Aprendizaje Automatizado

Practica: Explorar Datos de Empleados de las dependencias de la UANL

Nombre Cynthia Selene Martínez Espinoza **Matricula** 1011238

Carga de Datos

In [2]:
```
#Importar librerias
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import colors
from scipy.stats import f_oneway, kruskal, normaltest, ttest_ind
from statsmodels.stats.multicomp import pairwise_tukeyhsd
```

In [3]:
```
# Leer los datos de archivo csv, typed_uanl.csv con el URL
url = "https://raw.githubusercontent.com/ppGodel/data_mining/main/csv/typed_uanl.cs
df = pd.read_csv(url)
```

Identificar Datos

Identificar las datos del Archivo typed_uanl.csv :

- **Fecha:** Fecha del dato 'Fecha'
- **Nombre:** Nombre del empleado 'Nombre'
- **Tipo:** Tipo de Empleado 'Tipo'
- **Sueldo Neto:** Monto del Sueldo del empleado 'Sueldo Neto'
- **Dependencia:** Dependencia de la UANL 'dependencia'

Descriptivas Datos

In [13]:
```
# Identificar los nombres de las columnas
print("Columnas :\n", df.columns)
```

```
Columnas :
 Index(['Nombre', 'Sueldo Neto', 'dependencia', 'Fecha', 'Tipo'], dtype='object')
```

In [14]:
```
# Obtener estadísticas descriptivas de cada entidad
EstadisticasDescriptivas = df.describe()
print(f"Estadísticas descriptivas:\n{EstadisticasDescriptivas}\n")
```

```
Estadísticas descriptivas:
         Sueldo Neto
count  636201.000000
mean    14241.682401
std      9578.442311
min       175.410000
25%      8007.660000
50%     11426.500000
75%     17654.630000
max    147051.590000
```

Hallazgos

Hallazgos (Datos Agrupados por fecha) :

- Al conocer la desviacion estandar por fecha se identifican con mayor desviacion los meses de Junio y diciembre a partir del 2020
- La media de los meses de Junio y diciembre a partir del 2020 son muy elevadas, por lo que robustece lo encontrado en el punto anterior
- La media mas baja se observa en el mes de Junio 2020 Dependencia de la UANL
- No se cuenta con informacion de los meses de Diciembre 2021 y Enero 2022
- Solo contamos con el mes de Enero el 2024 (Agrupado_anio)
- Los valores maximos por año los tiene el 2023, en base el crecimiento natural(Agrupado_anio)

Explorando Datos

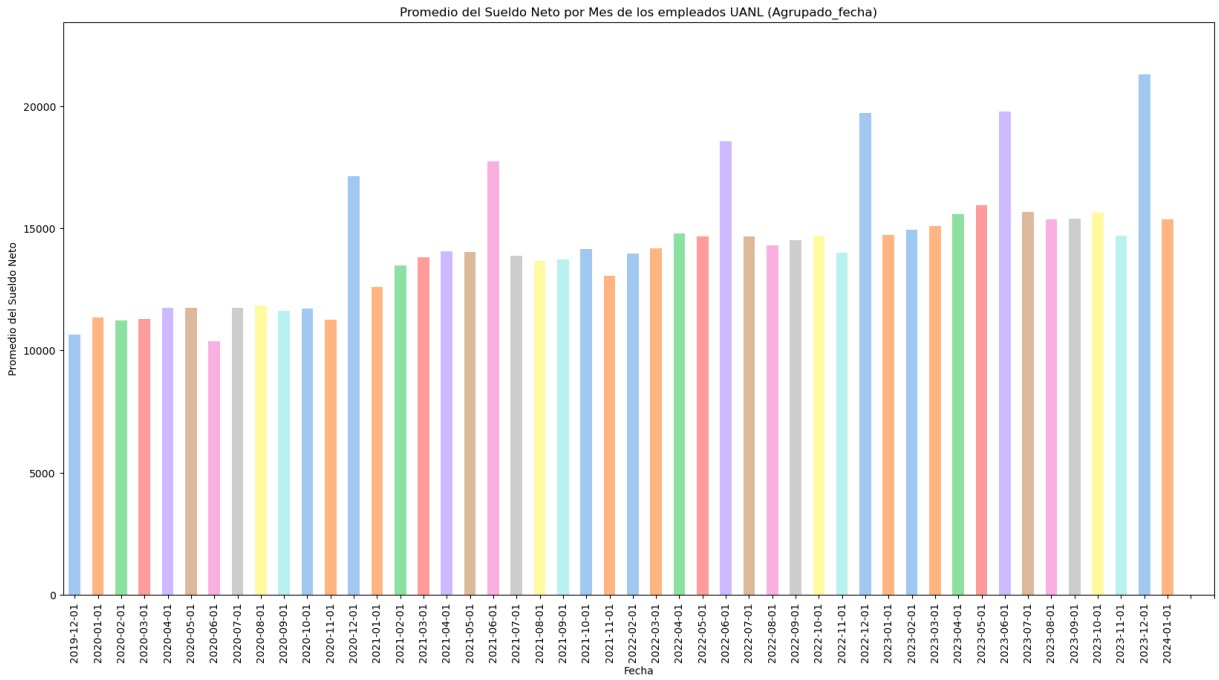
In [18]:
```
# No se localizan valores nullos en la informacion
print(df.isnull().sum())
```

```
Nombre         0
Sueldo Neto    0
dependencia    0
Fecha          0
Tipo           0
dtype: int64
```

In [73]:
```
# Obtengo Medidas descriptivas agrupando por fecha
Agrupado_fecha = df.groupby('Fecha')['Sueldo Neto'].agg(['min', 'max', 'mean', 'std
```

```
#print(Agrupado_fecha)
```

In [38]:
```
# Definir colores  para las barras
colors = sns.color_palette('pastel', 12)
# Gráfica de barras de las medias del Sueldo Neto por mes
plt.figure(figsize=(20, 10))
Agrupado_fecha['mean'].plot( kind='bar', color=colors)
plt.title(f'Promedio del Sueldo Neto por Mes de los empleados UANL (Agrupado_fecha)
plt.ylabel('Promedio del Sueldo Neto')
plt.xticks(ticks=range(50))
plt.ylim(0, Agrupado_fecha['mean'].max() * 1.1)  # Ajuste del rango del eje y
plt.savefig("C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Grafica
plt.show()
plt.close()
```



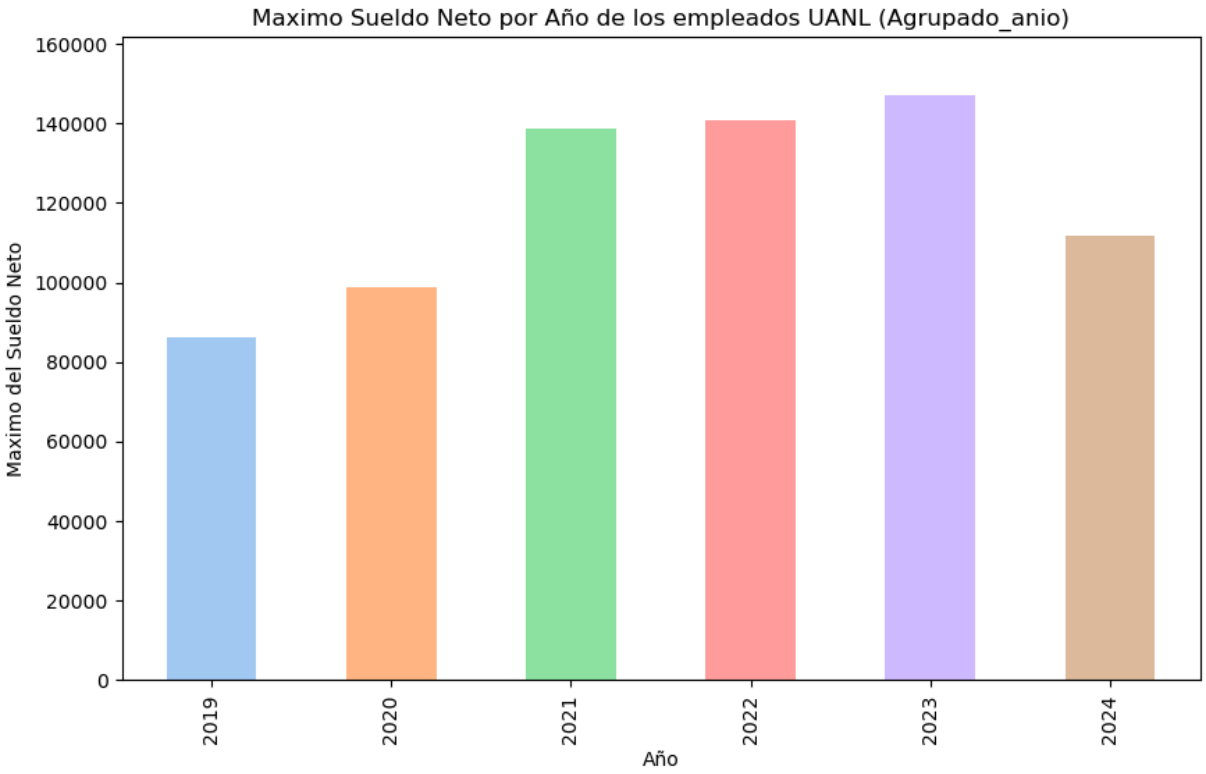
In [18]:
```
# Convertir la columna 'Fecha' a datetime
df['Fecha'] = pd.to_datetime(df['Fecha'])
```

In [19]:
```
# Agrupar por año y calcular estadísticas descriptivas
df['Año'] = df['Fecha'].dt.year
Agrupado_anio = df.groupby('Año')['Sueldo Neto'].agg(['min', 'max', 'mean', 'std'])
print(Agrupado_anio)
```

```
          min        max          mean           std
Año
2019   294.44   86245.29  10640.372254   7764.839265
2020   175.41   98881.96  11949.890176   8794.587944
2021   187.54  138744.51  14014.582055   9296.192065
2022   189.65  140739.88  15291.581540   9669.770671
2023   230.03  147051.59  16209.321490  10075.871084
2024   825.27  111868.46  15371.374204   9796.259695
```

In [20]:
```
# Definir colores  para las barras
colors = sns.color_palette('pastel',6)
# Gráfica de barras de las medias del Sueldo Neto por mes
```

```
plt.figure(figsize=(10, 6))
Agrupado_anio['max'].plot( kind='bar', color=colors)
plt.title(f'Maximo Sueldo Neto por Año de los empleados UANL (Agrupado_anio)')
plt.ylabel('Maximo del Sueldo Neto')
plt.xticks(ticks=range(6))
plt.ylim(0, Agrupado_fecha['max'].max() * 1.1)  # Ajuste del rango del eje y
plt.savefig("C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Grafica
plt.show()
plt.close()
```



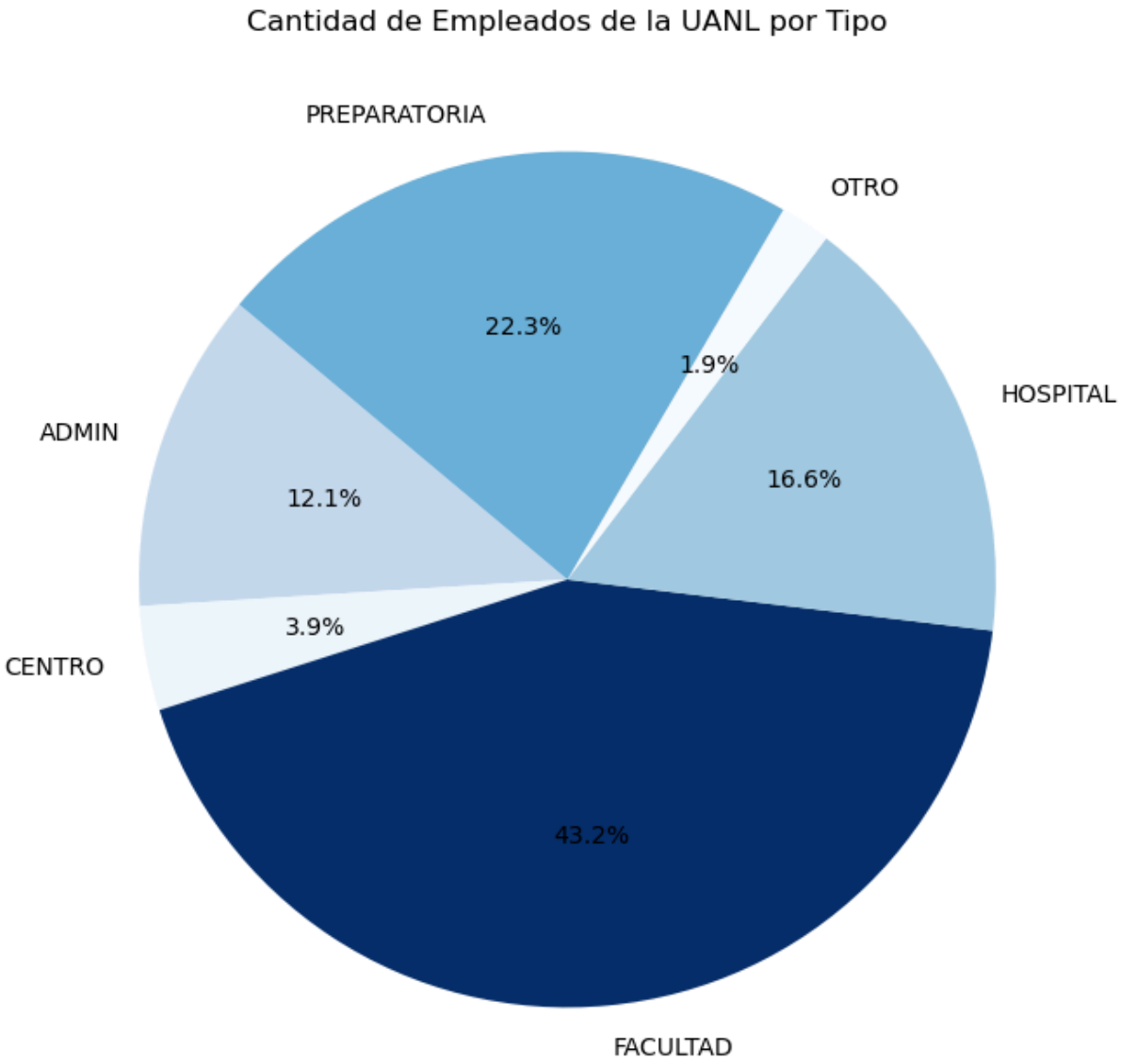
In [21]:
```
# Agrupar por tipo y calcular la media y suma del Sueldo Neto
agrupado_tipo = df.groupby(df['Tipo'])['Nombre'].agg(['count'])
print(agrupado_tipo)
```

```
              count
Tipo
ADMIN          76994
CENTRO         25117
FACULTAD      274527
HOSPITAL      105549
OTRO           12190
PREPARATORIA  141824
```

In [24]:
```
# Normalizar los valores
normdata = colors.Normalize(vmin= agrupado_tipo['count'].min(), vmax= agrupado_tipo
colormap = plt.cm.get_cmap("Blues")
#colormap =plt.cm.viridis
colores =colormap(normdata(agrupado_tipo['count']))

# Crear la gráfica de pastel
plt.figure(figsize=(8, 8))  # Tamaño de la figura
plt.pie(agrupado_tipo['count'], labels=agrupado_tipo.index, autopct='%1.1f%%', star
```

```
# Añadir título
plt.title('Cantidad de Empleados de la UANL por Tipo')

# Guardar la gráfica en un archivo
plt.savefig("C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Grafica
# Mostrar la gráfica
plt.show()
plt.close()
```

```
C:\Users\PC\AppData\Local\Temp\ipykernel_18356\2824550479.py:3: MatplotlibDeprecatio
nWarning: The get_cmap function was deprecated in Matplotlib 3.7 and will be removed
two minor releases later. Use ``matplotlib.colormaps[name]`` or ``matplotlib.colorma
ps.get_cmap(obj)`` instead.
  colormap = plt.cm.get_cmap("Blues")
```



Analizando el Sueldo neto

Hallazgos (max_sueldo) :

- Se encuentra un sueldo levado en el mes de Junio 23, se encuentra en dos dependencias el empleado recibiendo un sueldo de 230,517.48 en cada una (VICTOR EDUARDO SANCHEZ PLASCENCIA)
- Junio 2021, 2022, se tienen una media elevada, vemos que el sueldo del rector esos meses esta arriba de la media
- La media mas baja se observa en el mes de Junio 2020 Dependencia de la UANL

Consultas

In [25]:
```
#  Obtener los sueldos netos por fecha , empleado
#   Sumar sueldos por fecha y empleado
suma_sueldo = df.groupby(['Fecha', 'Nombre'])['Sueldo Neto'].sum().reset_index()
print("Suma de sueldos por fecha y empleado:")
print(suma_sueldo)
```

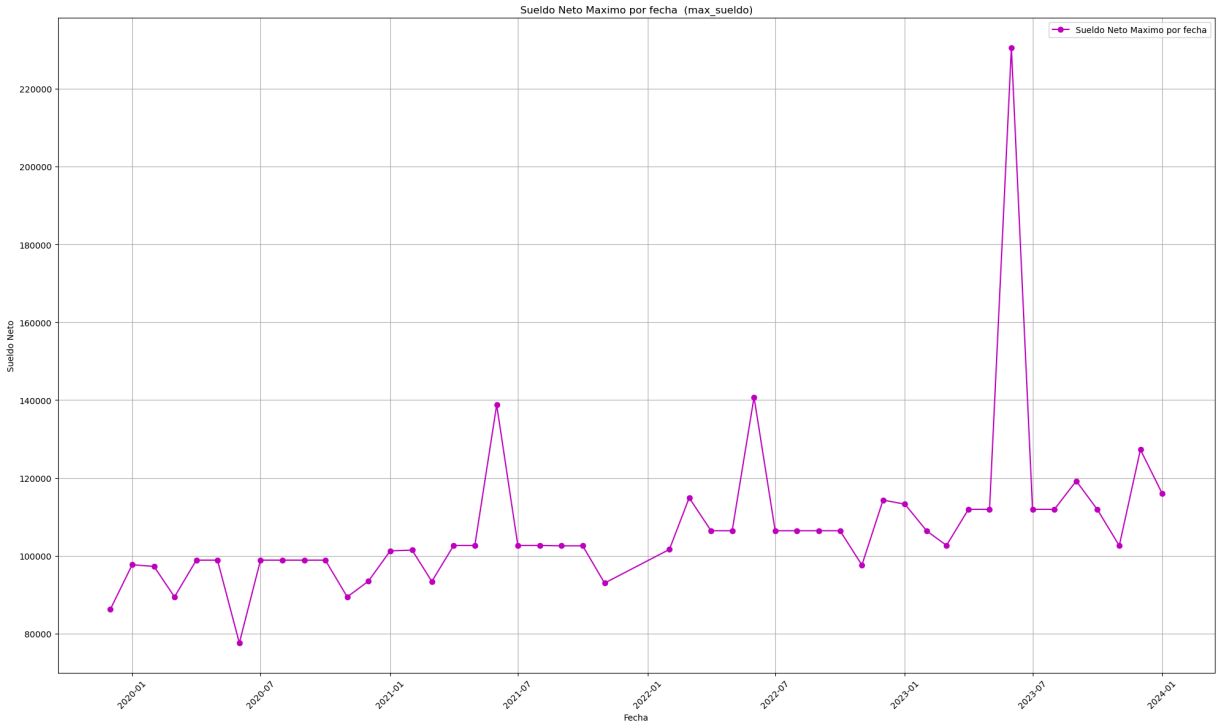
```
Suma de sueldos por fecha y empleado:
            Fecha                                Nombre  Sueldo Neto
0      2019-12-01        AARON ABDIEL TREJO VILLALOBOS      3518.39
1      2019-12-01                   AARON CRUZ VARGAS      4835.71
2      2019-12-01        AARON EDGARDO ESPINOZA GARCIA       714.28
3      2019-12-01          AARON GERALDO SOTO ESPINOZA      5561.06
4      2019-12-01   AARON GILBERTO VILLARREAL ELIZONDO     36523.66
...           ...                                  ...          ...
632473 2024-01-01              ZULLY BAZALDUA GUERRERO      9211.97
632474 2024-01-01                  ZULMA ESPINOZA MATA      5342.94
632475 2024-01-01        ZULY MARLENE COVARRUBIAS MATA      7554.58
632476 2024-01-01          ZURYA VANESSA LLAMAS MEDINA     11928.65
632477 2024-01-01  ZUZANKA ALEJANDRA VILLARREAL ARIZPE     23273.49

[632478 rows x 3 columns]
```

In [27]:
```
# Encontrar el sueldo neto maximo por fecha sin distinguir dependencia
# Agrupar por fecha y nombre, sueldo neto maximo
max_sueldo = suma_sueldo.loc[suma_sueldo.groupby('Fecha')['Sueldo Neto'].idxmax()].
print(max_sueldo )
```

	Fecha	Nombre	Sueldo Neto
0	2019-12-01	REYES S. TAMEZ GUERRA	86245.29
1	2020-01-01	REYES S. TAMEZ GUERRA	97698.38
2	2020-02-01	REYES S. TAMEZ GUERRA	97274.55
3	2020-03-01	REYES S. TAMEZ GUERRA	89433.90
4	2020-04-01	REYES S. TAMEZ GUERRA	98881.96
5	2020-05-01	REYES S. TAMEZ GUERRA	98881.96
6	2020-06-01	REYES S. TAMEZ GUERRA	77608.82
7	2020-07-01	REYES S. TAMEZ GUERRA	98881.96
8	2020-08-01	REYES S. TAMEZ GUERRA	98881.96
9	2020-09-01	REYES S. TAMEZ GUERRA	98881.96
10	2020-10-01	REYES S. TAMEZ GUERRA	98881.96
11	2020-11-01	REYES S. TAMEZ GUERRA	89433.90
12	2020-12-01	JESUS ANCER RODRIGUEZ	93467.50
13	2021-01-01	REYES S. TAMEZ GUERRA	101260.38
14	2021-02-01	REYES S. TAMEZ GUERRA	101463.43
15	2021-03-01	REYES S. TAMEZ GUERRA	93409.31
16	2021-04-01	REYES S. TAMEZ GUERRA	102661.88
17	2021-05-01	REYES S. TAMEZ GUERRA	102661.88
18	2021-06-01	REYES S. TAMEZ GUERRA	138744.51
19	2021-07-01	REYES S. TAMEZ GUERRA	102661.88
20	2021-08-01	REYES S. TAMEZ GUERRA	102661.88
21	2021-09-01	REYES S. TAMEZ GUERRA	102559.88
22	2021-10-01	REYES S. TAMEZ GUERRA	102559.88
23	2021-11-01	REYES S. TAMEZ GUERRA	92987.31
24	2022-02-01	REYES S. TAMEZ GUERRA	101647.77
25	2022-03-01	HECTOR LUIS AGUILAR GONZALEZ	114902.42
26	2022-04-01	SANTOS GUZMAN LOPEZ	106441.03
27	2022-05-01	SANTOS GUZMAN LOPEZ	106441.03
28	2022-06-01	REYES S. TAMEZ GUERRA	140739.88
29	2022-07-01	SANTOS GUZMAN LOPEZ	106441.03
30	2022-08-01	SANTOS GUZMAN LOPEZ	106441.03
31	2022-09-01	SANTOS GUZMAN LOPEZ	106441.03
32	2022-10-01	SANTOS GUZMAN LOPEZ	106441.03
33	2022-11-01	HECTOR LUIS AGUILAR GONZALEZ	97609.16
34	2022-12-01	HECTOR LUIS AGUILAR GONZALEZ	114321.30
35	2023-01-01	ARNOLDO TELLEZ LOPEZ	113252.14
36	2023-02-01	SANTOS GUZMAN LOPEZ	106374.47
37	2023-03-01	SANTOS GUZMAN LOPEZ	102664.94
38	2023-04-01	SANTOS GUZMAN LOPEZ	111936.46
39	2023-05-01	SANTOS GUZMAN LOPEZ	111936.46
40	2023-06-01	VICTOR EDUARDO SANCHEZ PLASCENCIA	230517.48
41	2023-07-01	SANTOS GUZMAN LOPEZ	111936.46
42	2023-08-01	SANTOS GUZMAN LOPEZ	111936.46
43	2023-09-01	LINDA ANGELICA OSORIO CASTILLO	119217.34
44	2023-10-01	SANTOS GUZMAN LOPEZ	111868.46
45	2023-11-01	SANTOS GUZMAN LOPEZ	102596.94
46	2023-12-01	HECTOR LUIS AGUILAR GONZALEZ	127263.08
47	2024-01-01	HECTOR LUIS AGUILAR GONZALEZ	115961.52

In [28]:
```
# Graficar los fechas con los sueltos netos maximos
plt.figure(figsize=(20, 12))
plt.plot(max_sueldo['Fecha'], max_sueldo['Sueldo Neto'], marker='o', linestyle='-',

plt.xlabel('Fecha')
plt.ylabel('Sueldo Neto')
```

```
plt.title('Sueldo Neto Maximo por fecha  (max_sueldo)')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()

# Guardar la gráfica en un archivo
plt.savefig("C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Grafica
# Mostrar la gráfica
plt.show()
plt.close()
```



In [30]:
```
df_filtromayorsueldo = df[df['Nombre'].str.contains('VICTOR EDUARDO SANCHEZ PLASCEN
df_filtromayorsueldo.head(50)
```

Out[30]:

	Nombre	Sueldo Neto	dependencia	Fecha	Tipo	Año
20	VICTOR EDUARDO SANCHEZ PLASCENCIA	72171.38	RECTORIA	2019-12-01	ADMIN	2019
14430	VICTOR EDUARDO SANCHEZ PLASCENCIA	79086.70	RECTORIA	2020-01-01	ADMIN	2020
27131	VICTOR EDUARDO SANCHEZ PLASCENCIA	79141.75	RECTORIA	2020-02-01	ADMIN	2020
40996	VICTOR EDUARDO SANCHEZ PLASCENCIA	74360.51	RECTORIA	2020-03-01	ADMIN	2020
55206	VICTOR EDUARDO SANCHEZ PLASCENCIA	79998.30	RECTORIA	2020-04-01	ADMIN	2020
69515	VICTOR EDUARDO SANCHEZ PLASCENCIA	79998.30	RECTORIA	2020-05-01	ADMIN	2020
83685	VICTOR EDUARDO SANCHEZ PLASCENCIA	67107.77	RECTORIA	2020-06-01	ADMIN	2020
97938	VICTOR EDUARDO SANCHEZ PLASCENCIA	79998.30	RECTORIA	2020-07-01	ADMIN	2020
110486	VICTOR EDUARDO SANCHEZ PLASCENCIA	79998.29	RECTORIA	2020-08-01	ADMIN	2020
122779	VICTOR EDUARDO SANCHEZ PLASCENCIA	79998.29	RECTORIA	2020-09-01	ADMIN	2020
136685	VICTOR EDUARDO SANCHEZ PLASCENCIA	79998.29	RECTORIA	2020-10-01	ADMIN	2020
150694	VICTOR EDUARDO SANCHEZ PLASCENCIA	74360.51	RECTORIA	2020-11-01	ADMIN	2020
164728	VICTOR EDUARDO SANCHEZ PLASCENCIA	73154.37	RECTORIA	2020-12-01	ADMIN	2020
179615	VICTOR EDUARDO SANCHEZ PLASCENCIA	82376.72	RECTORIA	2021-01-01	ADMIN	2021

	Nombre	Sueldo Neto	dependencia	Fecha	Tipo	Año
193555	VICTOR EDUARDO SANCHEZ PLASCENCIA	83009.41	RECTORIA	2021-02-01	ADMIN	2021
207088	VICTOR EDUARDO SANCHEZ PLASCENCIA	79150.93	RECTORIA	2021-03-01	ADMIN	2021
220986	VICTOR EDUARDO SANCHEZ PLASCENCIA	84633.62	RECTORIA	2021-04-01	ADMIN	2021
234897	VICTOR EDUARDO SANCHEZ PLASCENCIA	84633.62	RECTORIA	2021-05-01	ADMIN	2021
248798	VICTOR EDUARDO SANCHEZ PLASCENCIA	106934.82	RECTORIA	2021-06-01	ADMIN	2021
262638	VICTOR EDUARDO SANCHEZ PLASCENCIA	84633.62	RECTORIA	2021-07-01	ADMIN	2021
274838	VICTOR EDUARDO SANCHEZ PLASCENCIA	84633.62	RECTORIA	2021-08-01	ADMIN	2021
288338	VICTOR EDUARDO SANCHEZ PLASCENCIA	84531.62	RECTORIA	2021-09-01	ADMIN	2021
302122	VICTOR EDUARDO SANCHEZ PLASCENCIA	79906.19	RECTORIA	2021-10-01	ADMIN	2021
315226	VICTOR EDUARDO SANCHEZ PLASCENCIA	61356.26	RECTORIA	2021-11-01	ADMIN	2021
329041	VICTOR EDUARDO SANCHEZ PLASCENCIA	83632.45	RECTORIA	2022-02-01	ADMIN	2022
342014	VICTOR EDUARDO SANCHEZ PLASCENCIA	80990.34	RECTORIA	2022-03-01	ADMIN	2022
355095	VICTOR EDUARDO SANCHEZ PLASCENCIA	87015.58	RECTORIA	2022-04-01	ADMIN	2022
368227	VICTOR EDUARDO SANCHEZ PLASCENCIA	87015.58	RECTORIA	2022-05-01	ADMIN	2022

	Nombre	Sueldo Neto	dependencia	Fecha	Tipo	Año
381345	VICTOR EDUARDO SANCHEZ PLASCENCIA	110106.14	RECTORIA	2022-06-01	ADMIN	2022
394433	VICTOR EDUARDO SANCHEZ PLASCENCIA	87015.58	RECTORIA	2022-07-01	ADMIN	2022
405894	VICTOR EDUARDO SANCHEZ PLASCENCIA	87015.58	RECTORIA	2022-08-01	ADMIN	2022
418518	VICTOR EDUARDO SANCHEZ PLASCENCIA	87015.58	RECTORIA	2022-09-01	ADMIN	2022
431510	VICTOR EDUARDO SANCHEZ PLASCENCIA	87015.58	RECTORIA	2022-10-01	ADMIN	2022
444621	VICTOR EDUARDO SANCHEZ PLASCENCIA	80990.35	RECTORIA	2022-11-01	ADMIN	2022
457727	VICTOR EDUARDO SANCHEZ PLASCENCIA	72364.56	RECTORIA	2022-12-01	ADMIN	2022
470851	VICTOR EDUARDO SANCHEZ PLASCENCIA	88364.56	RECTORIA	2023-01-01	ADMIN	2023
482353	VICTOR EDUARDO SANCHEZ PLASCENCIA	87485.23	RECTORIA	2023-02-01	ADMIN	2023
495175	VICTOR EDUARDO SANCHEZ PLASCENCIA	84987.70	RECTORIA	2023-03-01	ADMIN	2023
508180	VICTOR EDUARDO SANCHEZ PLASCENCIA	91244.58	RECTORIA	2023-04-01	ADMIN	2023
521321	VICTOR EDUARDO SANCHEZ PLASCENCIA	91244.58	RECTORIA	2023-05-01	ADMIN	2023
534446	VICTOR EDUARDO SANCHEZ PLASCENCIA	115258.74	RECTORIA	2023-06-01	ADMIN	2023
536820	VICTOR EDUARDO SANCHEZ PLASCENCIA	115258.74	ESCUELA PREPARATORIA #12	2023-06-01	PREPARATORIA	2023

	Nombre	Sueldo Neto	dependencia	Fecha	Tipo	Año
547565	VICTOR EDUARDO SANCHEZ PLASCENCIA	22708.45	RECTORIA	2023-07-01	ADMIN	2023
549690	VICTOR EDUARDO SANCHEZ PLASCENCIA	40403.13	ESCUELA PREPARATORIA #12	2023-07-01	PREPARATORIA	2023
561259	VICTOR EDUARDO SANCHEZ PLASCENCIA	47064.67	ESCUELA PREPARATORIA #12	2023-08-01	PREPARATORIA	2023

Trabajadores en distintas dependencias

In [127…
```
# Encontrar trabajadores en distintas dependencias
# Agrupar por nombre empleado,  para obtener solo empleados  y el conteo de las vec
empleados_varias_dependencias = df.groupby('Nombre')['dependencia'].nunique()
```

In [128…
```
# Filtrar empleados que trabajan en más de una dependencia
empleados_varias_dependencias = empleados_varias_dependencias[empleados_varias_depe
```

In [129…
```
# Obtener la lista de empleados que trabajan en varias dependencias
empleados_varias_dependencias_lista = empleados_varias_dependencias.index.tolist()
print("Empleados que trabajan en varias dependencias:", empleados_varias_dependenci
```

In [130…
```
# Se unen los datos para presentar la informacion de la depenencias en las cuales e
# Hacer la union con (inner join) por la entidad por 'nombre' que es la llave
df_uniondatos = pd.merge(df, empleados_varias_dependencias, on='Nombre', how='inner
```

In [131…
```
# Para conocer como quedaron los datos de la union
# Muestro los datos que se unieron, para explorar la informacion
# df_uniondatos.head(5)
# consulto las columnas con las que se dio de alta el dataframe
print("Columnas :\n", df_uniondatos.columns)
```

```
Columnas :
 Index(['Nombre', 'Sueldo Neto', 'dependencia_x', 'Fecha', 'Tipo',
       'dependencia_y'],
      dtype='object')
```

In [132…
```
# De la union de datos UANL con Empleados en mas de una dependencia, se dejan los e
# quitar datos distintos
df_EmpleadosDepedenciaMas = df_uniondatos[['Nombre', 'Tipo', 'dependencia_x', 'depe
#df_EmpleadosDepedenciaMas.head(100)
```

In [151…
```
df_filtroempleado = df_EmpleadosDepedenciaMas[df_EmpleadosDepedenciaMas['Nombre'].s
print(df_filtroempleado)
```

```
                                Nombre          Tipo  \
674  VICTOR EDUARDO SANCHEZ PLASCENCIA         ADMIN
715  VICTOR EDUARDO SANCHEZ PLASCENCIA  PREPARATORIA

               dependencia_x  dependencia_y
674                 RECTORIA              2
715  ESCUELA PREPARATORIA #12              2
```

In [152…
```
df_filtroempleado = df_EmpleadosDepedenciaMas[df_EmpleadosDepedenciaMas['Nombre'].s
print(df_filtroempleado)
```

```
                           Nombre          Tipo  \
44252  ELTON DAMIAN AGUILA GARCIA         ADMIN
44262  ELTON DAMIAN AGUILA GARCIA  PREPARATORIA

                                  dependencia_x  dependencia_y
44252  DIRECCION DEL PROGRAMA DE FUTBOL AMERICANO              2
44262          ESC.IND.Y PREPA.TEC.ALVARO OBREGON              2
```

Graficas Barras fecha / dependencia

In [159…
```
# Convertir la columna 'Fecha' a datetime
df['Fecha'] = pd.to_datetime(df['Fecha'])
```

In [31]:
```
import os
# Crear una carpeta principal para las gráficas
if not os.path.exists('GraficasBarras_Fecha_Dependencia'):
    os.makedirs('C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Gra
```

In [32]:
```
import matplotlib.pyplot as plt

# Agrupar por fecha y dependencia y calcular estadísticas descriptivas
Agrupado_fecha_dependencia = df.groupby(['Fecha', 'dependencia'])['Sueldo Neto'].ag
#print(Agrupado_fecha_dependencia)
```

In [180…
```
# Graficar las estadísticas descriptivas para cada combinación de fecha y dependenc
for _, row in Agrupado_fecha_dependencia.iterrows():
    fecha = row['Fecha']
    dependencia = row['dependencia']
    data_fecha_dependencia = df[(df['Fecha'] == fecha) & (df['dependencia'] == depe

    plt.figure(figsize=(10, 6))

    # Gráfica de barras del sueldo neto por fecha y dependencia
    plt.bar(data_fecha_dependencia.index, data_fecha_dependencia, alpha=0.7, label=

      # Mostrar estadísticas en la gráfica
    textstr = '\n'.join((
        f'Min: {row["min"]:.2f}',
        f'Max: {row["max"]:.2f}',
        f'Mean: {row["mean"]:.2f}',
        f'Std: {row["std"]:.2f}'))

    # Posición del texto
```

```
    plt.gca().text(0.95, 0.95, textstr, transform=plt.gca().transAxes, fontsize=10,
                   verticalalignment='top', horizontalalignment='right',
                   bbox=dict(facecolor='white', alpha=0.5))

    # Mostrar leyenda y gráfico
    plt.legend()

    # Crear una subcarpeta para la fecha
    fecha_str = fecha.strftime('%Y-%m-%d')
    subfolder_path = os.path.join('C:/Users/PC/Documents/GitHub/GitFlow-en-Github/M
    if not os.path.exists(subfolder_path):
        os.makedirs(subfolder_path)

    # Guardar la gráfica en un archivo
    filename = f'{subfolder_path}/{dependencia}.png'
    plt.savefig(filename)
    plt.close()
```

Graficas histogramas por dependencia

In [35]:
```
import os
# Crear una carpeta principal para las gráficas
if not os.path.exists('graficasHistograma_dependencia'):
    os.makedirs('C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Gra
```

In [34]:
```
# Agrupar por dependencia y calcular estadísticas descriptivas
Agrupado_dependencia = df.groupby('dependencia')['Sueldo Neto'].agg(['min', 'max',
print(Agrupado_dependencia)
```

```
                                                      min       max  \
dependencia
AUDITORIA INTERNA DE LA U.A.N.L.                   2694.80  68323.61
C. INNOVACION; INVEST. Y DESLLO. DE INGENIERIA ...  5045.30  35590.67
C.DE ESTUDIOS HUMANISTICOS                         2895.11  43505.89
C.DE INV.Y DES.DE ED.BILINGUE                      2903.41  77130.15
CAPILLA ALFONSINA BIBLIOTECA UNIVERSITARIA         1535.48  78959.29
...                                                    ...       ...
TEATRO UNIVERSITARIO                               2611.44  67532.31
TESORERIA GENERAL                                   245.38  87384.51
UNIDAD DE TRANSPARENCIA                             285.73  49809.89
WORLD TRADE CENTER MONTEREY-UANL                   7184.08  58697.17
´CAPILLA ALFONSINA´ BIBLIOTECA UNIVERSITARIA        311.22  35182.17

                                                           mean           std
dependencia
AUDITORIA INTERNA DE LA U.A.N.L.                   12993.491939   6173.853285
C. INNOVACION; INVEST. Y DESLLO. DE INGENIERIA ...  12454.441083   5296.206710
C.DE ESTUDIOS HUMANISTICOS                         15591.806762   5655.643685
C.DE INV.Y DES.DE ED.BILINGUE                      14503.070670   8174.759578
CAPILLA ALFONSINA BIBLIOTECA UNIVERSITARIA         11789.009435   6349.997558
...                                                         ...           ...
TEATRO UNIVERSITARIO                               10873.276872   8876.623893
TESORERIA GENERAL                                  16224.493451  10929.214817
UNIDAD DE TRANSPARENCIA                            15935.450939   8058.354377
WORLD TRADE CENTER MONTEREY-UANL                   15711.140227  11753.817679
´CAPILLA ALFONSINA´ BIBLIOTECA UNIVERSITARIA       13694.670512   4459.250247

[152 rows x 4 columns]
```

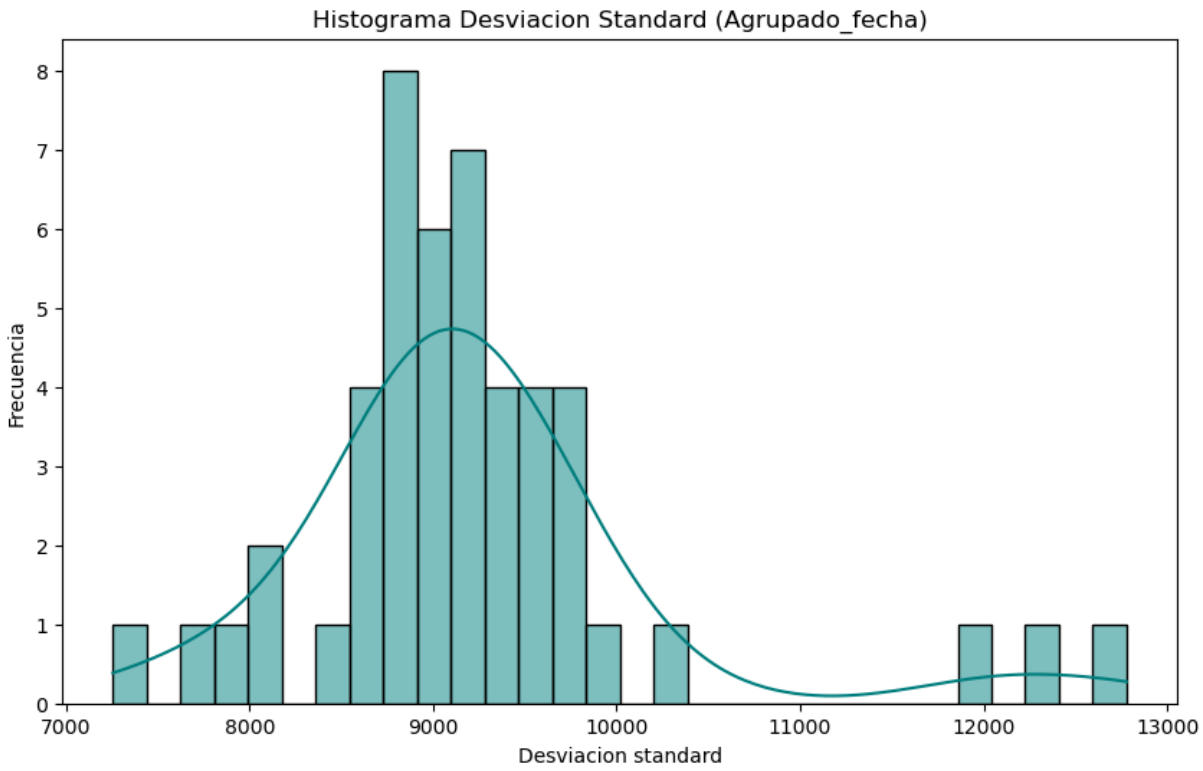
In [36]:
```python
# Graficar las estadísticas descriptivas para cada dependencia
for dependencia in Agrupado_dependencia.index:
    data_dependencia = df[df['dependencia'] == dependencia]['Sueldo Neto']

    plt.figure(figsize=(10, 6))

    # Histograma del sueldo neto por dependencia
    plt.hist(data_dependencia, bins=10, alpha=0.7, label='Histograma')

    # Agregar título y etiquetas
    plt.title(f'Estadísticas descriptivas para {dependencia}')
    plt.xlabel('Sueldo Neto')
    plt.ylabel('Frecuencia')

    # Mostrar estadísticas en la gráfica
    stats = Agrupado_dependencia.loc[dependencia]
    textstr = '\n'.join((
        f'Min: {stats["min"]:.2f}',
        f'Max: {stats["max"]:.2f}',
        f'Mean: {stats["mean"]:.2f}',
        f'Std: {stats["std"]:.2f}'))

    # Posición del texto
    plt.gca().text(0.95, 0.95, textstr, transform=plt.gca().transAxes, fontsize=10,
                   verticalalignment='top', horizontalalignment='right',
                   bbox=dict(facecolor='white', alpha=0.5))
```

```
    # Guardar la gráfica en un archivo
    plt.savefig(f'C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Gr

   # Mostrar la gráfica
    plt.legend()
   # plt.show()
    plt.close()
```

Histogramas UANL

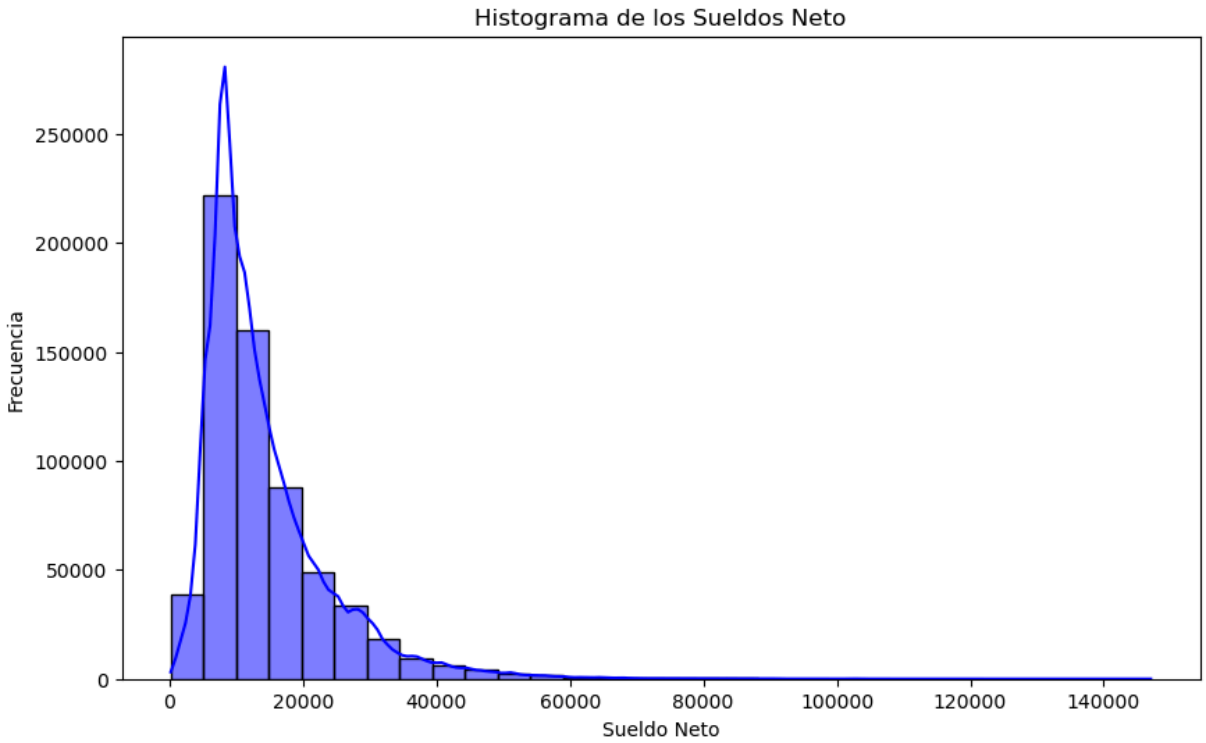
In [74]:
```
# Histograma de los std   de UANL
plt.figure(figsize=(10, 6))
sns.histplot(Agrupado_fecha['std'], bins=30,kde=True,  color='teal')
plt.title(f'Histograma Desviacion Standard (Agrupado_fecha)')
plt.xlabel('Desviacion standard')
plt.ylabel('Frecuencia')

# Guardar la gráfica en un archivo
plt.savefig("C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Grafica
# Mostrar la gráfica
plt.show()
plt.close()
```

```
C:\Users\PC\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use
_inf_as_na option is deprecated and will be removed in a future version. Convert inf
values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



In [12]:
```
# Histograma de los sueldo netos
plt.figure(figsize=(10, 6))
sns.histplot(df['Sueldo Neto'], bins=30,kde=True,  color='blue')
plt.title(f'Histograma de los Sueldos Neto (UANL)')
plt.xlabel('Sueldo Neto')
plt.ylabel('Frecuencia')

# Guardar la gráfica en un archivo
plt.savefig("C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Grafica
# Mostrar la gráfica
plt.show()
plt.close()
```

```
C:\Users\PC\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use
_inf_as_na option is deprecated and will be removed in a future version. Convert inf
values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



Graficas de XPLOT

In [ ]:
```
import os
# Crear una carpeta principal para las gráficas
if not os.path.exists('GraficasLinea_Dependencia'):
    os.makedirs('C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Gra
# Convertir la columna 'Fecha' a datetime
df['Fecha'] = pd.to_datetime(df['Fecha'])
```

In [69]:
```
# Definir la función create_boxplot_by_year
def Crea_boxplot_anio(df):

    # Extraer el año de la columna 'Fecha'
```

```
    df['Año'] = df['Fecha'].dt.year

    # Crear un boxplot para cada año
    for year in sorted(df['Año'].unique()):
        plt.figure(figsize=(10, 6))

        # Filtrar el DataFrame por el año actual
        df_year = df[df['Año'] == year]

        # Crear el boxplot usando seaborn
        sns.boxplot(x='dependencia', y='Sueldo Neto', data=df_year)

        # Agregar título y etiquetas
        plt.title(f'Boxplot de Sueldos por Dependencia en {year}')
        plt.xlabel('Dependencia')
        plt.ylabel('Sueldo Neto')

        # Guardar la gráfica en un archivo
        filename = f'C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas
        plt.savefig(filename)
        plt.close()
```

In [64]:
```
# Llamar a la función para crear los boxplots
Crea_boxplot_anio(df)
```

Graficas Linea por Dependencia

In [4]:
```
import os
# Crear una carpeta principal para las gráficas
if not os.path.exists('GraficasLinea_Dependencia'):
    os.makedirs('C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Gra
```

In [9]:
```
import matplotlib.pyplot as plt

# Agrupar por fecha y dependencia y calcular el sueldo neto máximo
max_sueldo_por_fecha_dependencia = df.groupby(['Fecha', 'dependencia'])['Sueldo Net
print(max_sueldo_por_fecha_dependencia)

# Obtener la lista de dependencias
dependencias = df['dependencia'].unique()

# Graficar el sueldo neto máximo por fecha para cada dependencia
for dependencia in dependencias:
    data_dependencia = max_sueldo_por_fecha_dependencia[max_sueldo_por_fecha_depend

    plt.figure(figsize=(10, 6))

    # Gráfica de líneas del sueldo neto máximo por fecha para la dependencia
    plt.plot(data_dependencia['Fecha'], data_dependencia['Sueldo Neto'], marker='o'

    # Agregar título y etiquetas
    plt.title(f'Comportamiento del Sueldo Neto Máximo para {dependencia} por Fecha'
    plt.xlabel('Fecha')
    plt.ylabel('Sueldo Neto Máximo')
```

```
    # Mostrar leyenda y gráfico
    plt.legend()

     # Guardar la gráfica en un archivo
    plt.savefig(f'C:/Users/PC/Documents/GitHub/GitFlow-en-Github/ML003/Practicas/Gr

    plt.close()
```

```
          Fecha                                        dependencia  \
0     2019-12-01                  AUDITORIA INTERNA DE LA U.A.N.L.
1     2019-12-01  C. INNOVACION; INVEST. Y DESLLO. DE INGENIERIA...
2     2019-12-01                        C.DE ESTUDIOS HUMANISTICOS
3     2019-12-01                     C.DE INV.Y DES.DE ED.BILINGUE
4     2019-12-01        CAPILLA ALFONSINA BIBLIOTECA UNIVERSITARIA
...          ...                                               ...
6119  2024-01-01  SRIA. DE INVESTIGACIÓN CIENTÍFICA Y DESARROLLO...
6120  2024-01-01                                TEATRO UNIVERSITARIO
6121  2024-01-01                                   TESORERIA GENERAL
6122  2024-01-01                               UNIDAD DE TRANSPARENCIA
6123  2024-01-01       ´CAPILLA ALFONSINA´ BIBLIOTECA UNIVERSITARIA

      Sueldo Neto
0        36971.50
1        20327.44
2        13720.68
3        48355.60
4        17748.66
...           ...
6119     20946.49
6120     53238.55
6121     56646.69
6122     41028.56
6123     20770.13

[6124 rows x 3 columns]
```

ANOVA

In [7]:
```
# ANOVA o Prueba no paramétrica
# Comprobando la normalidad de los Saldos Netos por fecha
normality_tests = df.groupby('Fecha')['Sueldo Neto'].apply(lambda x: normaltest(x).
print("Normality test p-values:\n", normality_tests)
```

```
Normality test p-values:
 Fecha
2019-12-01    0.0
2020-01-01    0.0
2020-02-01    0.0
2020-03-01    0.0
2020-04-01    0.0
2020-05-01    0.0
2020-06-01    0.0
2020-07-01    0.0
2020-08-01    0.0
2020-09-01    0.0
2020-10-01    0.0
2020-11-01    0.0
2020-12-01    0.0
2021-01-01    0.0
2021-02-01    0.0
2021-03-01    0.0
2021-04-01    0.0
2021-05-01    0.0
2021-06-01    0.0
2021-07-01    0.0
2021-08-01    0.0
2021-09-01    0.0
2021-10-01    0.0
2021-11-01    0.0
2022-02-01    0.0
2022-03-01    0.0
2022-04-01    0.0
2022-05-01    0.0
2022-06-01    0.0
2022-07-01    0.0
2022-08-01    0.0
2022-09-01    0.0
2022-10-01    0.0
2022-11-01    0.0
2022-12-01    0.0
2023-01-01    0.0
2023-02-01    0.0
2023-03-01    0.0
2023-04-01    0.0
2023-05-01    0.0
2023-06-01    0.0
2023-07-01    0.0
2023-08-01    0.0
2023-09-01    0.0
2023-10-01    0.0
2023-11-01    0.0
2023-12-01    0.0
2024-01-01    0.0
Name: Sueldo Neto, dtype: float64
```

In [68]:
```
# ANOVA o Prueba no paramétrica
# Comprobando la normalidad de los Saldos Netos por tipo de empleado
normality_tests = df.groupby('Tipo')['Sueldo Neto'].apply(lambda x: normaltest(x).p
print("Normality test p-values:\n", normality_tests)
```

```
Normality test p-values:
 Tipo
ADMIN           0.0
CENTRO          0.0
FACULTAD        0.0
HOSPITAL        0.0
OTRO            0.0
PREPARATORIA    0.0
Name: Sueldo Neto, dtype: float64
```