Analisis de tasa de Mortalidad en argentina(1990-2021)

Autor: Pedro Burgos

Datos: https://datos.gob.ar/dataset/salud-tasa-mortalidad-infantil

Herramientas:

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

In [2]:
```
ds =  pd.read_csv("tasa-mortalidad-infantil-deis-1990-2021.csv")
```

In [3]:
```
ds.head(5)
#Veamos que contiene el dataset.
```

Out[3]:

	indice_tiempo	mortalidad_infantil_argentina	mortalidad_infantil_caba	mortalidad_in
0	1990-01-01	25.6	16.8	
1	1991-01-01	24.7	15.2	
2	1992-01-01	23.9	14.9	
3	1993-01-01	22.9	14.6	
4	1994-01-01	22.0	14.3	

5 rows × 26 columns

In [4]:
```
ds.tail(5)
```

Out[4]:

	indice_tiempo	mortalidad_infantil_argentina	mortalidad_infantil_caba	mortalidad_i
27	2017-01-01	9.3	6.9	
28	2018-01-01	8.8	6.0	
29	2019-01-01	9.2	7.3	
30	2020-01-01	8.4	4.9	
31	2021-01-01	8.0	4.6	

5 rows × 26 columns

In [5]:
```
#El dataset solo al parecer contiene un tipo de indice de tiempo y la can
#De cada provincia + caba entre los años 1990-2021 siempre evaluando deso
#para poder usar la columna de tiempo cambie el nombre por comodidad.
```

```
ds=ds.rename(columns={"indice_tiempo":"year"})
```

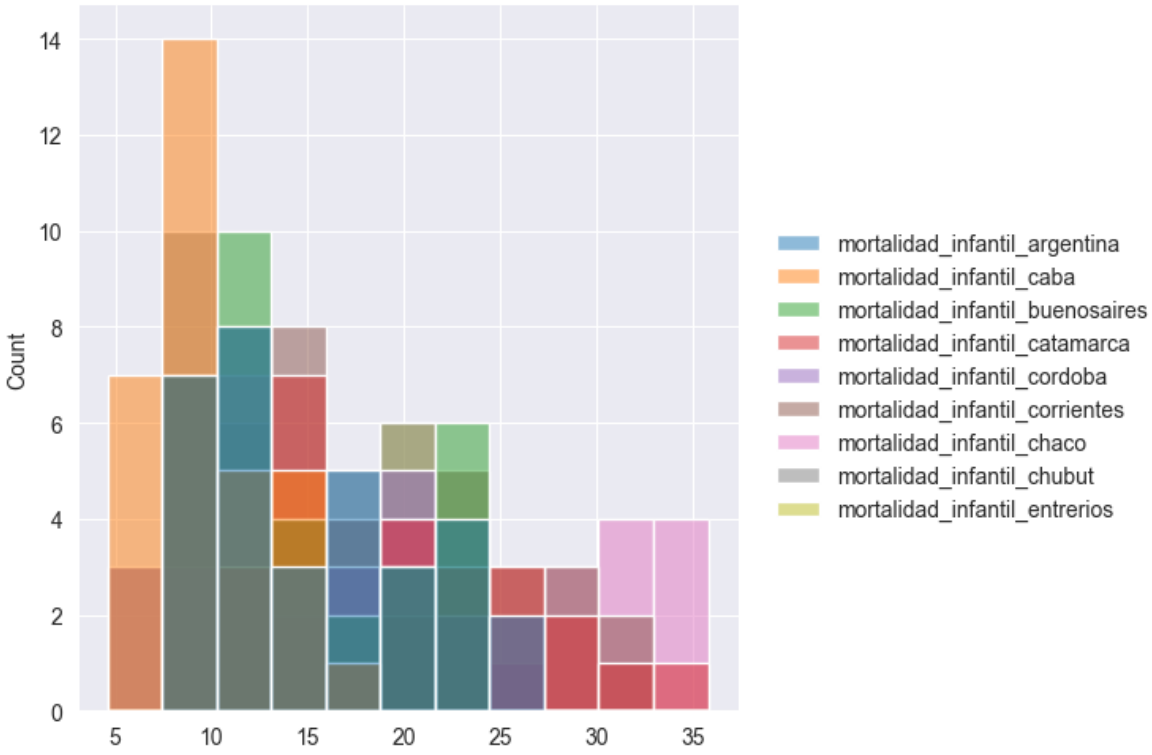
In [6]:
```
ds.info()
#reviso si hay nulos en la columnas y no hay segun non-null ,con esta inf
#puedo dar por sentado el filtrado o limpieza , ademas que con la natural
#de lo mismo datos no puedo realizar tantas operaciones sino de comparaci
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 26 columns):
 #   Column                                 Non-Null Count  Dtype  
---  ------                                 --------------  -----  
 0   year                                   32 non-null     object 
 1   mortalidad_infantil_argentina          32 non-null     float64
 2   mortalidad_infantil_caba               32 non-null     float64
 3   mortalidad_infantil_buenosaires        32 non-null     float64
 4   mortalidad_infantil_catamarca          32 non-null     float64
 5   mortalidad_infantil_cordoba            32 non-null     float64
 6   mortalidad_infantil_corrientes         32 non-null     float64
 7   mortalidad_infantil_chaco              32 non-null     float64
 8   mortalidad_infantil_chubut             32 non-null     float64
 9   mortalidad_infantil_entrerios          32 non-null     float64
 10  mortalidad_infantil_formosa            32 non-null     float64
 11  mortalidad_infantil_jujuy              32 non-null     float64
 12  mortalidad_infantil_lapampa            32 non-null     float64
 13  mortalidad_infantil_larioja            32 non-null     float64
 14  mortalidad_infantil_mendoza            32 non-null     float64
 15  mortalidad_infantil_misiones           32 non-null     float64
 16  mortalidad_infantil_neuquen            32 non-null     float64
 17  mortalidad_infantil_rionegro           32 non-null     float64
 18  mortalidad_infantil_salta              32 non-null     float64
 19  mortalidad_infantil_sanjuan            32 non-null     float64
 20  mortalidad_infantil_sanluis            32 non-null     float64
 21  mortalidad_infantil_santacruz          32 non-null     float64
 22  mortalidad_infantil_santafe            32 non-null     float64
 23  mortalidad_infantil_santiagodelestero  32 non-null     float64
 24  mortalidad_infantil_tucuman            32 non-null     float64
 25  mortalidad_infantil_tierradelfuego     32 non-null     float64
dtypes: float64(25), object(1)
memory usage: 6.6+ KB
```

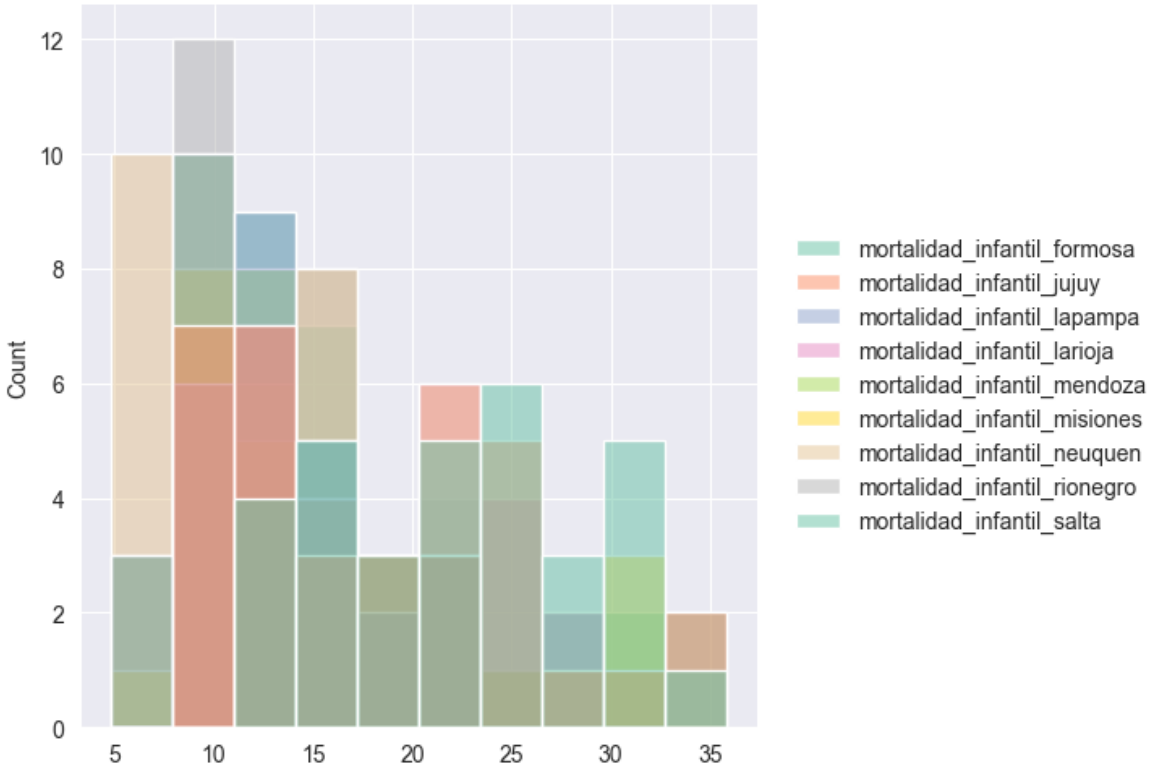
In [7]:
```
# al ver que year es tipo objeto ,para clarificar a python lo paso aque e
#de tiempo
ds['year'] = pd.to_datetime(ds['year'])
```

In [8]:
```
#Para hacer comparaciones al dataset lo fragmento de distinto dataframe
#asi realizar comparaciones visuales mucho mas comoda ,volviendo a las co
#dataset original una lista de ellas.
ds_a_e= ds[list(ds.columns)[:10]]
ds_d_s= ds[list(ds.columns)[10:19]]
ds_s_t=ds[list(ds.columns)[19:]]
```

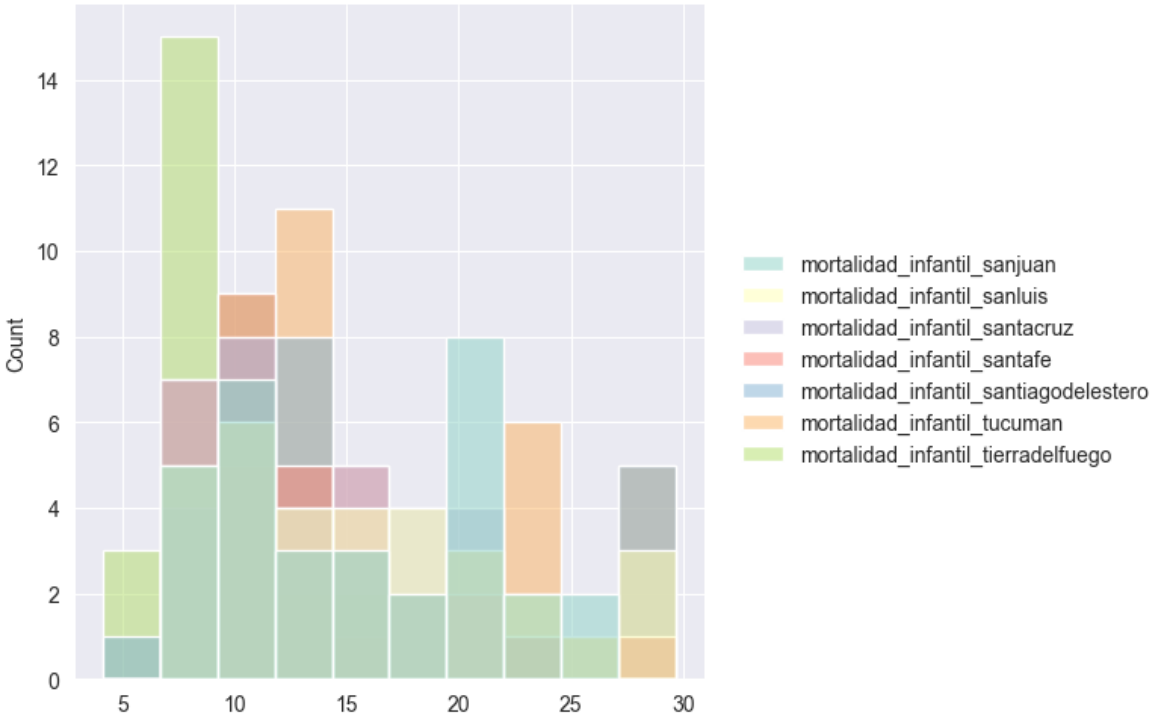
In [9]:
```
sns.set_style('darkgrid')
sns.displot(data=ds_a_e);
#los graficos a explorar nos dan a enteder la cantidad maxima de mortalid
#por provincia y cuanta veces pasa
#cada grafico tiene su propio tema de colores
```



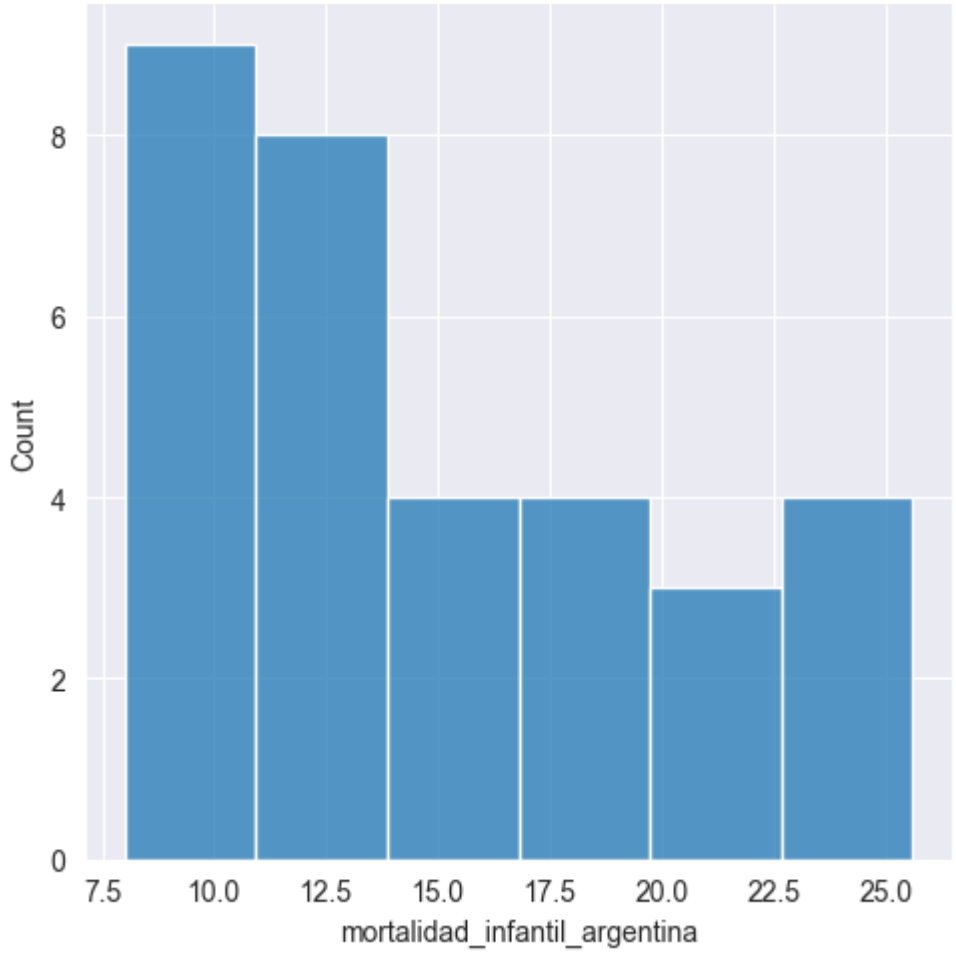
In [10]:
```
sns.displot(data=ds_d_s ,palette="Set2");
```



In [11]:
```
sns.displot(data=ds_s_t, palette="Set3");
```

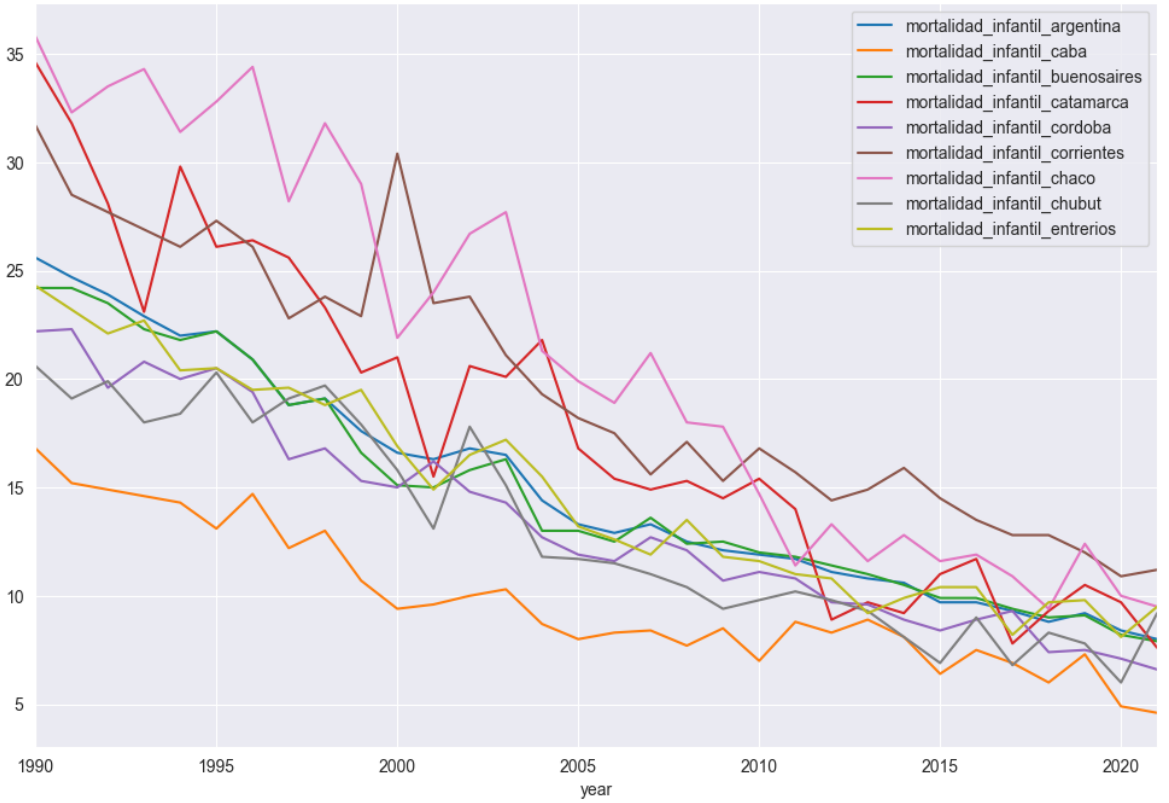


In [12]:
```
sns.displot(data=ds , x = "mortalidad_infantil_argentina");
#para tener una vista de argentina en general nos asegura que normalmente
#mayormente aparecia 7.5 sobre la cantidad de habitantes de mortalidad in
```

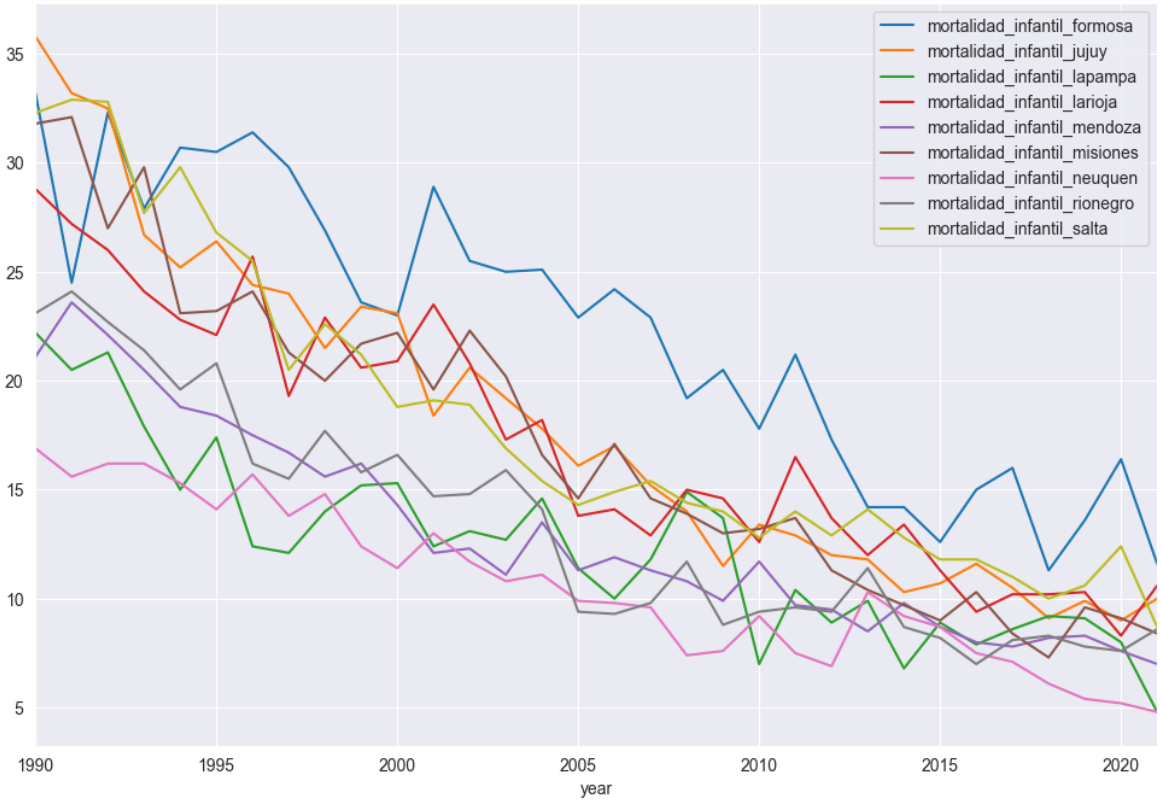


In [13]:
```
#Aprovechando que tengo una columna de tiempo hare graficos para ver como
#como era la mortalidad_infantil en cada provincia
```

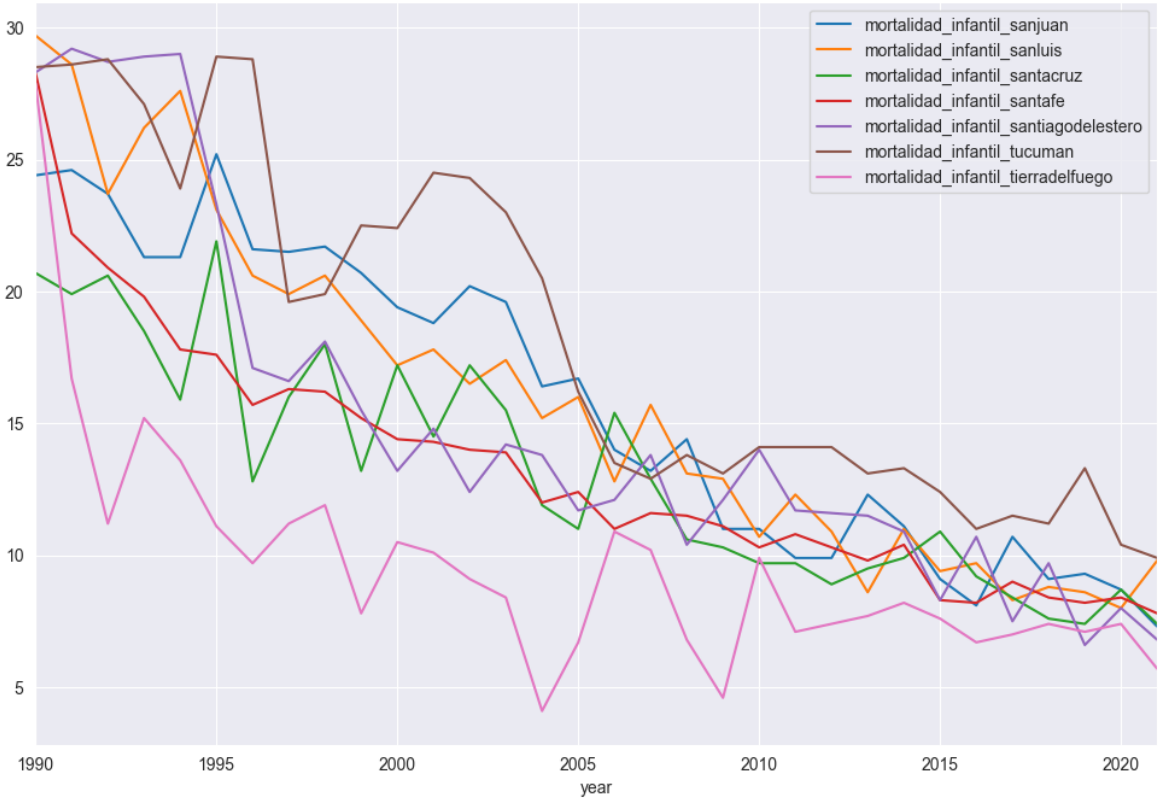
```
# Tramar gráfico de líneas con todas las columnas de valores
fig, ax = plt.subplots(figsize=(12, 8))  # Tamaño de 12x8 pulgadas

# Tramar gráfico de líneas con todas las columnas de valores
ds_a_e.set_index('year', inplace=True)
ds_a_e.plot(kind='line', ax=ax)

plt.show()
```



In [14]:
```
# Tramar gráfico de líneas con todas las columnas de valores
fig, ax = plt.subplots(figsize=(12, 8))  # Tamaño de 12x8 pulgadas

# Tramar gráfico de líneas con todas las columnas de valores
ds_d_s = ds_d_s.join(ds["year"])

ds_d_s.set_index("year", inplace=True)
ds_d_s.plot(kind='line', ax=ax)

plt.show()
```



In [15]:
```
# Tramar gráfico de líneas con todas las columnas de valores
fig, ax = plt.subplots(figsize=(12, 8))  # Tamaño de 12x8 pulgadas

# Tramar gráfico de líneas con todas las columnas de valores
ds_s_t = ds_s_t.join(ds["year"])

ds_s_t.set_index("year", inplace=True)
ds_s_t.plot(kind='line', ax=ax)

plt.show()
```



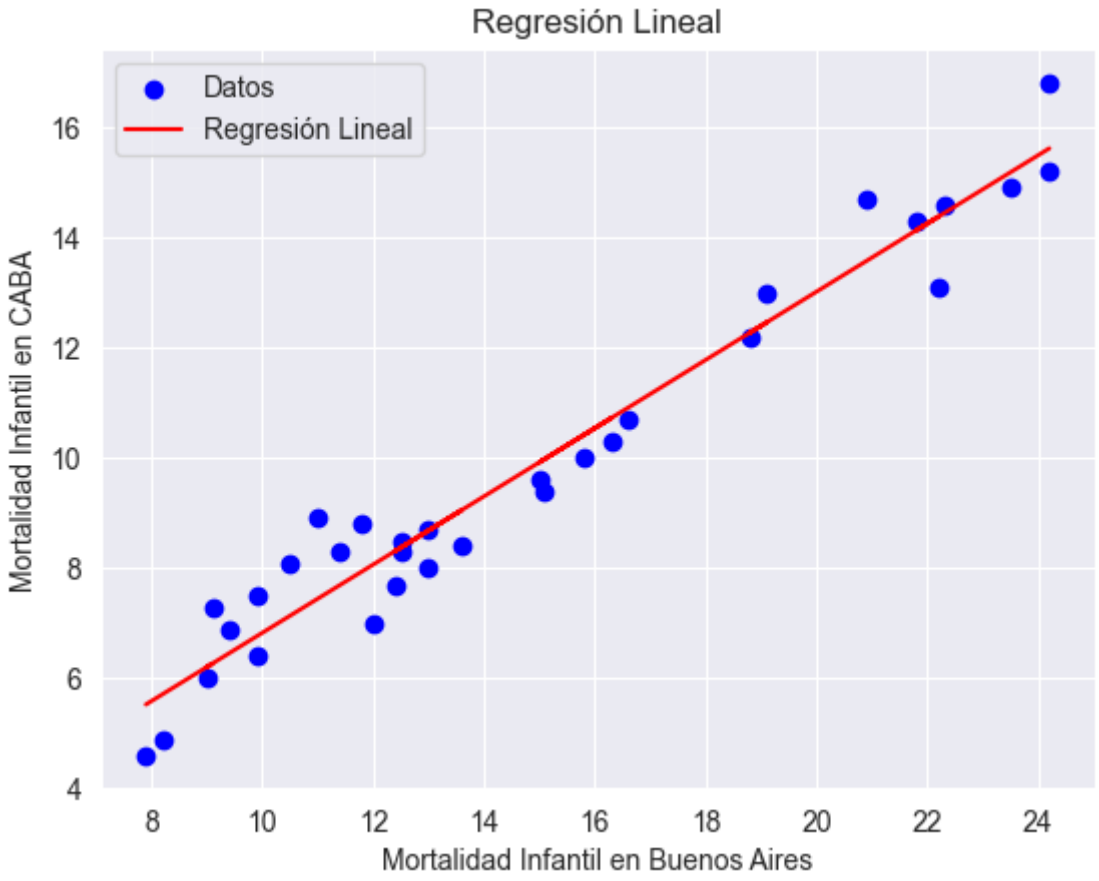
In [16]:
```
#Este es el grafico donde hago la comparacion general de todas.

fig, ax = plt.subplots(figsize=(15,15))  # Tamaño de 12x8 pulgadas

# Tramar gráfico de líneas con todas las columnas de valores
ds.plot(kind='line', ax=ax , x = "year")


#Conclusion positiva que puedo alegar a esto es que el pueblo argentino a
#mejorando aunque sea de poco en el cuidado infantil segun el contexto de
#igual seria bueno tener informacion sobre como eran esos casos de mortal
#sin mencionar que hay que revisar que entre 2020 y 2021 se legalizo el a
#si eso influye en calificacion de mortalidad infantil.
```

Out[16]: <Axes: xlabel='year'>

In [17]:
```
#Para finalizar con este analisis simple quisiera hacer un modelo de regr
#Donde quiero predecir la mortalidad infantil de caba usando la de buenos
#siendo tan cercanas debe haber una fuerte relacion supondre.
regression_model = LinearRegression()

#x variable predictora
#y variable a predecir
```

```python
X = ds['mortalidad_infantil_buenosaires'].values.reshape(-1, 1)  # Asegur
y = ds['mortalidad_infantil_caba'].values
regression_model = LinearRegression()
regression_model.fit(X, y)
coeficiente_angular = regression_model.coef_[0]
intercepto = regression_model.intercept_

print("Coeficiente Angular:", coeficiente_angular)
print("Intercepto:", intercepto)

# Crear una figura y un eje
fig, ax = plt.subplots()

# Graficar los puntos de datos
ax.scatter(X, y, color='blue', label='Datos')

# Calcular los valores predichos utilizando el modelo de regresión lineal
y_pred = regression_model.predict(X)

# Graficar la línea de regresión
ax.plot(X, y_pred, color='red', label='Regresión Lineal')

# Etiquetas de los ejes
ax.set_xlabel('Mortalidad Infantil en Buenos Aires')
ax.set_ylabel('Mortalidad Infantil en CABA')

# Título del gráfico
ax.set_title('Regresión Lineal')

# Leyenda
ax.legend()

# Mostrar el gráfico
plt.show()


#segun el coeficiente angular:
#Esto significa que, en promedio,
#por cada unidad adicional en la variable predictora,
#se espera que la variable objetivo aumente en 0.6187 unidades.
#Indica la pendiente de la línea de regresión.
#Si la variable predictora aumenta en una unidad,
#la variable objetivo aumentará en el coeficiente angular
#osea relacion muy fuerte


#Intercepto:
#Cuando la variable predictora es cero,
#se espera que la variable objetivo tenga un valor de aproximadamente 0.6
#son muy parecidad demostrando una fuerte relacion.
```

```
Coeficiente Angular: 0.6187128137787657
Intercepto: 0.6409596988756778
```


Regresión Lineal
Datos
Regresión Lineal
Mortalidad Infantil en CABA
Mortalidad Infantil en Buenos Aires
4
6
8
10
12
14
16
8
10
12
14
16
18
20
22
24

In [ ]: