dftr_in_mes.total / dftr_in_mes.tr_medio

dftr_ll_mes = dftr_ll_mes.copy()
dftr_ll_mes['tot_tr_medio'] = dftr_ll_mes.total / dftr_ll_mes.tr_medio

dftr_pr_mes = dftr_pr_mes.copy()
dftr_pr_mes['tot_tr_medio'] = dftr_pr_mes.total / dftr_pr_mes.tr_medio
```

[291]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=dftr_in_mes.index, y=dftr_in_mes.tot_tr_medio,␣
 ↪color='blue')
sns.lineplot(ax=ax, x=dftr_ll_mes.index, y=dftr_ll_mes.tot_tr_medio,␣
 ↪color='orange')
sns.lineplot(ax=ax, x=dftr_pr_mes.index, y=dftr_pr_mes.tot_tr_medio,␣
 ↪color='green')

ax.set_title('ENTRADAS vs TIEMPO MEDIO de RESPUESTA a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))
```

```
ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15);
```



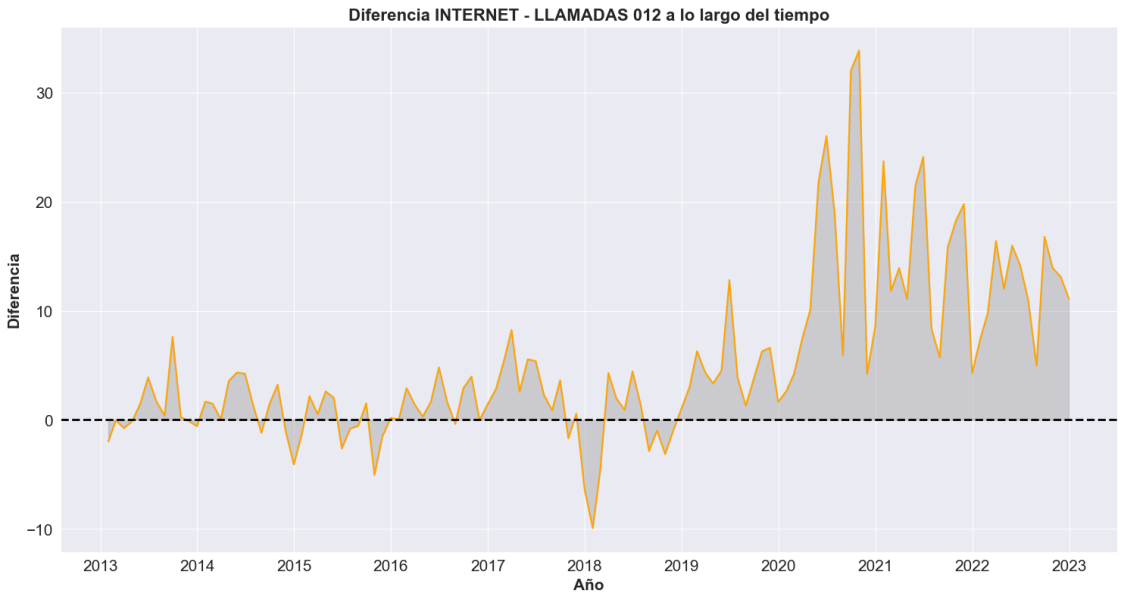
Diferencia de ratios y Diferencia acumulada

- Se observa el predominio claro de Internet respecto a Llamadas 012. Observando el gráfico acumulado esta superioridad es aún más clara.
- Respecto a Presencial, Internet domina en menor medida. Solo a partir de 2019 logra superar claramente a Presencial.
- Comparando Presencial frente a Llamadas 012 se observa un predominio constante desde el principio de Presencial sobre Llamadas 012, pero a una escala menor que Internet: algo menos de 160 frente a más de 600, en términos acumulados.

[292]:
```
# Diferencia Internet vs Llamadas 012
intr = dftr_in_mes.tot_tr_medio - dftr_ll_mes.tot_tr_medio
intr = intr.to_frame()

# Acumulación de diferencia
intr['dif_acumulada'] = intr.tot_tr_medio.cumsum()
```

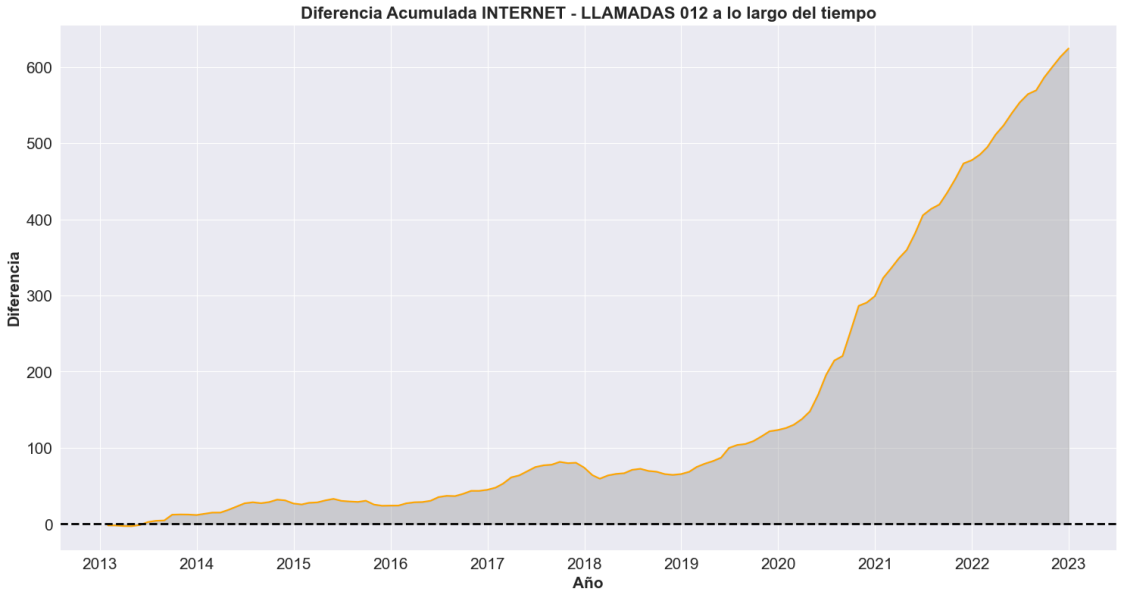
[293]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=intr.index, y=intr.tot_tr_medio, color='orange')
```

```
plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia INTERNET - LLAMADAS 012 a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



[294]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=intr.index, y=intr.dif_acumulada, color='orange')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada INTERNET - LLAMADAS 012 a lo largo del␣
 ↪tiempo',
```

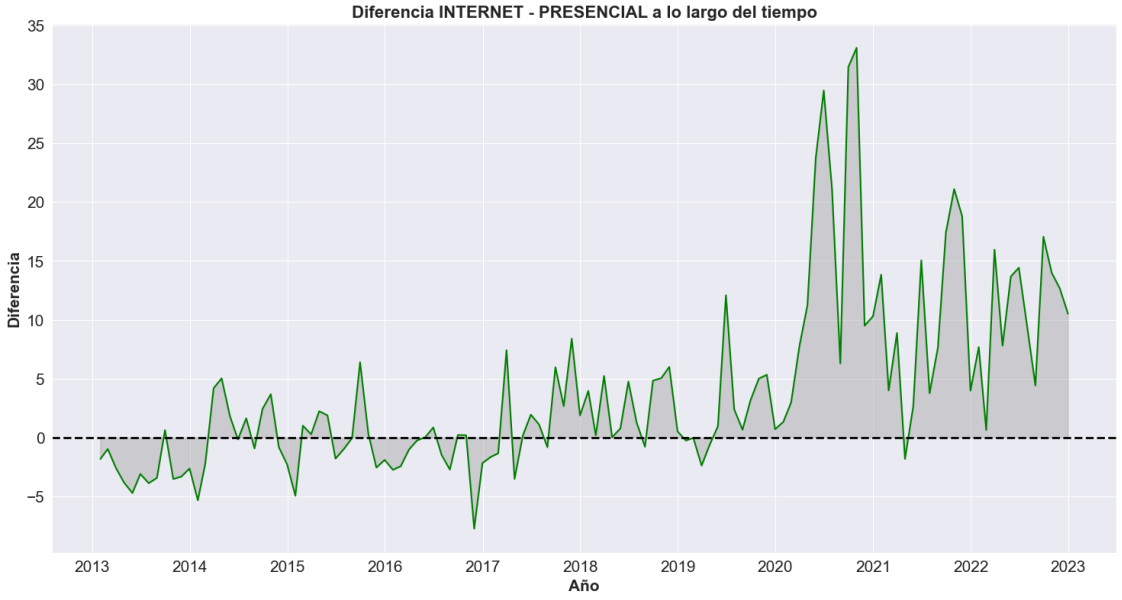
```
            fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



[295]:
```
# Diferencia Internet vs Presencial
inprtr = dftr_in_mes.tot_tr_medio - dftr_pr_mes.tot_tr_medio
inprtr = inprtr.to_frame()

# Acumulación de diferencia
inprtr['dif_acumulada'] = inprtr.tot_tr_medio.cumsum()
```

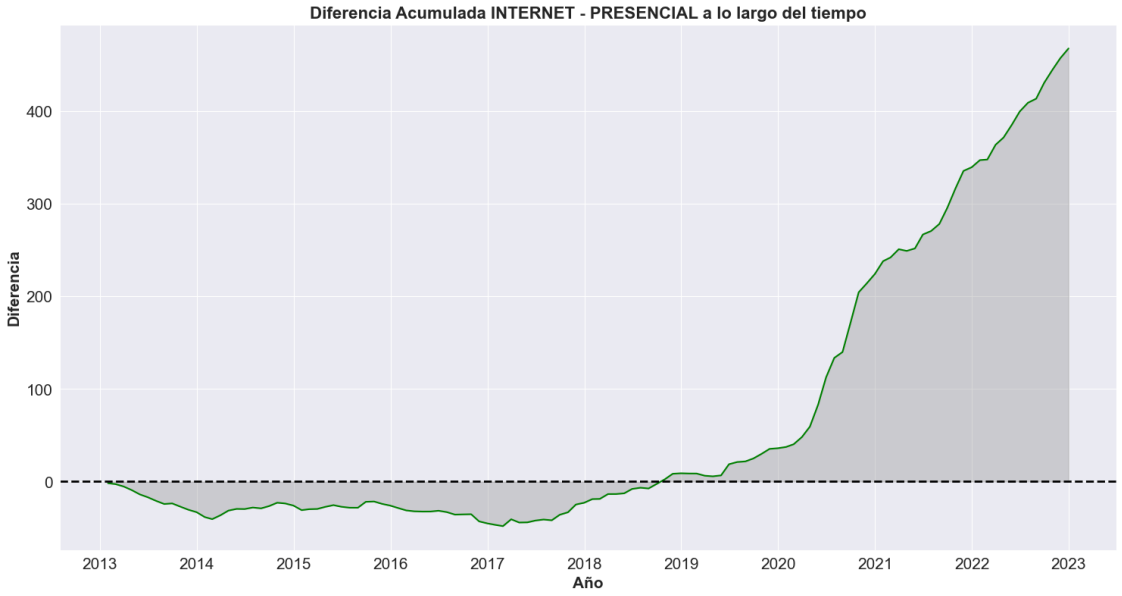
[296]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=inprtr.index, y=inprtr.tot_tr_medio, color='green')

plt.axhline(y=0, color='black', lw=2, ls='--')
```

```
ax.set_title('Diferencia INTERNET - PRESENCIAL a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



[297]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=inprtr.index, y=inprtr.dif_acumulada, color='green')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada INTERNET - PRESENCIAL a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
```

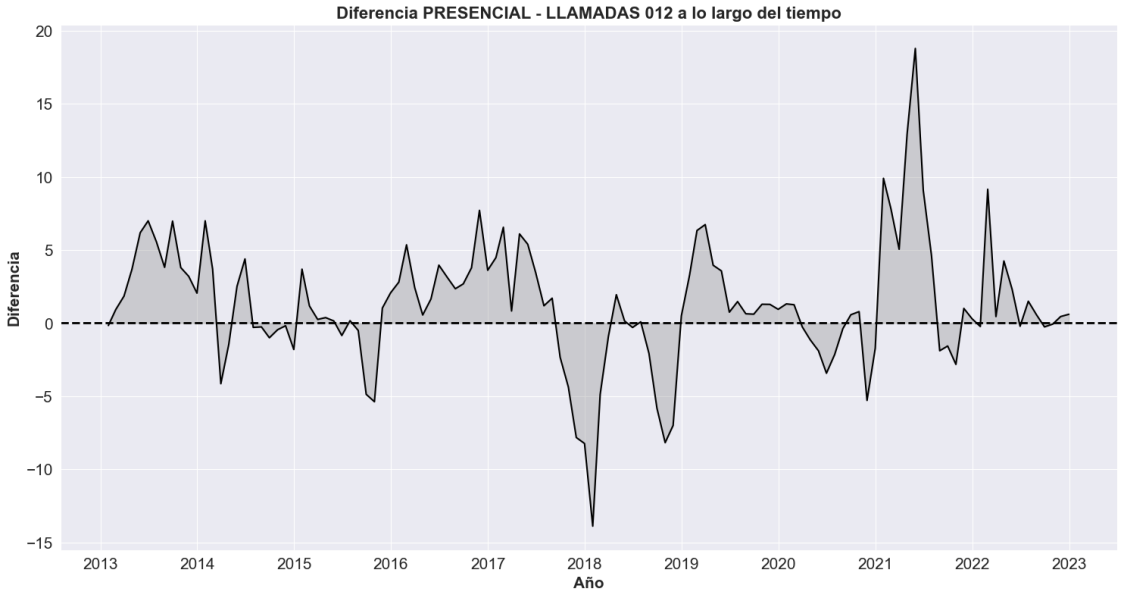
```
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



[298]:
```
# Diferencia Presencial vs Llamadas 012
prtr = dftr_pr_mes.tot_tr_medio - dftr_ll_mes.tot_tr_medio
prtr = prtr.to_frame()

# Acumulación de diferencia
prtr['dif_acumulada'] = prtr.tot_tr_medio.cumsum()
```

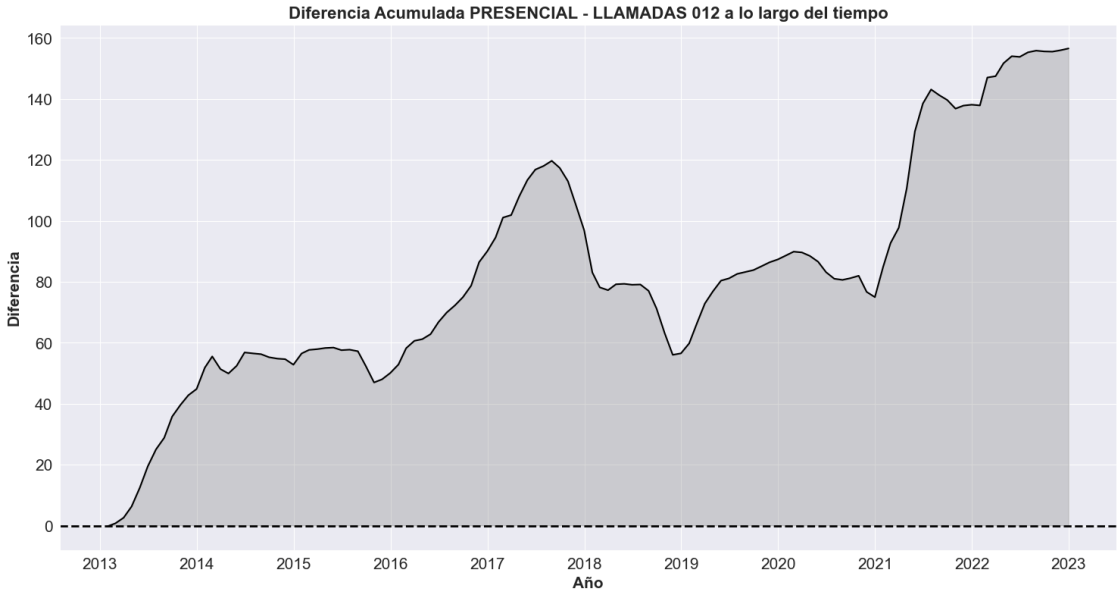
[299]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=prtr.index, y=prtr.tot_tr_medio, color='black')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia PRESENCIAL - LLAMADAS 012 a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))
```

```
ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



[300]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=prtr.index, y=prtr.dif_acumulada, color='black')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada PRESENCIAL - LLAMADAS 012 a lo largo del␣
 ↪tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))
```

```
ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



Estacionalidad Año

- A lo largo del año se observa el menor tiempo de respuesta de Internet, así como el peor rendimiento de Llamadas 012 durante los meses de enero a agosto, y el peor rendimiento de Presencial de septiembre a diciembre.
- A pesar del mayor número de entradas de Internet durante todos los meses de año su tiempo de respuesta es mejor. En cambio, Presencial, a pesar de tener el menor número de entradas durante todo el año su tiempo de respuesta es peor durante el último cuatrimestre.

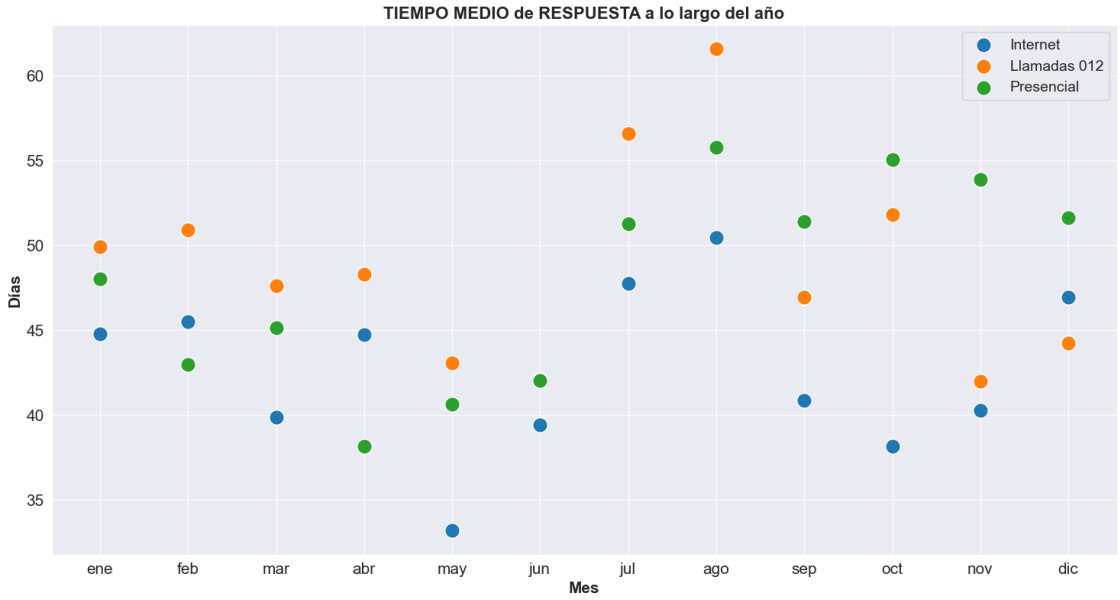
[301]:
```
dftr_canal_a = dftr.groupby(['mes', 'canal_entrada']).agg(
    tr_medio=('tiempo_respuesta', 'mean'),
    total=('canal_entrada', 'count')).reset_index()
```

[302]:
```
dftr_canal_a = dftr_canal_a.set_index('mes')
```

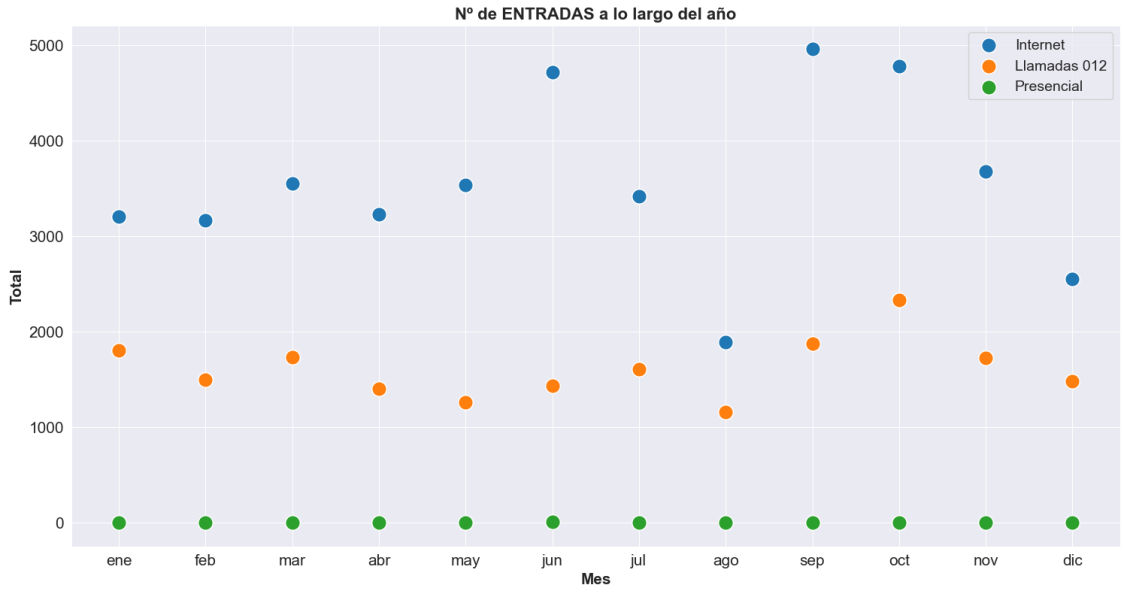
[303]:
```
# Internet
dftr_in_a = dftr_canal_a[dftr_canal_a.canal_entrada == 'INTERNET']
```

```
# Lamadas 012
dftr_ll_a = dftr_canal_a[dftr_canal_a.canal_entrada == 'LLAMADAS 012']

# Presencial
dftr_pr_a = dftr_canal_a[dftr_canal_a.canal_entrada == 'PRESENCIAL']
```

[304]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_in_a.index, y=dftr_in_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_ll_a.index, y=dftr_ll_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_pr_a.index, y=dftr_pr_a.tr_medio, s=200)

ax.set_title('TIEMPO MEDIO de RESPUESTA a lo largo del año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_a.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo para␣
 ↪poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Internet',
                                     'Llamadas 012',
                                     'Presencial'], fontsize=14);
```



[305]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_in_a.index, y=dftr_in_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_ll_a.index, y=dftr_ll_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_en_a.index, y=dftr_en_a.total, s=200)

ax.set_title('Nº de ENTRADAS a lo largo del año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Total', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_a.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo para␣
 ↪poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Internet',
                                     'Llamadas 012',
                                     'Presencial'], fontsize=14);
```



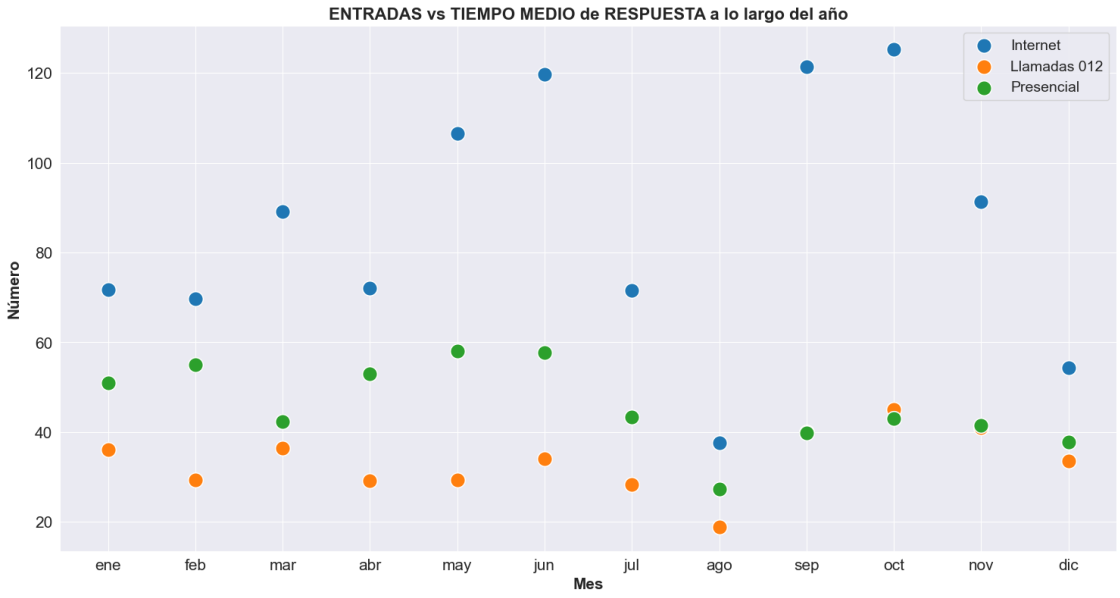
Comparación a lo largo del año (ratio)

- Internet es más eficiente a lo largo de todo el año.
- Se observa que Llamadas 012 tiene el peor rendimiento en todos los meses, excepto en octubre que es empeorado por Presencial.

[306]:
```
# Ratio de ENTRADAS / TIEMPO DE RESPUESTA MEDIO (más indica mejor desempeño)

dftr_in_a = dftr_in_a.copy()
dftr_in_a.loc[:, 'tot_tr_medio'] = dftr_in_a.total / dftr_in_a.tr_medio

dftr_ll_a = dftr_ll_a.copy()
dftr_ll_a.loc[:, 'tot_tr_medio'] = dftr_ll_a.total / dftr_ll_a.tr_medio

dftr_pr_a = dftr_pr_a.copy()
dftr_pr_a.loc[:, 'tot_tr_medio'] = dftr_pr_a.total / dftr_pr_a.tr_medio
```

[307]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_in_a.index, y=dftr_in_a.tot_tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_ll_a.index, y=dftr_ll_a.tot_tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_pr_a.index, y=dftr_pr_a.tot_tr_medio, s=200)

ax.set_title('ENTRADAS vs TIEMPO MEDIO de RESPUESTA a lo largo del año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))
```

```
ax.set_xticks(ticks=dftr_q_a.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo para␣
 ↪poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Internet',
                                     'Llamadas 012',
                                     'Presencial'], fontsize=14);
```



Estacionalidad Día Semana

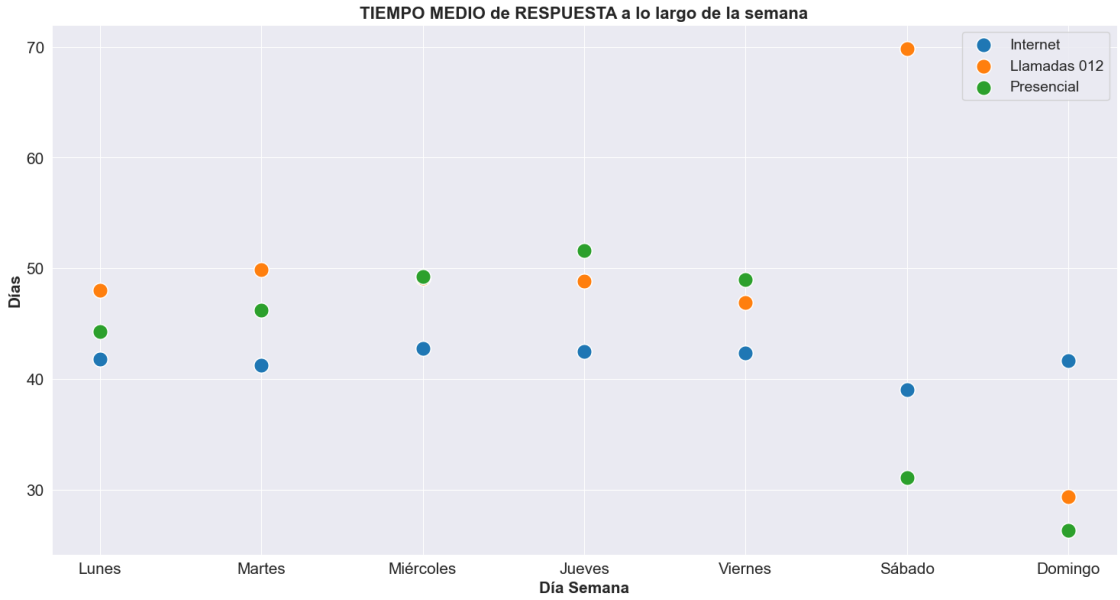
- El tiempo medio de respuesta el fin de semana es infierior en Presencial ¿Trabaja la administración el sábado y domingo recibiendo entradas? ¿Es trabajo desplazado desde el viernes? ¿Por qué las Llamadas 012 tienen un tiempo de respuesta tan alto si se supone que se resuelven en el momento?
- El número de entradas es claramente mayor en Internet y entre semana. Además, en fin de semana Presencial y Llamadas 012 reducen casi a cero las entradas.

[308]:
```
dftr_canal_ds = dftr.groupby(['dia_semana', 'canal_entrada']).agg(
    tr_medio=('tiempo_respuesta', 'mean'),
    total=('canal_entrada', 'count')).reset_index()
```

[309]:
```
dftr_canal_ds = dftr_canal_ds.set_index('dia_semana')
```

[310]:
```
# Internet
dftr_in_ds = dftr_canal_ds[dftr_canal_ds.canal_entrada == 'INTERNET']

# Llamadas 012
dftr_ll_ds = dftr_canal_ds[dftr_canal_ds.canal_entrada == 'LLAMADAS 012']

# Presencial
dftr_pr_ds = dftr_canal_ds[dftr_canal_ds.canal_entrada == 'PRESENCIAL']
```

[311]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_in_ds.index, y=dftr_in_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_ll_ds.index, y=dftr_ll_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_pr_ds.index, y=dftr_pr_ds.tr_medio, s=200)

ax.set_title('TIEMPO MEDIO de RESPUESTA a lo largo de la semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día Semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Internet',
                                     'Llamadas 012',
                                     'Presencial'], fontsize=14);
```



[312]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_in_ds.index, y=dftr_in_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_ll_ds.index, y=dftr_ll_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_pr_ds.index, y=dftr_pr_ds.total, s=200)

ax.set_title('Nº de ENTRADAS a lo largo de la semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día Semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=['Internet',
                                     'Llamadas 012',
                                     'Presencial'], fontsize=14);
```

5.4.4 Consejería ("nueva_conse")

Mensual

[313]:
```
dftr_nc = dftr.reset_index().groupby(['fecha_entrada', 'nueva_conse']).
↪agg(tr_medio=('tiempo_respuesta', 'mean'),
                                                                    ␣
↪total=('nueva_conse', 'count'))
```

[314]:
```
dftr_nc = dftr_nc.reset_index().set_index('fecha_entrada').
↪groupby('nueva_conse').resample('M') \
                                        .agg(tr_medio=('tr_medio', 'mean'),␣
↪total=('total', 'sum'))
```

[315]:
```
dftr_nc = dftr_nc.reset_index().set_index('fecha_entrada')
```

[316]:
```
# Transporte
dftr_tra_mes = dftr_nc[dftr_nc.nueva_conse == 'Transporte']
dftr_tra_mes = dftr_tra_mes.copy()
dftr_tra_mes['ratiottr'] = dftr_tra_mes.total / dftr_tra_mes.tr_medio

# Cultura
dftr_cul_mes = dftr_nc[dftr_nc.nueva_conse == 'Cultura']
dftr_cul_mes = dftr_cul_mes.copy()
dftr_cul_mes['ratiottr'] = dftr_cul_mes.total / dftr_cul_mes.tr_medio

# Digitalizacion
dftr_dig_mes = dftr_nc[dftr_nc.nueva_conse == 'Digitalizacion']
dftr_dig_mes = dftr_dig_mes.copy()
```

```
dftr_dig_mes['ratiottr'] = dftr_dig_mes.total / dftr_dig_mes.tr_medio

# Educacion
dftr_edu_mes = dftr_nc[dftr_nc.nueva_conse == 'Educacion']
dftr_edu_mes = dftr_edu_mes.copy()
dftr_edu_mes['ratiottr'] = dftr_edu_mes.total / dftr_edu_mes.tr_medio

# Subvenciones (Polítca Social)
dftr_sub_mes = dftr_nc[dftr_nc.nueva_conse == 'Subvenciones']
dftr_sub_mes = dftr_sub_mes.copy()
dftr_sub_mes['ratiottr'] = dftr_sub_mes.total / dftr_sub_mes.tr_medio

# Comunicación
dftr_com_mes = dftr_nc[dftr_nc.nueva_conse == 'Comunicacion']
dftr_com_mes = dftr_com_mes.copy()
dftr_com_mes['ratiottr'] = dftr_com_mes.total / dftr_com_mes.tr_medio

# Economía
dftr_eco_mes = dftr_nc[dftr_nc.nueva_conse == 'Economia']
dftr_eco_mes = dftr_eco_mes.copy()
dftr_eco_mes['ratiottr'] = dftr_eco_mes.total / dftr_eco_mes.tr_medio

# Medioambiental
dftr_med_mes = dftr_nc[dftr_nc.nueva_conse == 'Medioambiental']
dftr_med_mes = dftr_med_mes.copy()
dftr_med_mes['ratiottr'] = dftr_med_mes.total / dftr_med_mes.tr_medio
```

Principales volúmenes de entradas, valores actuales y tendencia:

- Transporte (100, decreciente). Ratio 7.5. Tiempo medio de respuesta 20.
- Educación (200, ligeramente creciente). Ratio 12. Tiempo medio de respuesta 15.
- Subvenciones-Políticas Sociales (425, creciente). Ratio 8. Tiempo medio de respuesta 52, aproximadamente.

[317]:
```
f, ax = plt.subplots(3,1, figsize=(18,15))

sns.lineplot(ax=ax[0], data=dftr_tra_mes, x=dftr_tra_mes.index, y=dftr_tra_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en nueva_conse␣
 ↪TRANSPORTE', loc='center')

sns.lineplot(ax=ax[1], data=dftr_tra_mes, x=dftr_tra_mes.index, y=dftr_tra_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en nueva_conse␣
 ↪TRANSPORTE', loc='center')

sns.lineplot(ax=ax[2], data=dftr_tra_mes, x=dftr_tra_mes.index, y=dftr_tra_mes.
 ↪ratiottr, color='grey') \
```

```
                    .set_title(label='ENTRADAS / TIEMPO RESPUESTA en␣
↪nueva_conse TRANSPORTE', loc='center');
```

Transporte

- El tiempo medio de respuesta ha disminuido a lo largo del tiempo, especialmente tras 2020.
- El número de entradas también ha disminuido a lo largo del tiempo, especialmente a partir de 2019.
- El ratio total de entradas respecto al tiempo medio de respuesta es estable en torno a 5, pero tiene dos picos de aumento importantes a principios de 2018 y a finales del mismo año.

[318]:
```
f, ax = plt.subplots(3,1, figsize=(18,15))

sns.lineplot(ax=ax[0], data=dftr_cul_mes, x=dftr_cul_mes.index, y=dftr_cul_mes.
↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en nueva_conse␣
↪CULTURA', loc='center')
```

```
sns.lineplot(ax=ax[1], data=dftr_cul_mes, x=dftr_cul_mes.index, y=dftr_cul_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en nueva_conse CULTURA',␣
 ↪loc='center')

sns.lineplot(ax=ax[2], data=dftr_cul_mes, x=dftr_cul_mes.index, y=dftr_cul_mes.
 ↪ratiottr, color='grey') \
                    .set_title(label='ENTRADAS / TIEMPO RESPUESTA en␣
 ↪nueva_conse CULTURA', loc='center');
```

Cultura

- El volumen de entradas es menor al de Transporte, pero también decreciente.
- Tiene los mismo picos en 2019 en el tiempo de respuesta.
- El ratio es ligeramente creciente, lo que demuestra mejora de la eficiencia. Dado que el número de entradas disminuye a partir de 2020 no podemos estar seguros del origen de la mejora.

[319]:
```
f, ax = plt.subplots(3,1, figsize=(18,15))

sns.lineplot(ax=ax[0], data=dftr_dig_mes, x=dftr_dig_mes.index, y=dftr_dig_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en nueva_conse␣
 ↪DIGITALIZACIÓN', loc='center')

sns.lineplot(ax=ax[1], data=dftr_dig_mes, x=dftr_dig_mes.index, y=dftr_dig_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en nueva_conse␣
 ↪DIGITALIZACIÓN', loc='center')

sns.lineplot(ax=ax[2], data=dftr_dig_mes, x=dftr_dig_mes.index, y=dftr_dig_mes.
 ↪ratiottr, color='grey') \
                    .set_title(label='ENTRADAS / TIEMPO RESPUESTA en␣
 ↪nueva_conse DIGITALIZACIÓN', loc='center');
```

Digitalización

- El número de entradas es bajo y aparentemente decreciente desde 2018.
- Hay un pico en el tiempo de respuesta en 2019 (igual que en Cultura y Transporte, pero el máximo se alcanza en 2021 y cuando las entradas están cayendo o son muy bajas. ¿Explicación?
- El ratio es muy bajo y se mantiene casi todo el tiempo por debajo de 1 (más días de tiempo medio de respuesta que número de entradas).

[320]:
```
f, ax = plt.subplots(3,1, figsize=(18,15))

sns.lineplot(ax=ax[0], data=dftr_edu_mes, x=dftr_edu_mes.index, y=dftr_edu_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en nueva_conse␣
 ↪EDUCACIÓN', loc='center')

sns.lineplot(ax=ax[1], data=dftr_edu_mes, x=dftr_edu_mes.index, y=dftr_edu_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en nueva_conse EDUCACIÓN',␣
 ↪loc='center')

sns.lineplot(ax=ax[2], data=dftr_edu_mes, x=dftr_edu_mes.index, y=dftr_edu_mes.
 ↪ratiottr, color='grey') \
                    .set_title(label='ENTRADAS / TIEMPO RESPUESTA en␣
 ↪nueva_conse EDUCACIÓN', loc='center');
```

Educación

- Reduce el tiempo medio de respuesta a lo largo del tiempo aunque con muchos altibajos. Desde 2021 la tendencia es más clara. Igualmente se observa el aumento en 2019.
- El número de entradas tiene tendencia creciente, aunque con más oscilaciones en los últimos años.
- El ratio de entradas/tiempo_respuesta se eleva a lo largo del tiempo, especialmente a partir de 2020. Actualmente está en 12 aproximadamente, que es el más alto de las "nueva_conse" visto hasta ahora.

[321]:
```
f, ax = plt.subplots(3,1, figsize=(18,15))

sns.lineplot(ax=ax[0], data=dftr_sub_mes, x=dftr_sub_mes.index, y=dftr_sub_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en nueva_conse␣
 ↪SUBVENCIONES (POLÍTICA SOCIAL)', loc='center')
```

```
sns.lineplot(ax=ax[1], data=dftr_sub_mes, x=dftr_sub_mes.index, y=dftr_sub_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en nueva_conse␣
 ↪SUBVENCIONES (POLÍTICA SOCIAL)', loc='center')

sns.lineplot(ax=ax[2], data=dftr_sub_mes, x=dftr_sub_mes.index, y=dftr_sub_mes.
 ↪ratiottr, color='grey') \
                    .set_title(label='ENTRADAS / TIEMPO RESPUESTA en␣
 ↪nueva_conse SUBVENCIONES (POLÍTICA SOCIAL)',
                               loc='center');
```

Subvenciones (Política Social)

- El tiempo de respuesta vuelve a marcar el pico en 2019, aunque esta vez más desplazado hacia principios de 2020. El tiempo de respuesta oscila entre 20 días y 80 días la mayor parte del periodo, pero desde 2021 muestra una tendencia creciente, pasando de menos de 20 a casi 60.
- El número de entradas es creciente a lo largo del tiempo, aunque con una caída a mitad de 2020, y luego una recuperación clara del número de entradas. El volumen de entradas actual

está por encima de 400 al mes, que es el más alta de todas las "nueva_conse" vistas hasta ahora.

- El ratio tiene subidas y caídas aunque desde 2020 para aumentar y mantenerse en el entorno de 8. Dado que las entradas están aumentando esto identifca una mejora en la eficiencia.

[322]:
```
f, ax = plt.subplots(3,1, figsize=(18,15))

sns.lineplot(ax=ax[0], data=dftr_com_mes, x=dftr_com_mes.index, y=dftr_com_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en nueva_conse␣
 ↪COMUNICACIÓN', loc='center')

sns.lineplot(ax=ax[1], data=dftr_com_mes, x=dftr_com_mes.index, y=dftr_com_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en nueva_conse␣
 ↪COMUNICACIÓN', loc='center')

sns.lineplot(ax=ax[2], data=dftr_com_mes, x=dftr_com_mes.index, y=dftr_com_mes.
 ↪ratiottr, color='grey') \
                    .set_title(label='ENTRADAS / TIEMPO RESPUESTA en␣
 ↪nueva_conse COMUNICACIÓN', loc='center');
```

Comunicación

- El tiempo de respuesta se ha estabilizado en los últimos años a partir de mitad de 2020, aunque parece que hay una tendencia decreciente. Igualmente, hay un doble pico máximo en 2019 y 2020.
- El número de entradas ha disminuido y parece que se ha estabilizado en números bajos. El número de entradas es muy bajo.
- El ratio es bajísimo (0.0-0.1) y decreciente, lo cual habla de la ineficiencia de la gestión a pesar de que el volumen de entradas es muy bajo y cada vez es menor.

[323]:
```
f, ax = plt.subplots(3,1, figsize=(18,15))

sns.lineplot(ax=ax[0], data=dftr_eco_mes, x=dftr_eco_mes.index, y=dftr_eco_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en nueva_conse␣
 ↪ECONOMÍA', loc='center')
```

```
sns.lineplot(ax=ax[1], data=dftr_eco_mes, x=dftr_eco_mes.index, y=dftr_eco_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en nueva_conse ECONOMÍA',␣
 ↪loc='center')

sns.lineplot(ax=ax[2], data=dftr_eco_mes, x=dftr_eco_mes.index, y=dftr_eco_mes.
 ↪ratiottr, color='grey') \
                    .set_title(label='ENTRADAS / TIEMPO RESPUESTA en␣
 ↪nueva_conse ECONOMÍA', loc='center');
```

Economía

- La oscilación en el tiempo de respuesta hasta mediados de 2020 es muy grande. A partir de ahí se estabiliza en valores más bajos (25 aproximadamente). Hay un pico en 2020 pero no destaca tanto como en casos vistos anteriormente.
- El número de entradas es muy bajo y discontinuo. Desde 2019 parece que tiende a disminuir. El pico de entradas destacado se produce a mitad de 2017. ¿Por qué?

- El ratio oscila en valores muy bajos (menor a 0.20), lo que indica que junto con el número de entradas tan bajo una clara ineficiencia.

[324]:
```
f, ax = plt.subplots(3,1, figsize=(18,15))

sns.lineplot(ax=ax[0], data=dftr_med_mes, x=dftr_med_mes.index, y=dftr_med_mes.
 ↪tr_medio, color='green') \
                .set_title(label='TIEMPO MEDIO de RESPUESTA en nueva_conse␣
 ↪MEDIOAMBIENTAL', loc='center')

sns.lineplot(ax=ax[1], data=dftr_med_mes, x=dftr_med_mes.index, y=dftr_med_mes.
 ↪total, color='red') \
                    .set_title(label='Nº de ENTRADAS en nueva_conse␣
 ↪MEDIOAMBIENTAL', loc='center')

sns.lineplot(ax=ax[2], data=dftr_med_mes, x=dftr_med_mes.index, y=dftr_med_mes.
 ↪ratiottr, color='grey') \
                    .set_title(label='ENTRADAS / TIEMPO RESPUESTA en␣
 ↪nueva_conse MEDIOAMBIENTAL', loc='center');
```

Medioambiental

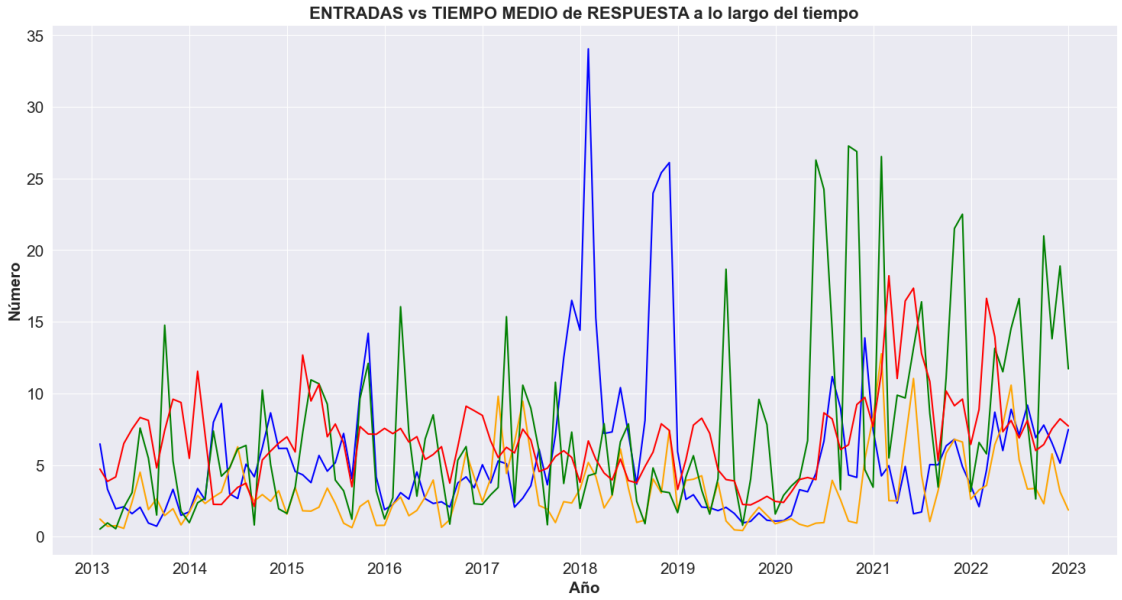
- El número de entradas es muy bajo o cero. Lo que hace difícil comentar nada. A partir de 2022 parece que hay más entradas.

Comparación ratio ENTRADAS / TIEMPO_MEDIO_RESPUESTA

- En 2018 Transportes tuvo dos picos importantes de mejora.
- A partir de 2020 Educación y Subvención-Políticas Sociales han tenido mejoras, especialmente Educación.
- Cultura tiene menos cambios pero también mejora a partir de 2020.

[325]:
```
# Ratio de ENTRADAS / TIEMPO DE RESPUESTA MEDIO (más indica mejor desempeño)

dftr_tra_mes = dftr_tra_mes.copy()
dftr_tra_mes.loc[:, 'tot_tr_medio'] = dftr_tra_mes.total / dftr_tra_mes.tr_medio

dftr_cul_mes = dftr_cul_mes.copy()
dftr_cul_mes.loc[:, 'tot_tr_medio'] = dftr_cul_mes.total / dftr_cul_mes.tr_medio

dftr_edu_mes = dftr_edu_mes.copy()
dftr_edu_mes.loc[:, 'tot_tr_medio'] = dftr_edu_mes.total / dftr_edu_mes.tr_medio

dftr_sub_mes = dftr_sub_mes.copy()
dftr_sub_mes.loc[:, 'tot_tr_medio'] = dftr_sub_mes.total / dftr_sub_mes.tr_medio
```

[326]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=dftr_tra_mes.index, y=dftr_tra_mes.tot_tr_medio,␣
 ↪color='blue')
sns.lineplot(ax=ax, x=dftr_cul_mes.index, y=dftr_cul_mes.tot_tr_medio,␣
 ↪color='orange')
sns.lineplot(ax=ax, x=dftr_edu_mes.index, y=dftr_edu_mes.tot_tr_medio,␣
 ↪color='green')
sns.lineplot(ax=ax, x=dftr_sub_mes.index, y=dftr_sub_mes.tot_tr_medio,␣
 ↪color='red')

ax.set_title('ENTRADAS vs TIEMPO MEDIO de RESPUESTA a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15);
```

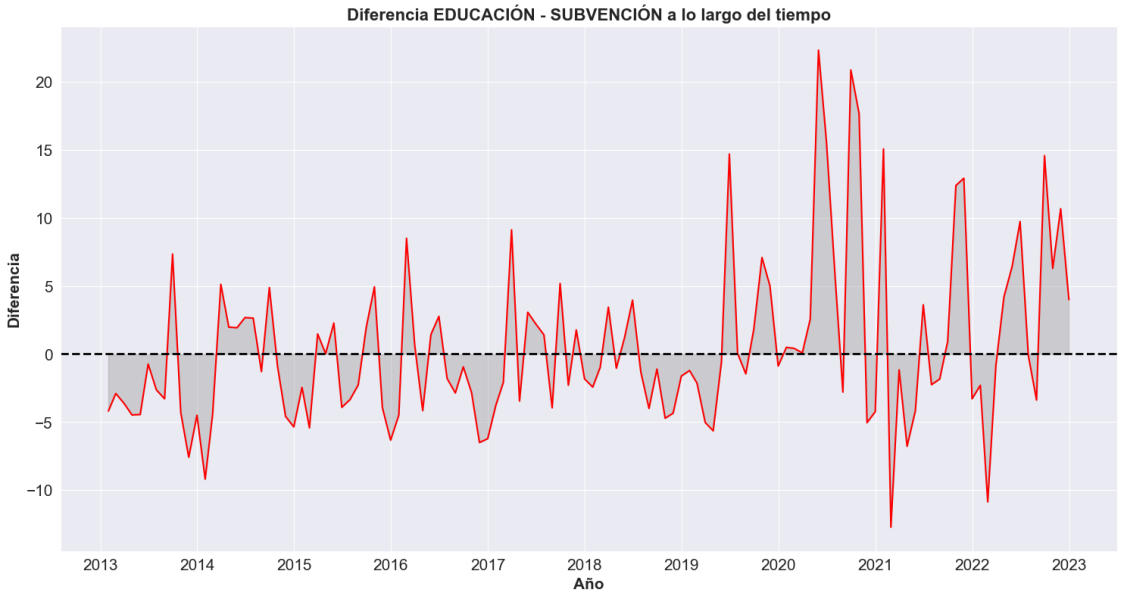


Diferencia de ratios y Diferencia acumulada

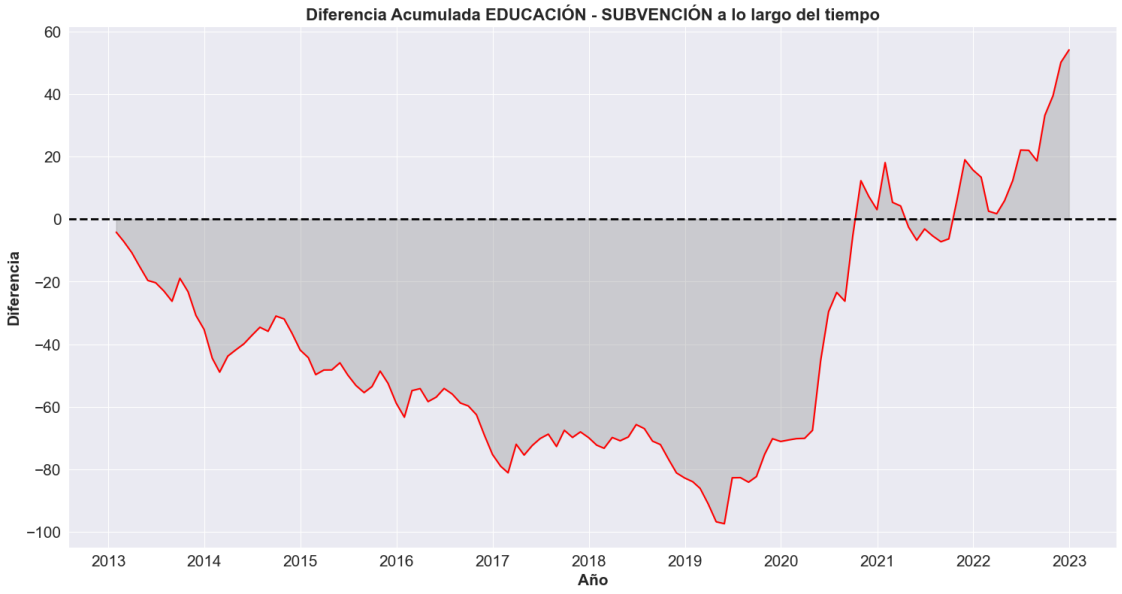
[327]:
```
# Diferencia Educación vs Subvención-Políticas Sociales
edutr = dftr_edu_mes.tot_tr_medio - dftr_sub_mes.tot_tr_medio
edutr = edutr.to_frame()

# Acumulación de diferencia
edutr['dif_acumulada'] = edutr.tot_tr_medio.cumsum()
```

[328]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=edutr.index, y=edutr.tot_tr_medio, color='red')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia EDUCACIÓN - SUBVENCIÓN a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]
```

```
ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



[329]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=edutr.index, y=edutr.dif_acumulada, color='red')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada EDUCACIÓN - SUBVENCIÓN a lo largo del␣
 ↪tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```

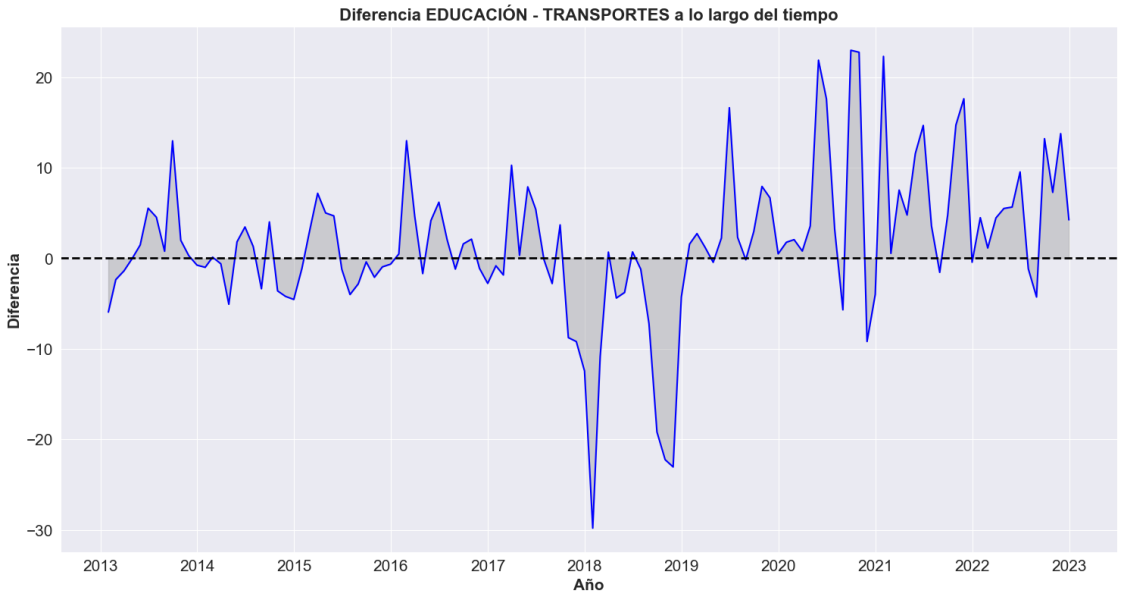


- Se observa como hasta 2020 Subvenión-Políticas Sociales ha sido más eficiente que Educación y a partir de ese año Educación supera a Subvención.

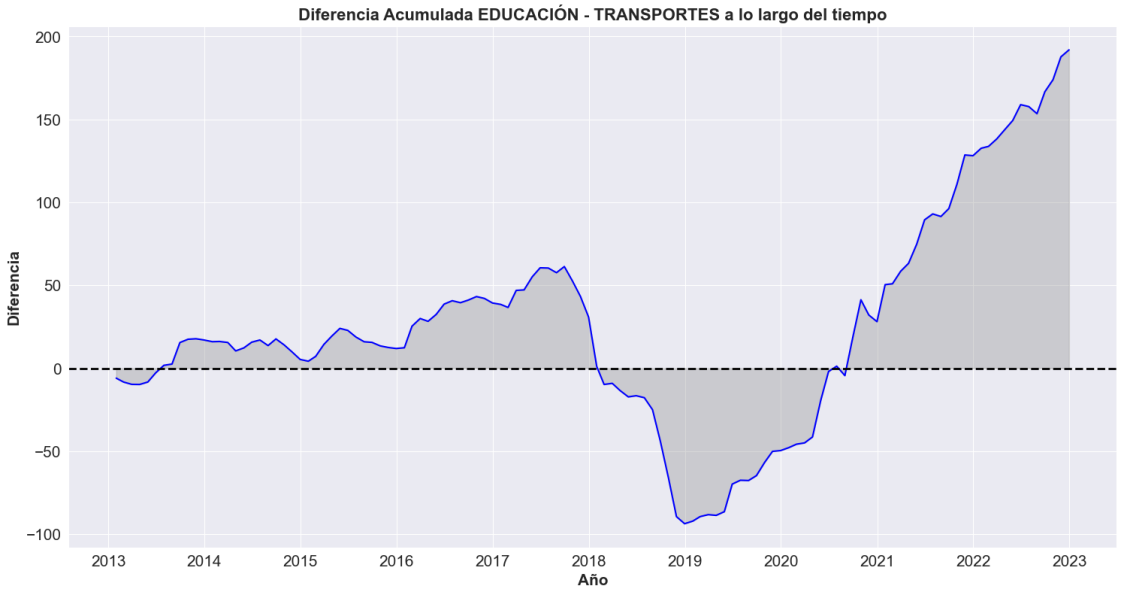
[330]:
```
# Diferencia Educación vs Transportes
edutratr = dftr_edu_mes.tot_tr_medio - dftr_tra_mes.tot_tr_medio
edutratr = edutratr.to_frame()

# Acumulación de diferencia
edutratr['dif_acumulada'] = edutratr.tot_tr_medio.cumsum()
```

[331]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=edutratr.index, y=edutratr.tot_tr_medio, color='blue')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia EDUCACIÓN - TRANSPORTES a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
```

```
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



[332]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=edutratr.index, y=edutratr.dif_acumulada, color='blue')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada EDUCACIÓN - TRANSPORTES a lo largo del␣
 ↪tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```

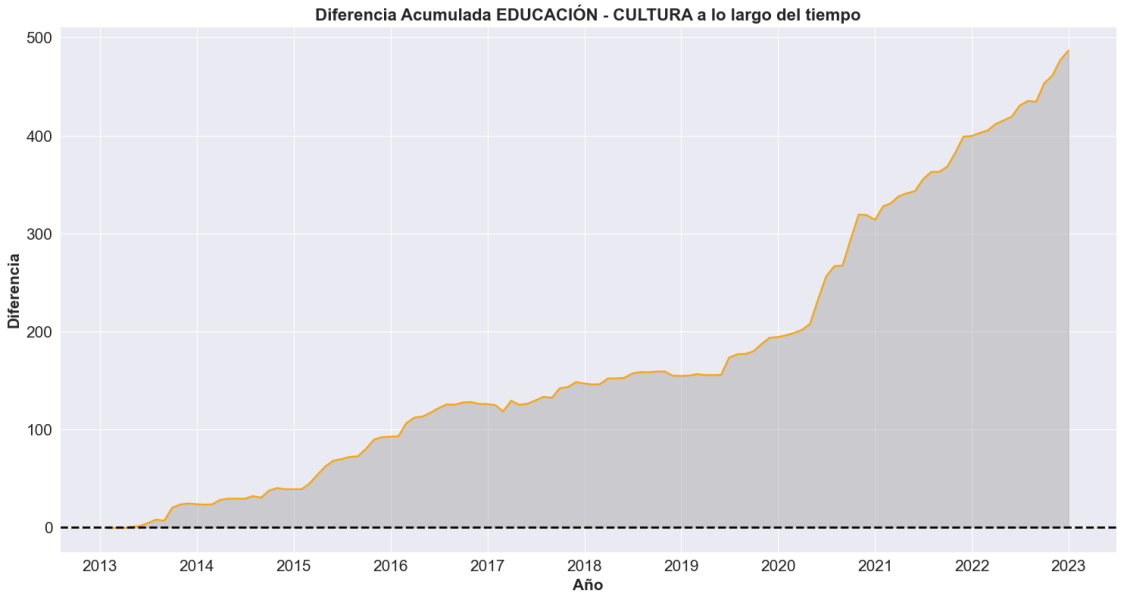


- Educación es más eficiente que Transportes hasta 2018, luego hasta mediados de 2020 Transportes es mejor que Educación, y a partir de ahí Educación supera rápida y definitivamente a Transportes.

[333]:
```
# Diferencia Educación vs Cultura
educultr = dftr_edu_mes.tot_tr_medio - dftr_cul_mes.tot_tr_medio
educultr = educultr.to_frame()

# Acumulación de diferencia
educultr['dif_acumulada'] = educultr.tot_tr_medio.cumsum()
```

[334]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=educultr.index, y=educultr.dif_acumulada, color='orange')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada EDUCACIÓN - CULTURA a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]
```

```
x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);

# plt.savefig("multimedia/
 ↪dif_acumulada_ratio_tiempo_respuesta_edu_cultura_historico.png")
```

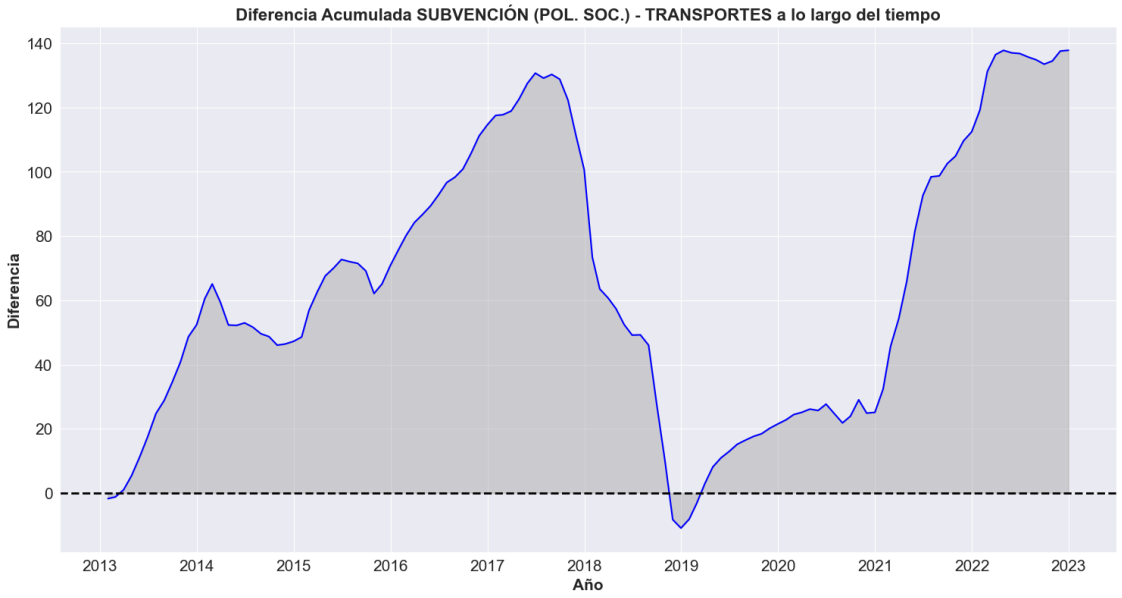


- Dominio de prinicipio a fin de Educación sobre Cultura.

[335]:
```
# Diferencia Subvención vs Transportes
subtratr = dftr_sub_mes.tot_tr_medio - dftr_tra_mes.tot_tr_medio
subtratr = subtratr.to_frame()

# Acumulación de diferencia
subtratr['dif_acumulada'] = subtratr.tot_tr_medio.cumsum()
```

[336]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=subtratr.index, y=subtratr.dif_acumulada, color='blue')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada SUBVENCIÓN (POL. SOC.) - TRANSPORTES a lo␣
 ↪largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))
```

```
ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```

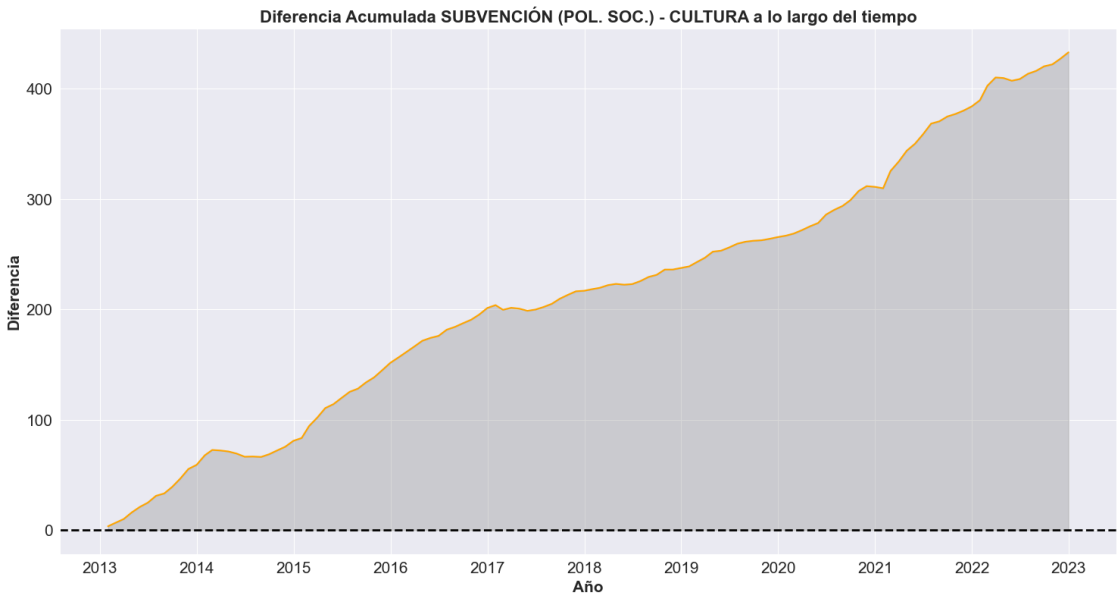


- Debido al conocido pico en 2018 de Transportes el gráfico muestra la recuperación de toda la desventaja que llevaba frente a Subvención, pero luego vuelve a perder toda la ventaja.

[337]:
```
# Diferencia Subvención vs Cultura
subcultr = dftr_sub_mes.tot_tr_medio - dftr_cul_mes.tot_tr_medio
subcultr = subcultr.to_frame()

# Acumulación de diferencia
subcultr['dif_acumulada'] = subcultr.tot_tr_medio.cumsum()
```

[338]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=subcultr.index, y=subcultr.dif_acumulada, color='orange')
```

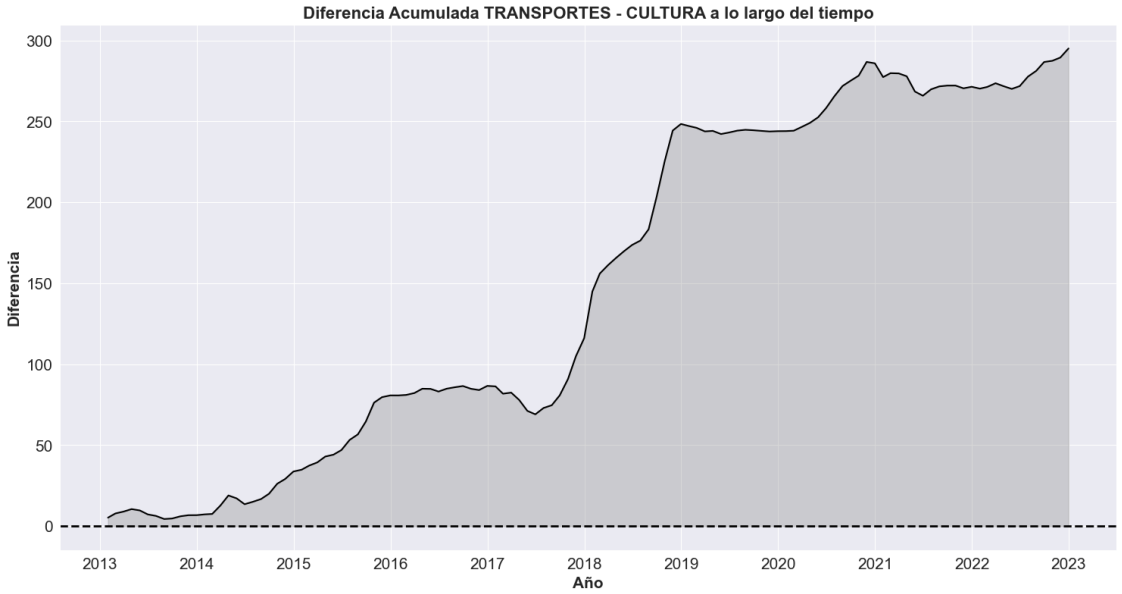
```
plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada SUBVENCIÓN (POL. SOC.) - CULTURA a lo largo␣
 ↪del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



- Patrón similar al que se veía entre Educación y Cultura.

[339]:
```
# Diferencia Transportes vs Cultura
tracultr = dftr_tra_mes.tot_tr_medio - dftr_cul_mes.tot_tr_medio
tracultr = tracultr.to_frame()

# Acumulación de diferencia
tracultr['dif_acumulada'] = tracultr.tot_tr_medio.cumsum()
```

[340]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.lineplot(ax=ax, x=tracultr.index, y=tracultr.dif_acumulada, color='black')

plt.axhline(y=0, color='black', lw=2, ls='--')

ax.set_title('Diferencia Acumulada TRANSPORTES - CULTURA a lo largo del tiempo',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Año", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Diferencia', fontsize=15, fontdict=dict(weight='bold'))

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

l1 = ax.lines[0]

x1 = l1.get_xydata()[:, 0]
y1 = l1.get_xydata()[:, 1]

ax.fill_between(x1, y1, color="grey", alpha=0.3);
```



- Patrón similar al de las anteriores consejerías (nueva_conse) frente a Cultura. Indica que Cultura es la más distinta del grupo.

Estacionalidad Año

- Durante el segundo semestre del año Economía tiene el tiempo de respuesta mayor (a excep-

ción de septiembre y noviembre que lo tiene Medioambiental con valores muy atípicos).

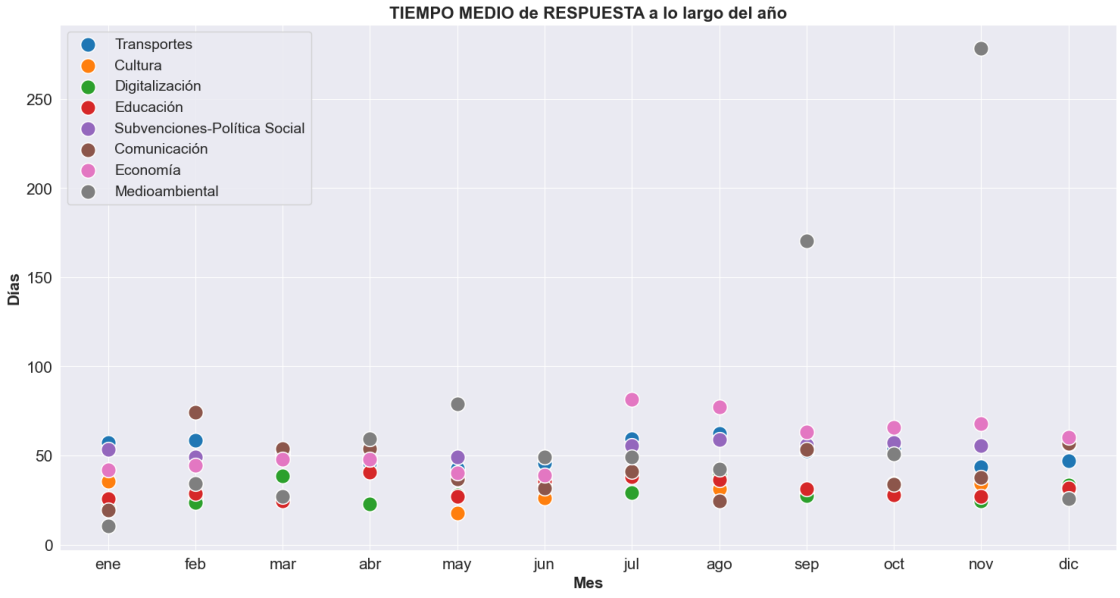
- El tiempo de respuesta mayor parece estar en julio y agosto.
- Transportes y Subvenciones (Política Social) tienen los peores tiempos de respuesta a lo largo de todo el año.
- Transportes, Subvenciones y Cultura son bajos en abril, mayo y junio, mientras que Educación tiene su máximo en abril y un valor muy alto en junio. Los peores tiempos de respuesta de Educación se dan en junio, julio, agosto y diciembre (vacaciones).
- Transportes tiene los mayores tiempos de respuesta en julio y agosto (verano).
- Se observa que las categoría más numerosas son: Subvenciones-Política Social, Educación, Transportes y Cultura (500-3500 entradas por mes). El resto tienen menos de 80 entradas por mes.
- El bajo número de entradas de categogrías como Economía y Medioambiental hace inexplicable los altos tiempos de respuesta que tienen. Son muy ineficientes.

[341]:
```
dftr_nc_a = dftr.groupby(['mes', 'nueva_conse']).agg(
    tr_medio=('tiempo_respuesta', 'mean'),
    total=('nueva_conse', 'count')).reset_index()
```

[342]:
```
dftr_nc_a = dftr_nc_a.set_index('mes')
```

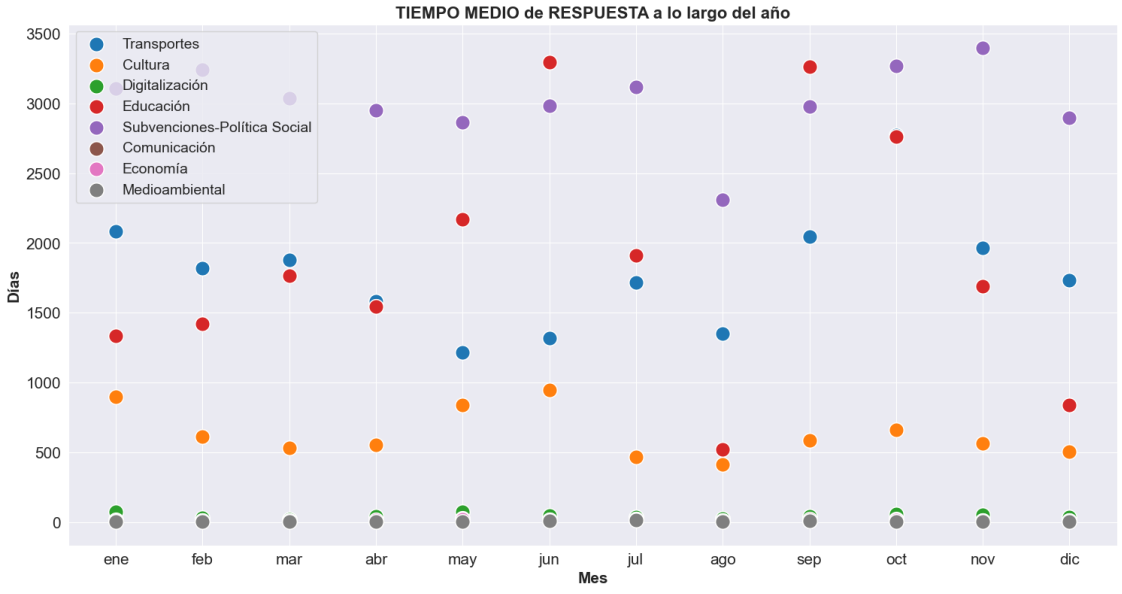
[343]:
```
# Transporte
dftr_tra_a = dftr_nc_a[dftr_nc_a.nueva_conse == 'Transporte']

# Cultura
dftr_cul_a = dftr_nc_a[dftr_nc_a.nueva_conse == 'Cultura']

# Digitalización
dftr_dig_a = dftr_nc_a[dftr_nc_a.nueva_conse == 'Digitalizacion']

# Educación
dftr_edu_a = dftr_nc_a[dftr_nc_a.nueva_conse == 'Educacion']

# Subvenciones (Políta Social)
dftr_sub_a = dftr_nc_a[dftr_nc_a.nueva_conse == 'Subvenciones']

# Comunicación
dftr_com_a = dftr_nc_a[dftr_nc_a.nueva_conse == 'Comunicacion']

# Economía
dftr_eco_a = dftr_nc_a[dftr_nc_a.nueva_conse == 'Economia']

# Medioambiental
dftr_med_a = dftr_nc_a[dftr_nc_a.nueva_conse == 'Medioambiental']
```

[344]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_tra_a.index, y=dftr_tra_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_cul_a.index, y=dftr_cul_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_dig_a.index, y=dftr_dig_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_edu_a.index, y=dftr_edu_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_sub_a.index, y=dftr_sub_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_com_a.index, y=dftr_com_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_eco_a.index, y=dftr_eco_a.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_med_a.index, y=dftr_med_a.tr_medio, s=200)

ax.set_title('TIEMPO MEDIO de RESPUESTA a lo largo del año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_a.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo para␣
 ↪poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper left', labels=[
    'Transportes',
                                    'Cultura',
                                    'Digitalización',
                                    'Educación',
                                    'Subvenciones-Política Social',
                                    'Comunicación',
                                    'Economía',
                                    'Medioambiental'
], fontsize=14);
```



[345]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_tra_a.index, y=dftr_tra_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_cul_a.index, y=dftr_cul_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_dig_a.index, y=dftr_dig_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_edu_a.index, y=dftr_edu_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_sub_a.index, y=dftr_sub_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_com_a.index, y=dftr_com_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_eco_a.index, y=dftr_eco_a.total, s=200)
sns.scatterplot(ax=ax, x=dftr_med_a.index, y=dftr_med_a.total, s=200)

ax.set_title('TIEMPO MEDIO de RESPUESTA a lo largo del año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_a.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo para␣
 ↪poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper left', labels=[
```

```
    'Transportes',
    'Cultura',
    'Digitalización',
    'Educación',
    'Subvenciones-Política Social',
    'Comunicación',
    'Economía',
    'Medioambiental'
    ], fontsize=14);
```



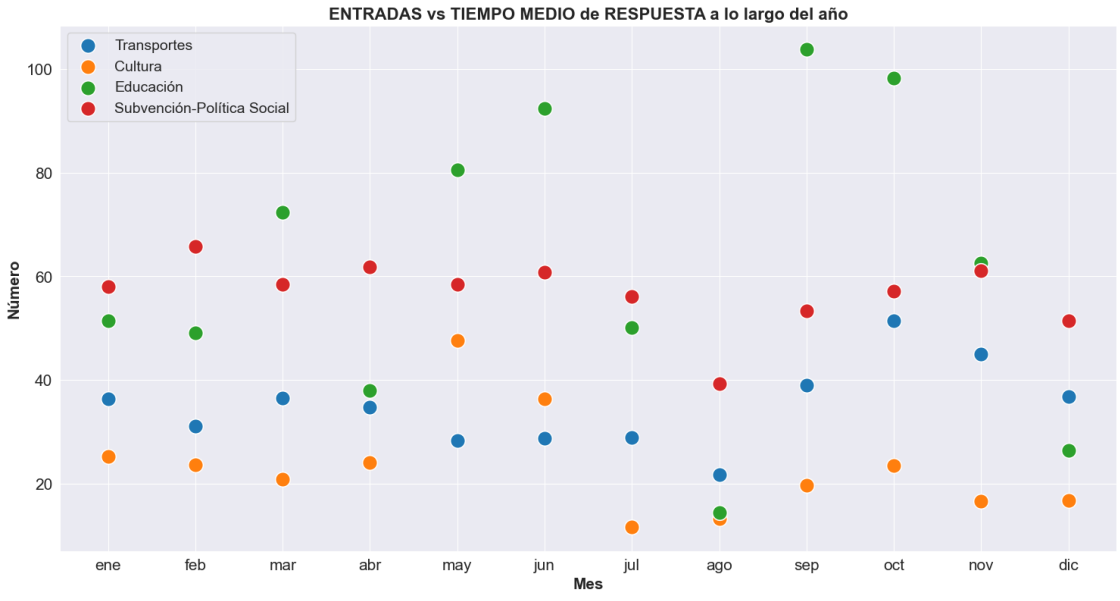
Comparación a lo largo del año (ratio)

- Educación es más eficiente en varios meses pero oscila más a lo largo del año que Subvención.
- Cultura es la más ineficiente en casi todos los meses, a excepción de mayo y junio.
- Transportes es la segunda más ineficiente, la peor en mayo y junio.

[346]:
```
# Ratio de ENTRADAS / TIEMPO DE RESPUESTA MEDIO (más indica mejor desempeño)

dftr_tra_a = dftr_tra_a.copy()
dftr_tra_a.loc[:, 'tot_tr_medio'] = dftr_tra_a.total / dftr_tra_a.tr_medio

dftr_cul_a = dftr_cul_a.copy()
dftr_cul_a.loc[:, 'tot_tr_medio'] = dftr_cul_a.total / dftr_cul_a.tr_medio

dftr_edu_a = dftr_edu_a.copy()
dftr_edu_a.loc[:, 'tot_tr_medio'] = dftr_edu_a.total / dftr_edu_a.tr_medio

dftr_sub_a = dftr_sub_a.copy()
```

```
dftr_sub_a.loc[:, 'tot_tr_medio'] = dftr_sub_a.total / dftr_sub_a.tr_medio
```

[347]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_tra_a.index, y=dftr_tra_a.tot_tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_cul_a.index, y=dftr_cul_a.tot_tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_edu_a.index, y=dftr_edu_a.tot_tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_sub_a.index, y=dftr_sub_a.tot_tr_medio, s=200)

ax.set_title('ENTRADAS vs TIEMPO MEDIO de RESPUESTA a lo largo del año',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Mes", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Número', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_a.index, labels=['ene', 'feb', 'mar',
                                            'abr', 'may', 'jun',
                                            'jul', 'ago', 'sep',
                                            'oct', 'nov', 'dic'])  # Solo para␣
 ↪poner las etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper left', labels=[
    'Transportes',
    'Cultura',
    'Educación',
    'Subvención-Política Social'
], fontsize=14);
```



Estacionalidad Día Semana

- El día con peor tiempo de respuesta en Transporte es el sábado y el mejor el domingo (¿?).
- En Cultura el peor tiempo de respuesta es el domingo y el mejor el sábado.
- En Educación el peor tiempo de respuesta es el miércoles y el mejor el viernes.
- En Subvenciones-Política Social el peor tiempo de respuesta es el jueves y el mejor el lunes.
- Valores atípicos muy altos el miércoles y jueves en Medioambiental.

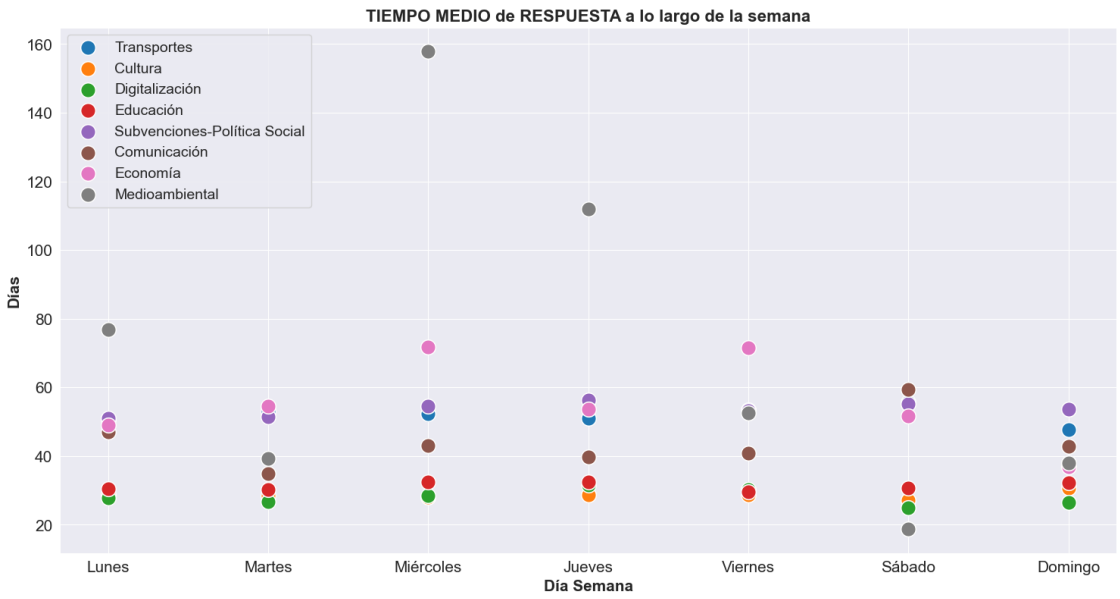
[348]:
```
dftr_nc_ds = dftr.groupby(['dia_semana', 'nueva_conse']).agg(
    tr_medio=('tiempo_respuesta', 'mean'),
    total=('nueva_conse', 'count')).reset_index()
```

[349]:
```
dftr_nc_ds = dftr_nc_ds.set_index('dia_semana')
```

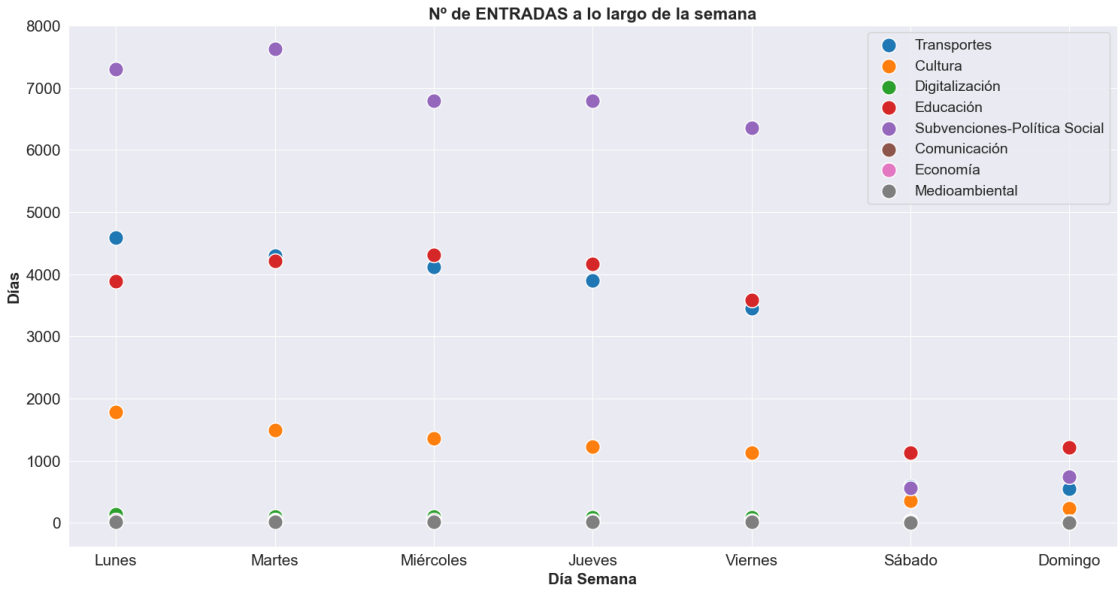
[350]:
```
# Transporte
dftr_tra_ds = dftr_nc_ds[dftr_nc_ds.nueva_conse == 'Transporte']

# Cultura
dftr_cul_ds = dftr_nc_ds[dftr_nc_ds.nueva_conse == 'Cultura']

# Digitalización
dftr_dig_ds = dftr_nc_ds[dftr_nc_ds.nueva_conse == 'Digitalizacion']

# Educación
dftr_edu_ds = dftr_nc_ds[dftr_nc_ds.nueva_conse == 'Educacion']

# Subvenciones (Polítca Social)
dftr_sub_ds = dftr_nc_ds[dftr_nc_ds.nueva_conse == 'Subvenciones']

# Comunicación
dftr_com_ds = dftr_nc_ds[dftr_nc_ds.nueva_conse == 'Comunicacion']

# Economía
dftr_eco_ds = dftr_nc_ds[dftr_nc_ds.nueva_conse == 'Economia']

# Medioambiental
dftr_med_ds = dftr_nc_ds[dftr_nc_ds.nueva_conse == 'Medioambiental']
```

[351]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_tra_ds.index, y=dftr_tra_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_cul_ds.index, y=dftr_cul_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_dig_ds.index, y=dftr_dig_ds.tr_medio, s=200)
```

```
sns.scatterplot(ax=ax, x=dftr_edu_ds.index, y=dftr_edu_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_sub_ds.index, y=dftr_sub_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_com_ds.index, y=dftr_com_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_eco_ds.index, y=dftr_eco_ds.tr_medio, s=200)
sns.scatterplot(ax=ax, x=dftr_med_ds.index, y=dftr_med_ds.tr_medio, s=200)

ax.set_title('TIEMPO MEDIO de RESPUESTA a lo largo de la semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día Semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper left', labels=[
    'Transportes',
    'Cultura',
    'Digitalización',
    'Educación',
    'Subvenciones-Política Social',
    'Comunicación',
    'Economía',
    'Medioambiental'
], fontsize=14);
```



[352]:
```
f, ax = plt.subplots(1,1, figsize=(18,9))

sns.scatterplot(ax=ax, x=dftr_tra_ds.index, y=dftr_tra_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_cul_ds.index, y=dftr_cul_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_dig_ds.index, y=dftr_dig_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_edu_ds.index, y=dftr_edu_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_sub_ds.index, y=dftr_sub_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_com_ds.index, y=dftr_com_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_eco_ds.index, y=dftr_eco_ds.total, s=200)
sns.scatterplot(ax=ax, x=dftr_med_ds.index, y=dftr_med_ds.total, s=200)

ax.set_title('Nº de ENTRADAS a lo largo de la semana',
             fontsize=16, loc='center', fontdict=dict(weight='bold'))

ax.set_xlabel("Día Semana", fontsize=15, fontdict=dict(weight='bold'))
ax.set_ylabel('Días', fontsize=15, fontdict=dict(weight='bold'))

ax.set_xticks(ticks=dftr_q_ds.index, labels=etiquetas)  # Solo para poner las␣
 ↪etiquetas al eje x.

ax.tick_params(axis='x', which='major', labelsize=15, labelrotation=0)
ax.tick_params(axis='y', which='major', labelsize=15)

ax.legend(loc='upper right', labels=[
    'Transportes',
    'Cultura',
    'Digitalización',
    'Educación',
    'Subvenciones-Política Social',
    'Comunicación',
    'Economía',
    'Medioambiental'
], fontsize=14);
```



6 FIN AL ANÁLISIS