



# ANÁLISIS DE INGRESOS PARA UNA EMPRESA DE CONSUMO MASIVO - FEBRERO 2024

Este análisis tiene como propósito proporcionar una visión integral de la demografía de la empresa 🌐, abarcando sus Unidades de Negocio en diversos países. Además, permite evaluar los ingresos personas por país e identificar los riesgos asociados a cada gerencia y puesto 📁, facilitando la toma de decisiones estratégicas 🇺🇸 y la gestión eficiente del talento y los recursos 👥.

## Primero empezamos importando las librerías que vamos a usar, recordar que la base de datos está en un repositorio en PGAdmin 4

```
In [60]: import psychopg
import seaborn as sns
import pandas as pd
from datetime import date
import matplotlib.pyplot as plt
from urllib.parse import quote_plus
from sqlalchemy import create_engine
```

```
In [61]: # Contraseña con caracteres especiales
password = "Pbczeuh5@"

# Codificar la contraseña
password_encoded = quote_plus(password)

# Crear engine con contraseña codificada
engine = create_engine(f"postgresql+psychopg2://postgres:{password_encoded}@localhost")

# Ejecutar la consulta PostgreSQL para importar los datos
ACTIVOS_FEB_24 = pd.read_sql("""
    SELECT
        "periodo",
        "UNIDAD DE NEGOCIO",
        "pais",
        "RAZON SOCIAL / PLANILLA",
        "gerencia",
        "area",
        "POSICION / PUESTO / CARGO",
        "DOCUMENTO IDENTIDAD / CEDULA / RUT",
        "FECHA DE NACIMIENTO (DD/MM/YYYY)",
        "GENERO (F/M)",
        "FECHA DE INGRESO (DD/MM/YYYY)"
    FROM "DATOS_COMPILADOS_FEB24"
    WHERE "FECHA DE CESE (DD/MM/YYYY)" IS NULL;
""", engine)
```

## Segundo modificamos las fechas para que se pueda trabajar de manera más eficiente

```
In [63]: # Convertir explícitamente a datetime
ACTIVOS_FEB_24["FECHA DE INGRESO (DD/MM/YYYY)"] = pd.to_datetime(
    ACTIVOS_FEB_24["FECHA DE INGRESO (DD/MM/YYYY)"],
```

```

format="%d/%m/%Y",      # Importante, formato es día/mes/año
errors="coerce"          # valores no válidos → NaT para detectarlos
)

# Calcular edad
hoy = date.today()
ACTIVOS_FEB_24["EDAD"] = ACTIVOS_FEB_24["FECHA DE NACIMIENTO (DD/MM/YYYY)"].dt.year.
    lambda x: hoy.year - x if pd.notnull(x) else None
)

# Extraer mes
ACTIVOS_FEB_24["MES_CUMPLE"] = ACTIVOS_FEB_24["FECHA DE NACIMIENTO (DD/MM/YYYY)"].dt

# Clasificación por grupos etarios
ACTIVOS_FEB_24["RANGO_EDAD"] = pd.cut(
    ACTIVOS_FEB_24["EDAD"],
    bins=[18, 25, 35, 45, 55, 65, 100],
    labels=["18-25", "26-35", "36-45", "46-55", "56-65", "65+"],
    right=True
)

ACTIVOS_FEB_24[["EDAD", "RANGO_EDAD"]].head()

```

Out [63]:

	EDAD	RANGO_EDAD
0	30.0	26-35
1	27.0	26-35
2	47.0	46-55
3	29.0	26-35
4	36.0	36-45

In [64]:

```

# Crear nuevas columnas temporales
ACTIVOS_FEB_24["AÑO"] = ACTIVOS_FEB_24["FECHA DE INGRESO (DD/MM/YYYY)"].dt.year
ACTIVOS_FEB_24["MES_NUM"] = ACTIVOS_FEB_24["FECHA DE INGRESO (DD/MM/YYYY)"].dt.month

```

In [65]:

```

# Diccionario de meses en español
meses_es = {
    1:"Enero", 2:"Febrero", 3:"Marzo", 4:"Abril",
    5:"Mayo", 6:"Junio", 7:"Julio", 8:"Agosto",
    9:"Septiembre", 10:"Octubre", 11:"Noviembre", 12:"Diciembre"
}

ACTIVOS_FEB_24["MES"] = ACTIVOS_FEB_24["MES_NUM"].map(meses_es)

```

## Tercero comprobamos si la base de datos está en un formato correcto

In [66]:

```

ACTIVOS_FEB_24[["FECHA DE INGRESO (DD/MM/YYYY)", "AÑO", "MES", "MES_NUM"]].head()
ACTIVOS_FEB_24.dtypes

```

```
Out[66]: periodo                object
          UNIDAD DE NEGOCIO      object
          pais                   object
          RAZON SOCIAL / PLANILLA object
          gerencia               object
          area                   object
          POSICION / PUESTO / CARGO object
          DOCUMENTO IDENTIDAD / CEDULA / RUT object
          FECHA DE NACIMIENTO (DD/MM/YYYY) datetime64[ns]
          GENERO (F/M)           object
          FECHA DE INGRESO (DD/MM/YYYY) datetime64[ns]
          EDAD                   float64
          MES_CUMPLE             float64
          RANGO_EDAD            category
          AÑO                   float64
          MES_NUM               float64
          MES                   object
          dtype: object
```

```
In [67]: ACTIVOS_FEB_24 = ACTIVOS_FEB_24[ACTIVOS_FEB_24["EDAD"] >= 18]
```

```
In [68]: ACTIVOS_FEB_24.isnull().sum()
```

```
Out[68]: periodo                0
          UNIDAD DE NEGOCIO      0
          pais                   0
          RAZON SOCIAL / PLANILLA 0
          gerencia               4
          area                   284
          POSICION / PUESTO / CARGO 0
          DOCUMENTO IDENTIDAD / CEDULA / RUT 4082
          FECHA DE NACIMIENTO (DD/MM/YYYY) 0
          GENERO (F/M)           0
          FECHA DE INGRESO (DD/MM/YYYY) 4
          EDAD                   0
          MES_CUMPLE             0
          RANGO_EDAD            0
          AÑO                   4
          MES_NUM               4
          MES                   4
          dtype: int64
```

```
In [69]: ACTIVOS_FEB_24 = ACTIVOS_FEB_24.dropna()
```

```
In [70]: ACTIVOS_FEB_24.isnull().sum()
```

```
Out[70]: periodo                                0
          UNIDAD DE NEGOCIO                      0
          pais                                  0
          RAZON SOCIAL / PLANILLA                0
          gerencia                              0
          area                                  0
          POSICION / PUESTO / CARGO              0
          DOCUMENTO IDENTIDAD / CEDULA / RUT     0
          FECHA DE NACIMIENTO (DD/MM/YYYY)       0
          GENERO (F/M)                          0
          FECHA DE INGRESO (DD/MM/YYYY)          0
          EDAD                                  0
          MES_CUMPLE                            0
          RANGO_EDAD                            0
          AÑO                                    0
          MES_NUM                                0
          MES                                    0
          dtype: int64
```

```
In [71]: duplicados_por_unidad = (
          ACTIVOS_FEB_24
          .groupby(["UNIDAD DE NEGOCIO", "DOCUMENTO IDENTIDAD / CEDULA / RUT"])
          .size()
          .reset_index(name="repeticiones")
          .query("repeticiones > 1")           # filtra solo duplicados reales
          .groupby("UNIDAD DE NEGOCIO")       # vuelve a agrupar por unidad
          .size()
          .reset_index(name="total_documentos_duplicados")
        )

duplicados_por_unidad
```

```
Out[71]:
```

	UNIDAD DE NEGOCIO	total_documentos_duplicados
0	AGRO ISM	49
1	BRASIL	607
2	CARIBE	2599
3	GT	311
4	NUEVOS NEGOCIOS	32
5	PERU-CHILE	2310

```
In [72]: ACTIVOS_FEB_24 = ACTIVOS_FEB_24.drop_duplicates(
          subset=["DOCUMENTO IDENTIDAD / CEDULA / RUT"],
          keep="first"   # Se queda con el primero; "last" si quieres conservar el último
        )
```

Cuarto en este paso es cuando empezamos a realizar los análisis correspondientes

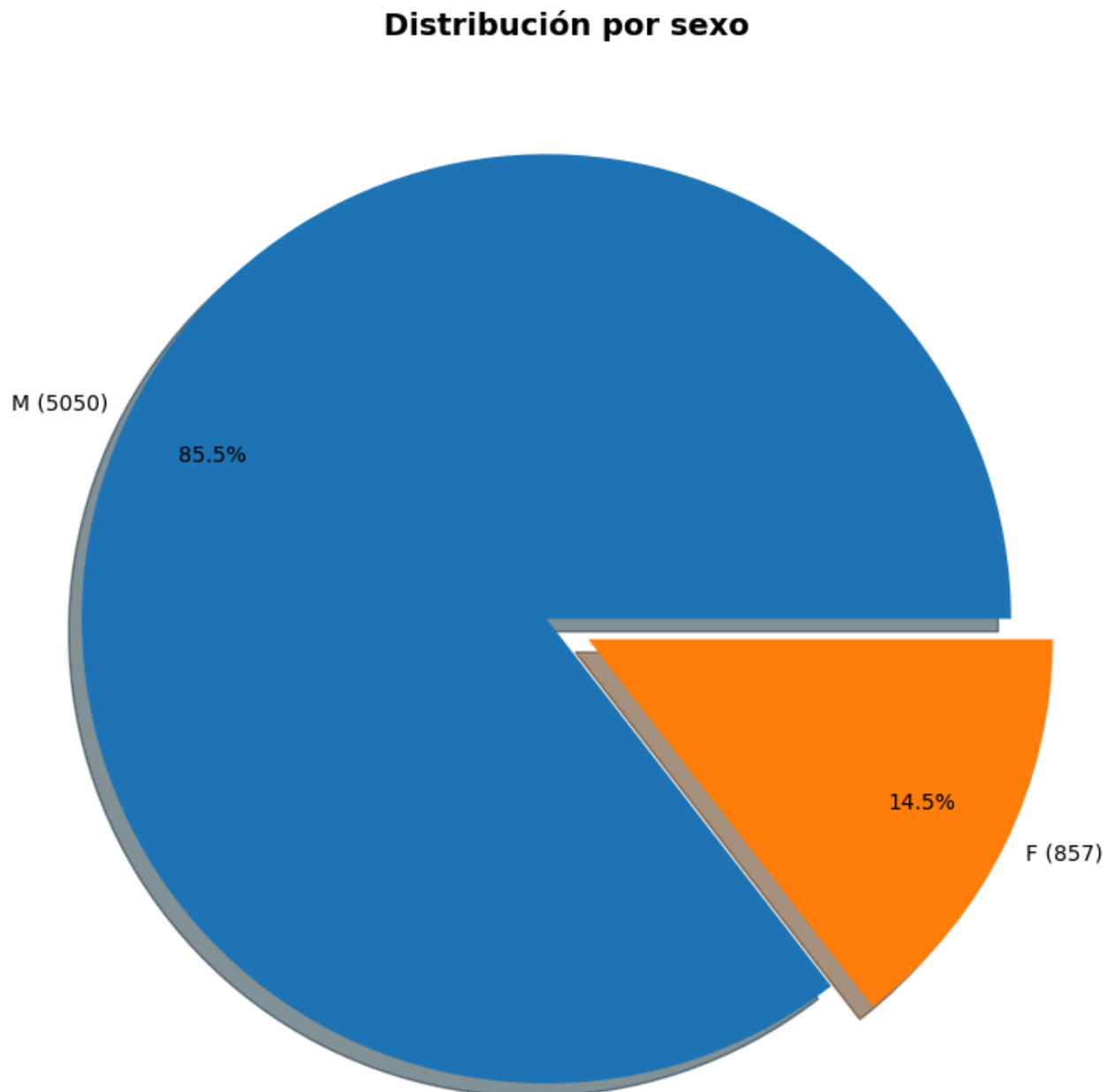
## 4.1 Análisis por género

```
In [15]: # Contar frecuencia por categoría
counts = ACTIVOS_FEB_24['GENERO (F/M)'].value_counts()

# Crear colores y explotar un poco cada segmento para mayor divisibilidad
explode = [0.05] * len(counts) # separa todas las porciones ligeramente
```

```
plt.figure(figsize=(10,8))
plt.pie(
    counts,
    labels=[f"{cat} ({counts[cat]})" for cat in counts.index], # agrega cantidad po
    autopct= lambda pct: f"{pct:.1f}%" if pct > 2 else '', # oculta porcentajes
    explode=explode,
    shadow=True,
    pctdistance=0.8, # porcentaje más cerca del borde
    labeldistance=1.05 # etiquetas externas más visibles
)

plt.title('Distribución por sexo', fontsize=14, fontweight='bold')
plt.tight_layout()
plt.show()
```



Se observa que la proporción de hombres y mujeres en la empresa es de 1 a 6, lo que indica que, al mes de febrero de 2024, la plantilla activa está mayoritariamente compuesta por mujeres. Esta información puede ser relevante para análisis de ingresos, políticas de compensación y gestión del talento.

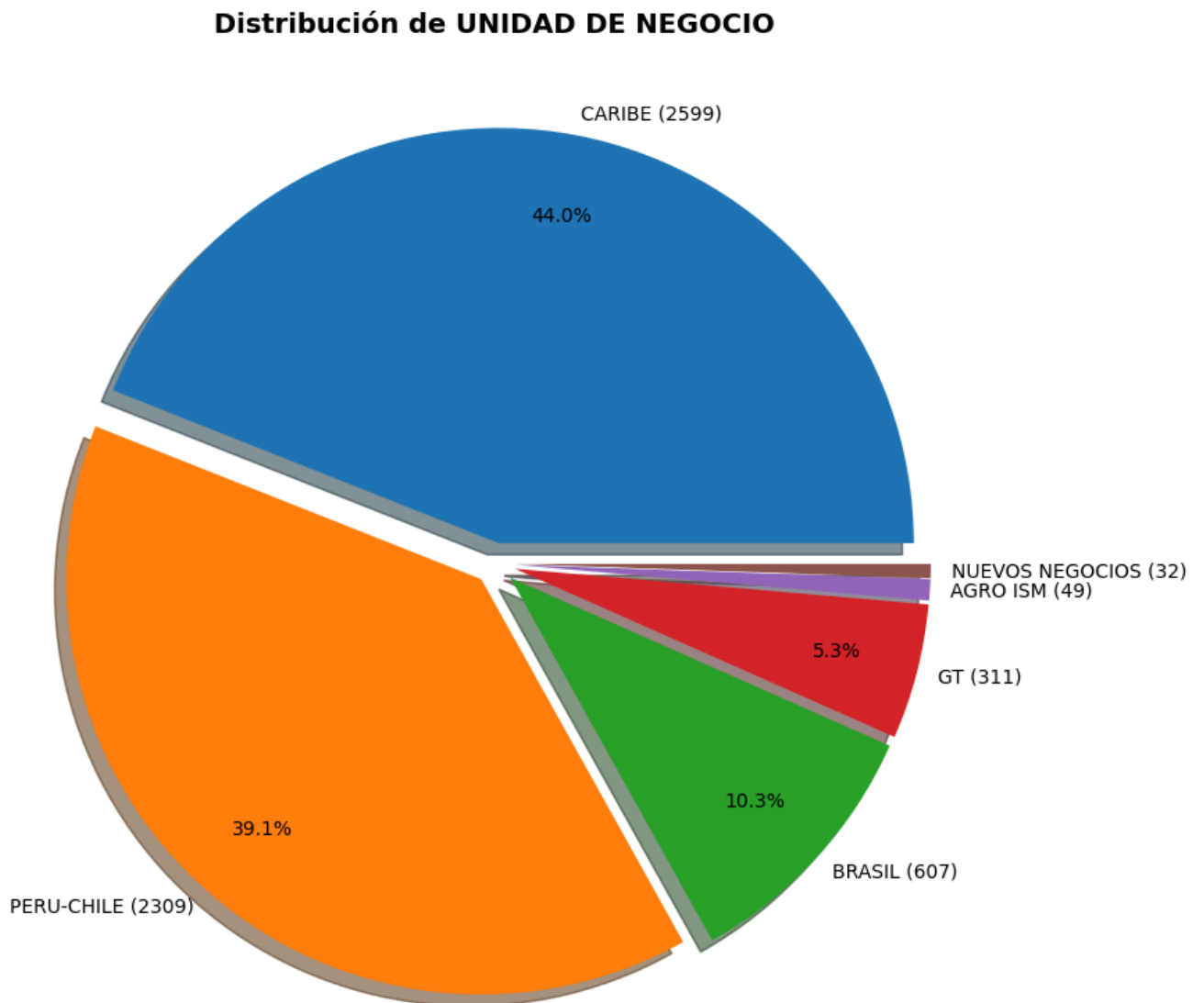
## 4.2 Análisis por Unidad de Negocio

```
In [73]: # Contar frecuencia por categoría
counts = ACTIVOS_FEB_24['UNIDAD DE NEGOCIO'].value_counts()

# Crear colores y explotar un poco cada segmento para mayor divisibilidad
explode = [0.05] * len(counts) # separa todas las porciones ligeramente

plt.figure(figsize=(10,8))
plt.pie(
    counts,
    labels=[f"{cat} ({counts[cat]})" for cat in counts.index], # agrega cantidad po
    autopct=lambda pct: f"{pct:.1f}%" if pct > 2 else '', # oculta porcentajes
    explode=explode,
    shadow=True,
    pctdistance=0.8, # porcentaje más cerca del borde
    labeldistance=1.05 # etiquetas externas más visibles
)

plt.title('Distribución de UNIDAD DE NEGOCIO', fontsize=14, fontweight='bold')
plt.tight_layout()
plt.show()
```



Se observa que en esta empresa de origen peruano se tiene registrado en Caribe la mayor cantidad de colaboradores, seguido de Perú - Chile, ambos representan el 83.1% del total de

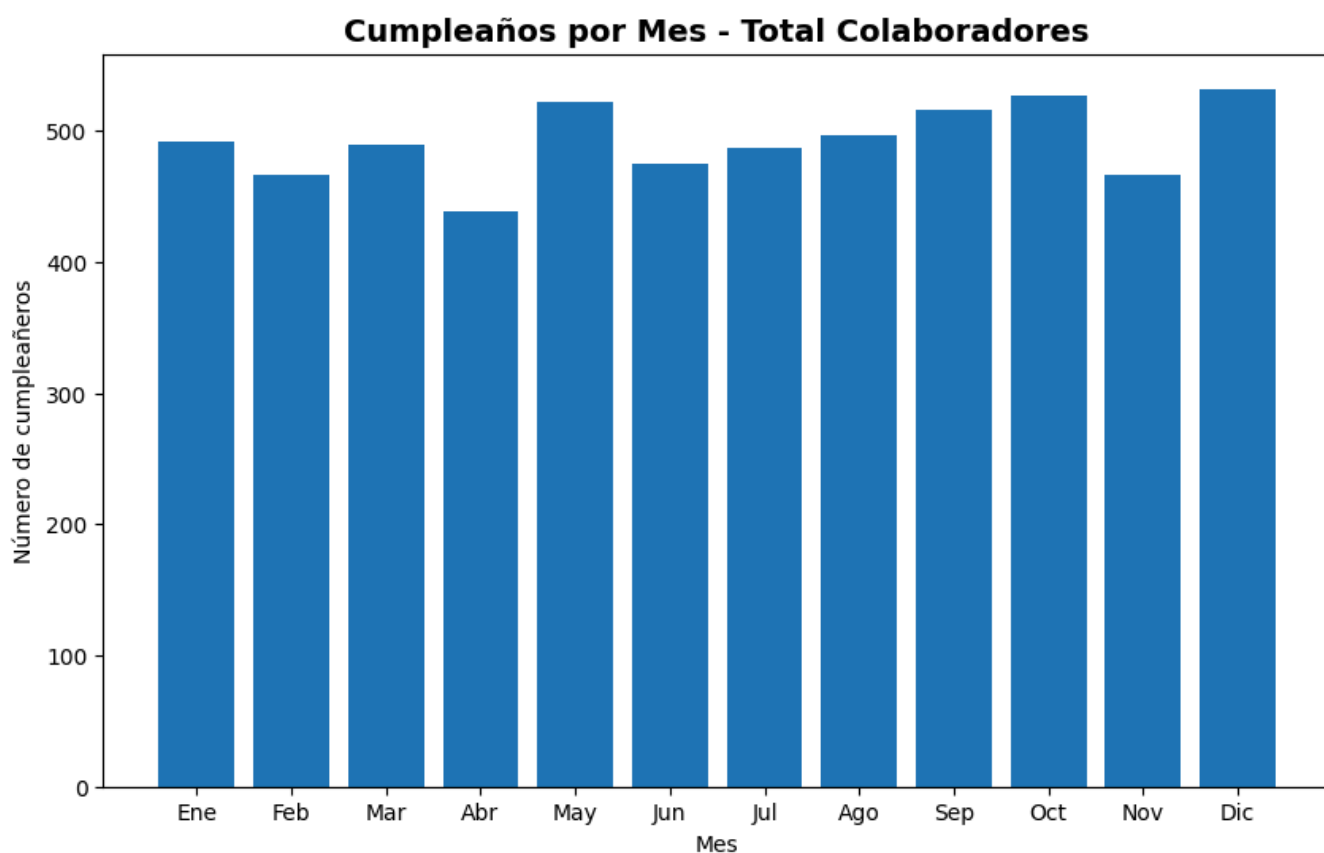
colaboradores, siendo estas unidades el grueso de negocio.

## 4.3 Análisis por mes de cumpleaños

```
In [74]: # Conteo de cumpleaños por mes
cumple_mes = ACTIVOS_FEB_24['MES_CUMPLE'].value_counts().sort_index()

# Elaboración del gráfico
plt.figure(figsize=(10,6))
plt.bar(cumple_mes.index, cumple_mes.values)
plt.xticks(range(1,13),
            ['Ene', 'Feb', 'Mar', 'Abr', 'May', 'Jun', 'Jul', 'Ago', 'Sep', 'Oct', 'Nov', 'Dic'])
plt.title("Cumpleaños por Mes - Total Colaboradores", fontsize=14, fontweight="bold")
plt.xlabel("Mes")
plt.ylabel("Número de cumpleaños")
plt.show()

###
```



🎉 La política de la empresa permite un día libre por cumpleaños.

Al analizar los meses de nacimiento de los colaboradores, se identifica que **mayo, octubre y diciembre** son los meses con mayor cantidad de días libres.

⚠️ Esto representa un **potencial riesgo para el negocio**, ya que la disponibilidad de personal podría reducirse en estos meses.

💡 Recomendaciones para mitigar el riesgo:

- 🔄 Asignar tareas de manera **multitarea**, evitando depender de una sola persona.
- 📅 Planificar la carga de trabajo para que **la operación no se vea afectada** durante estos meses críticos.

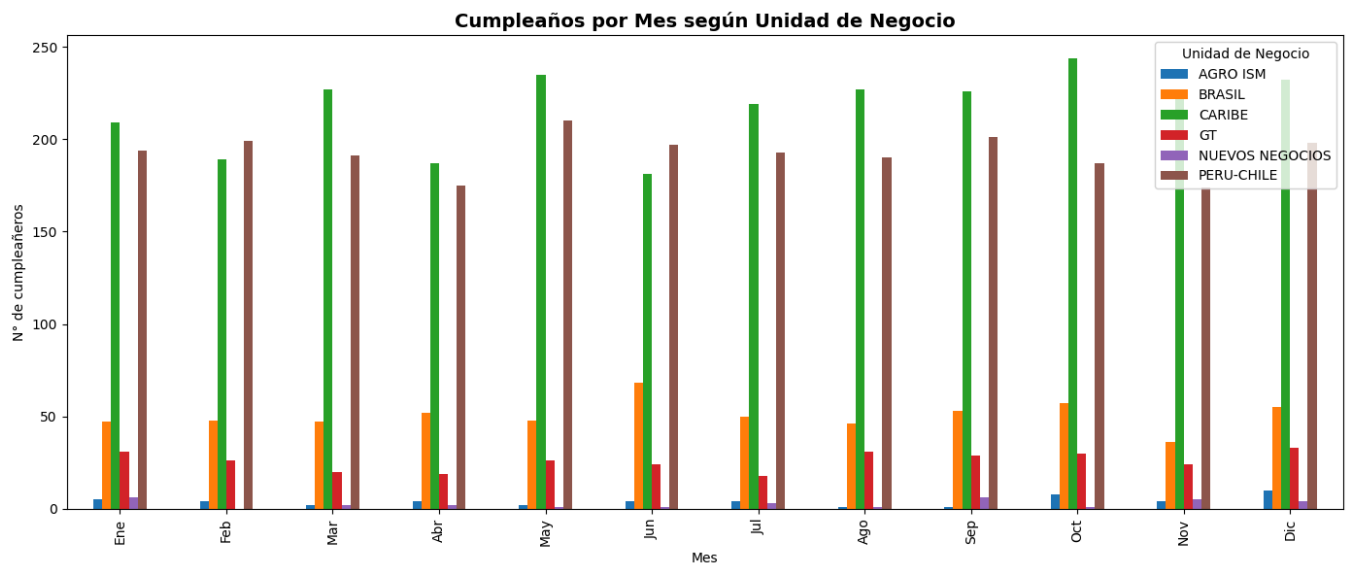
## 4.4 Análisis por mes de cumpleaños según Unidad de Negocio

```
In [19]: # Agrupación y conteo
cumple_por_unidad = ACTIVOS_FEB_24.groupby(["UNIDAD DE NEGOCIO", "MES_CUMPLE"]).size()

plt.figure(figsize=(14,7))
cumple_por_unidad.T.plot(kind='bar', figsize=(14,6))

plt.xticks(range(12),
            ['Ene', 'Feb', 'Mar', 'Abr', 'May', 'Jun', 'Jul', 'Ago', 'Sep', 'Oct', 'Nov', 'Dic'])
plt.title("Cumpleaños por Mes según Unidad de Negocio", fontsize=14, fontweight="bold")
plt.xlabel("Mes")
plt.ylabel("N° de cumpleaños")
plt.legend(title="Unidad de Negocio")
plt.tight_layout()
plt.show()
```

<Figure size 1400x700 with 0 Axes>



Al analizar los días de descanso por **Unidad de Negocio**, se observa lo siguiente:

- 🇧🇷 **Brasil:** junio es el mes con mayor riesgo de negocio debido a la concentración de días libres.
- 🇵🇪 **Perú y Chile:** el mayor riesgo se presenta en mayo.
- 🌴 **Caribe:** los meses con mayor riesgo son mayo y octubre.

✅ La diversificación de los meses con días libres entre las distintas Unidades de Negocio **reduce el riesgo de negocio de manera global**, ya que no todos los colaboradores están ausentes al mismo tiempo.

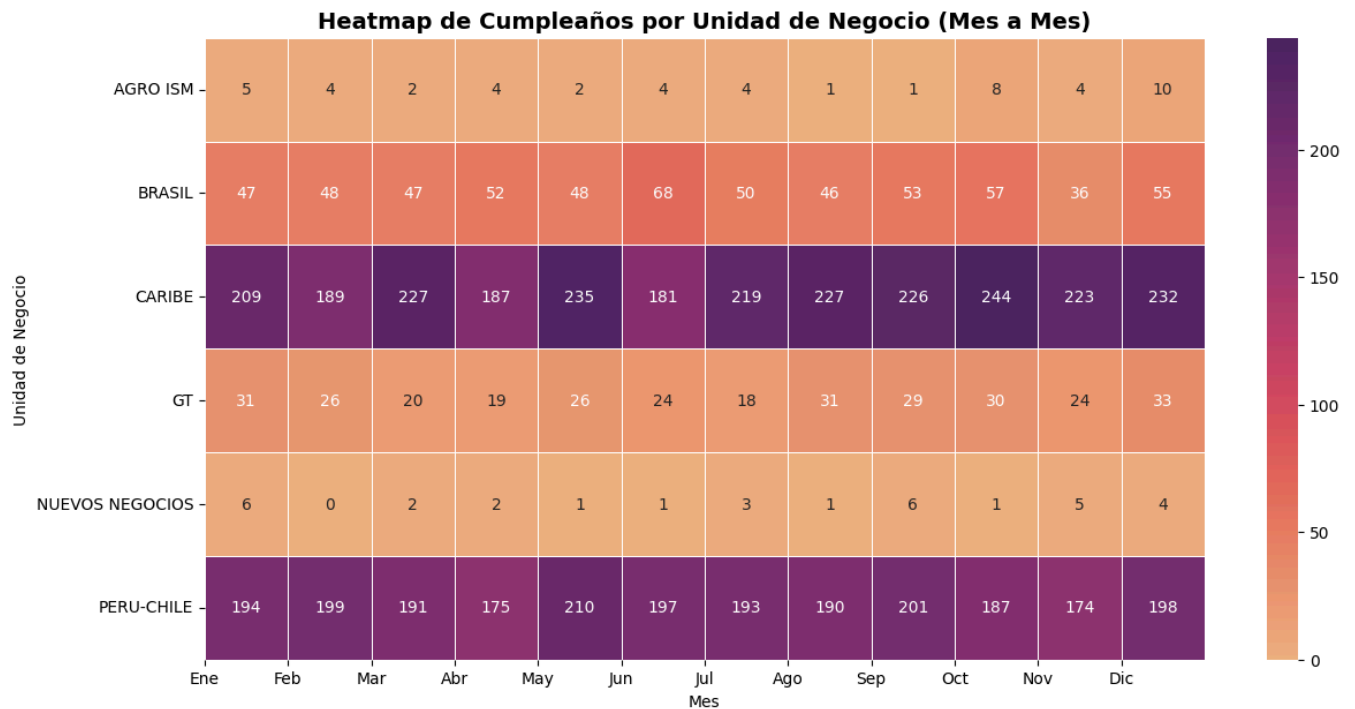
## 4.5 Heatmap por Unidad de Negocio

```
In [76]: # Tabla dinámica para el heatmap
heatmap_data = ACTIVOS_FEB_24.groupby(["UNIDAD DE NEGOCIO", "MES_CUMPLE"]).size().unstack()

plt.figure(figsize=(14,7))
sns.heatmap(heatmap_data, annot=True, fmt="d", cmap="flare", linewidths=.5)

plt.title("Heatmap de Cumpleaños por Unidad de Negocio (Mes a Mes)", fontsize=14, fontweight="bold")
plt.xlabel("Mes")
plt.ylabel("Unidad de Negocio")
plt.xticks(ticks=range(12), labels=['Ene', 'Feb', 'Mar', 'Abr', 'May', 'Jun', 'Jul', 'Ago', 'Sep', 'Oct', 'Nov', 'Dic'])
plt.yticks(rotation=0)
plt.show()
```





Con el **heatmap** confirmamos los meses de mayor riesgo por Unidad de Negocio (por riesgo nos referimos a **días libres según la política de la empresa**):

- 🇧🇷 **Brasil:** 🟡 **Junio** → mayor cantidad de días libres.
- 🌴 **Caribe:** 🟠 **Octubre**, seguido de 🟢 **Mayo** y 🔴 **Diciembre**  
 ⚠️ **Diciembre:** coincide con el fin de año contable → riesgo de retrasos en procesos y posibles **pagos adicionales** por venta de días libres.
- 🇵🇪🇨🇱 **Perú y Chile:** 🟡 **Mayo** → mes con mayor riesgo por días libres.

✅ **Conclusión:** La diversificación de los meses con días libres entre Unidades de Negocio ayuda a **reducir el riesgo global de negocio**, evitando ausencias concentradas en un mismo periodo.

```
In [77]: fig, axes = plt.subplots(2,2, figsize=(15,12))

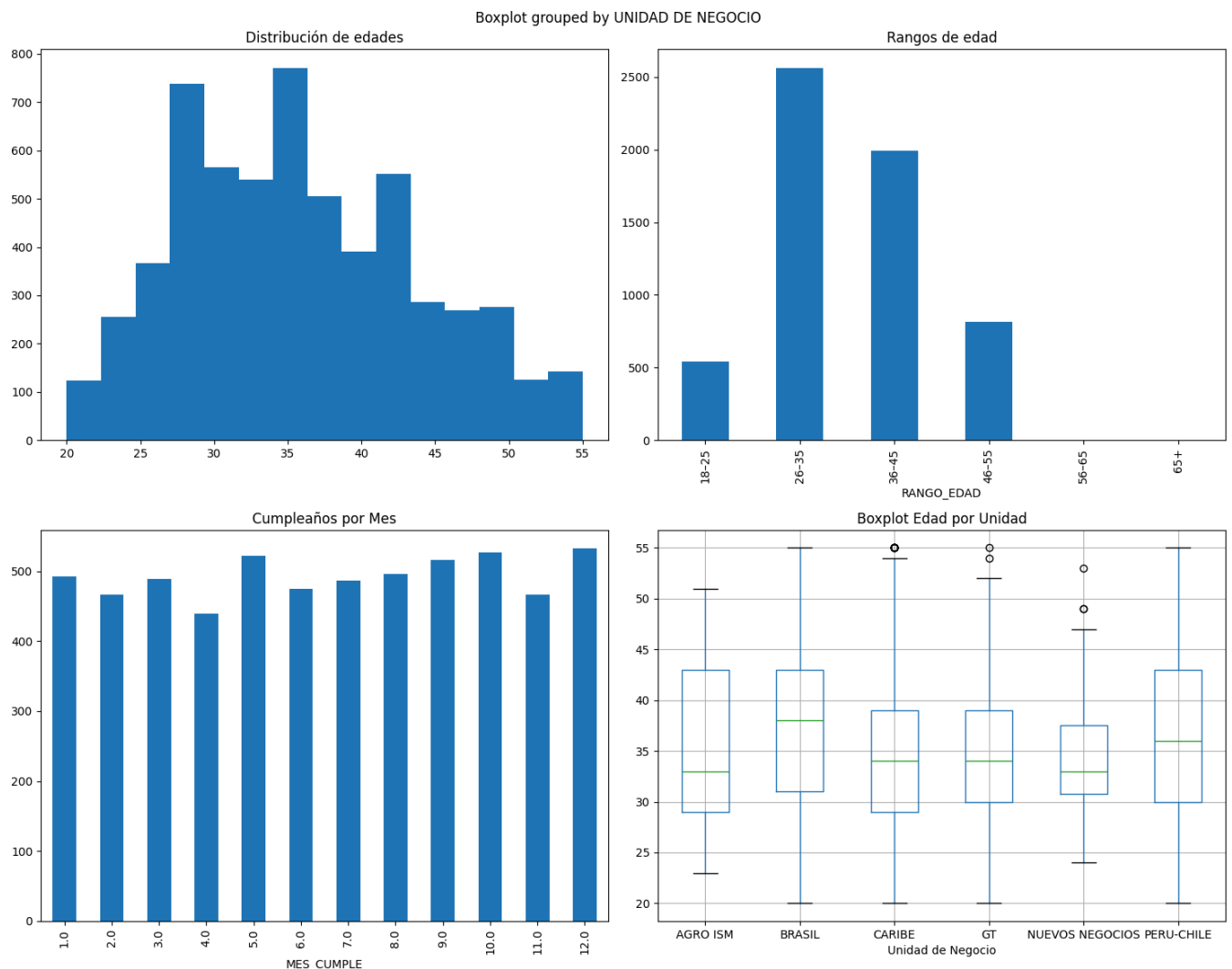
# 1. Distribución general de edades
axes[0,0].hist(ACTIVOS_FEB_24["EDAD"], bins=15)
axes[0,0].set_title("Distribución de edades")

# 2. Barras por rango etario
ACTIVOS_FEB_24["RANGO_EDAD"].value_counts().sort_index().plot(kind="bar", ax=axes[0,1])
axes[0,1].set_title("Rangos de edad")

# 3. Meses de cumpleaños
ACTIVOS_FEB_24.groupby("MES_CUMPLE").size().plot(kind="bar", ax=axes[1,0])
axes[1,0].set_title("Cumpleaños por Mes")

# 4. Edades por Unidad de negocio
ACTIVOS_FEB_24.boxplot(column="EDAD", by="UNIDAD DE NEGOCIO", ax=axes[1,1])
axes[1,1].set_title("Boxplot Edad por Unidad")
axes[1,1].set_xlabel("Unidad de Negocio")

plt.tight_layout()
plt.show()
```



## Conclusiones

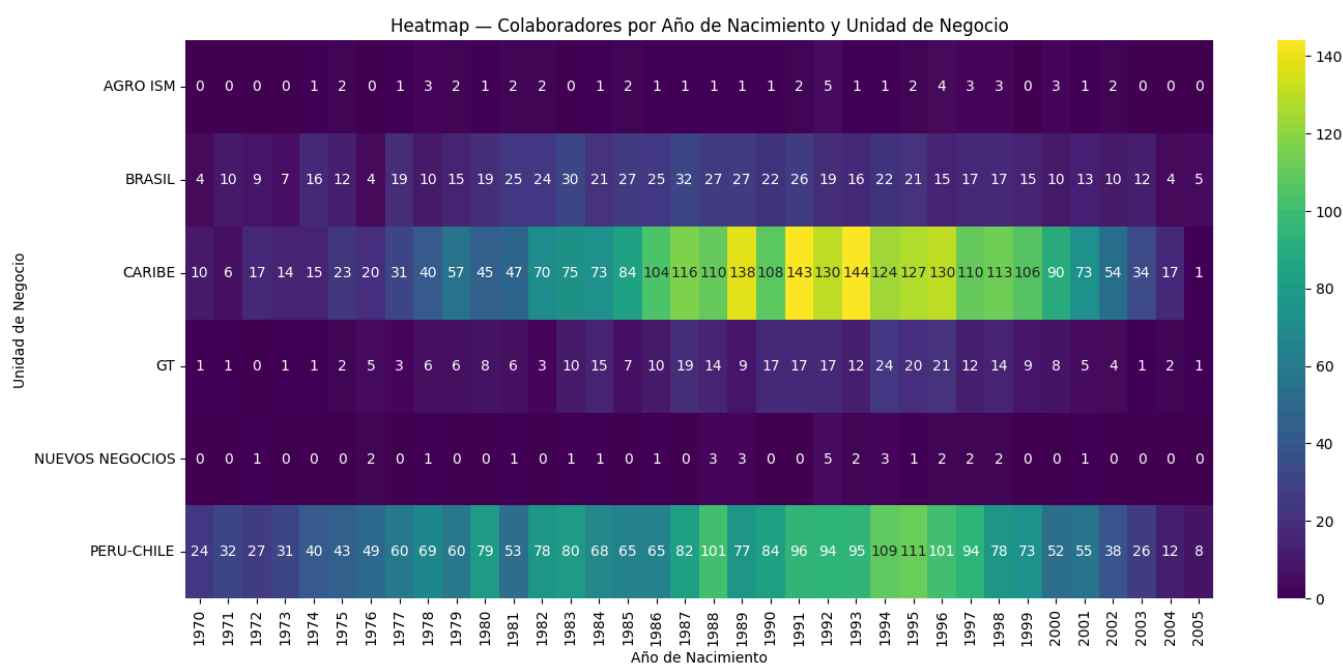
- La **Generación Y** representa el **mayor porcentaje de colaboradores** en esta empresa de consumo masivo estudiadas. Esto implica que las estrategias de desarrollo y gestión deben adaptarse principalmente a sus expectativas y motivaciones, ya que su comportamiento y desempeño marcarán gran parte del funcionamiento del sector.
- La diversidad generacional en las organizaciones es un factor clave que impacta positiva o negativamente en los resultados del negocio. La convivencia de distintos grupos requiere políticas adaptadas a las características de cada generación, especialmente de la Generación Y (Alvarado y Huayta, 2025).
- El desarrollo de los jóvenes de la Generación Y no solo depende de la adquisición de conocimientos y competencias, sino también de hábitos y comportamientos nuevos. Por ello, un plan de desarrollo enfocado en esta generación es fundamental para alinear sus capacidades con los objetivos estratégicos de la empresa (Alvarado y Huayta, 2025).
- La investigación primaria confirmó que los jóvenes de la Generación Y asocian el desarrollo con la adquisición de poder y posiciones de decisión relevantes en la organización. Esto refleja que, para motivarlos y retenerlos, las empresas deben ofrecer oportunidades de crecimiento y participación en la toma de decisiones (Alvarado y Huayta, 2025).
- La implementación de un plan de desarrollo adaptado a la Generación Y permite a las empresas del sector de consumo masivo **optimizar el talento y garantizar resultados**

alineados con los objetivos organizacionales, considerando tanto la perspectiva de los empleadores como la de los colaboradores (Alvarado y Huayta, 2025).

```
In [78]: ACTIVOS_FEB_24["AÑO_NAC"] = ACTIVOS_FEB_24["FECHA DE NACIMIENTO (DD/MM/YYYY)"].dt.year

heatmap_edad = ACTIVOS_FEB_24.groupby(["UNIDAD DE NEGOCIO", "AÑO_NAC"]).size().unstack()

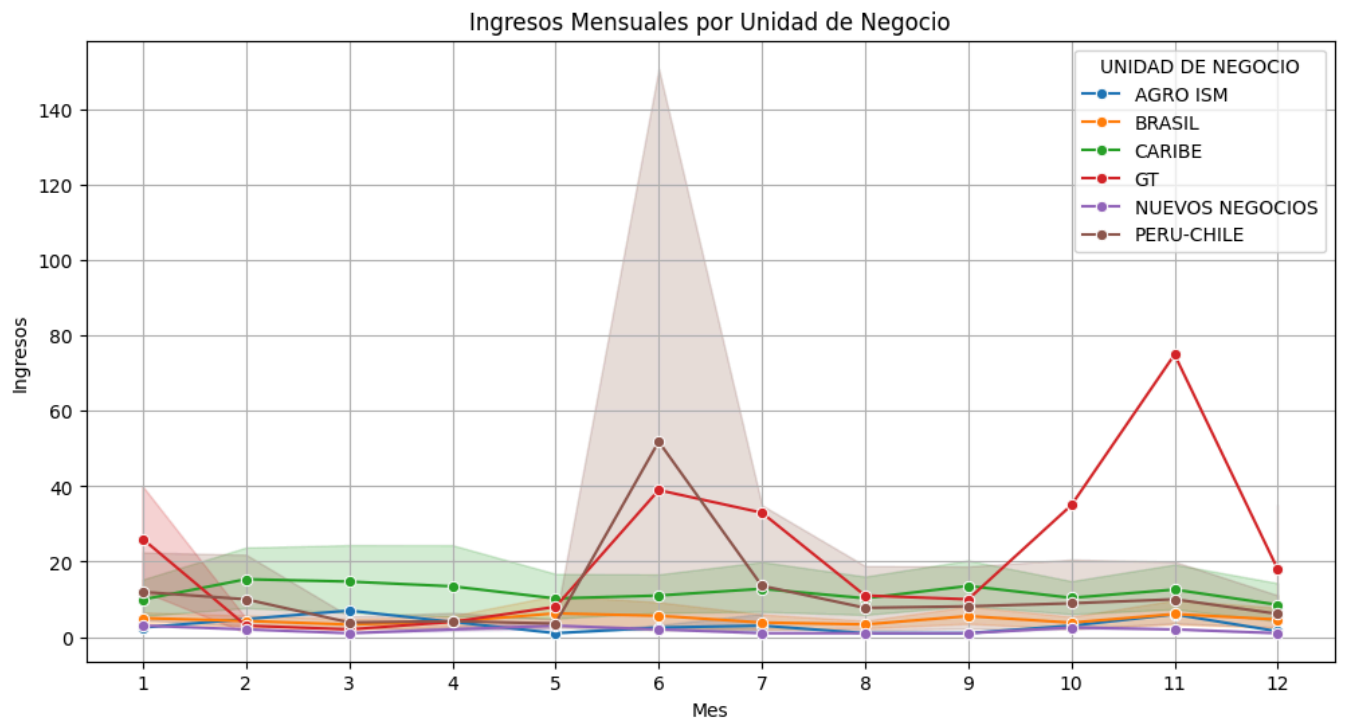
plt.figure(figsize=(16,7))
sns.heatmap(heatmap_edad, cmap="viridis", annot=True, fmt="d")
plt.title("Heatmap - Colaboradores por Año de Nacimiento y Unidad de Negocio")
plt.ylabel("Unidad de Negocio")
plt.xlabel("Año de Nacimiento")
plt.show()
```



## 4.6 Ingresos de nuevos colaboradores - Estacionalidad de ingresos

```
In [79]: ## Definir la variable
ingresos_unidad = ACTIVOS_FEB_24.groupby(["UNIDAD DE NEGOCIO", "AÑO", "MES_NUM"]).si

## Realizar la gráfica
plt.figure(figsize=(12,6))
sns.lineplot(data=ingresos_unidad, x="MES_NUM", y="INGRESOS", hue="UNIDAD DE NEGOCIO")
plt.title("Ingresos Mensuales por Unidad de Negocio")
plt.xticks(range(1,13))
plt.xlabel("Mes")
plt.ylabel("Ingresos")
plt.grid(True)
plt.show()
```



Los meses con mayores ingresos son **junio** para Guatemala y Perú-Chile, y **noviembre** para Guatemala. Esta distribución sugiere un patrón de **estacionalidad**.

- En Perú-Chile, los ingresos más altos se registran en **junio**, siendo estos el segundo mes con mayores ingresos a nivel global. Esta estacionalidad podría estar relacionada con las **fiestas patrias en julio**, que impulsan campañas de ventas y generación de ingresos año tras año.
- En Guatemala, el pico de ingresos en **noviembre** puede explicarse por la estacionalidad asociada a **campañas de ventas de fin de año**, que comienzan en octubre durante el "mes morado" y se intensifican en noviembre al acercarse la temporada navideña. Además que Guatemala es una nueva Unidad de negocio.

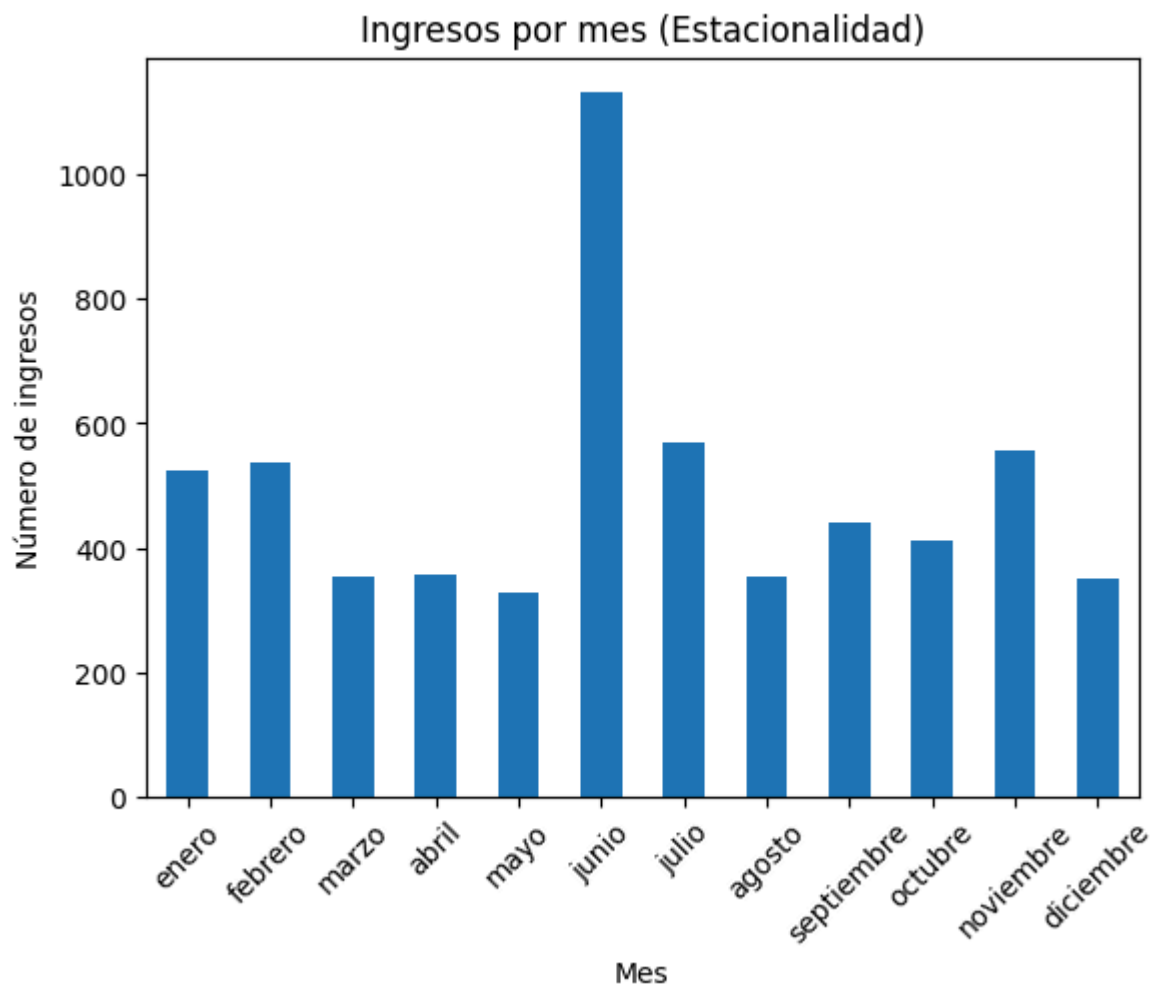
En conjunto, estos patrones indican que los ingresos en el sector presentan **variaciones predecibles por estacionalidad**, lo que permite planificar mejor las operaciones y estrategias comerciales.

```
In [81]: ACTIVOS_FEB_24["Mes_ingreso"] = ACTIVOS_FEB_24["FECHA DE INGRESO (DD/MM/YYYY)"].dt.m
orden_mes = ["enero", "febrero", "marzo", "abril", "mayo", "junio",
            "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"]

ACTIVOS_FEB_24["Mes_ingreso"] = pd.Categorical(
    ACTIVOS_FEB_24["Mes_ingreso"].str.lower(),
    categories=orden_mes,
    ordered=True
)

ingresos_mes = ACTIVOS_FEB_24["Mes_ingreso"].value_counts().sort_index()

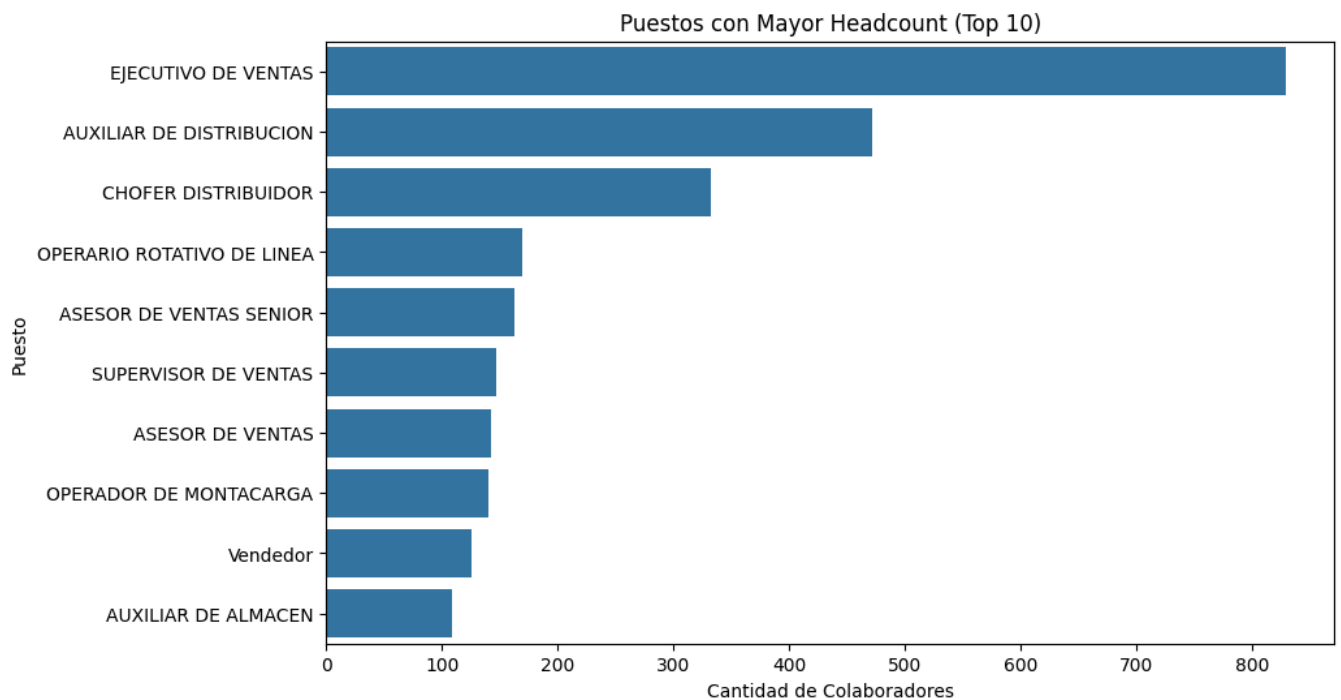
ingresos_mes.plot(kind="bar")
plt.title("Ingresos por mes (Estacionalidad)")
plt.xlabel("Mes")
plt.ylabel("Número de ingresos")
plt.xticks(rotation=45)
plt.show()
```



A nivel global, los ingresos muestran una **alta estacionalidad en junio**, lo cual podría explicarse porque, al ser una empresa de origen peruano, se busca **celebrar las fiestas patrias** no solo en Perú, sino también en los países donde la empresa tiene presencia.

```
In [82]: # Comprobación de estacionalidad
### Creación de la variable para ver top puestos:
top_puestos = ACTIVOS_FEB_24["POSICION / PUESTO / CARGO"].value_counts().head(10)

### Creación del gráfico
plt.figure(figsize=(10,6))
sns.barplot(x=top_puestos.values, y=top_puestos.index)
plt.title("Puestos con Mayor Headcount (Top 10)")
plt.xlabel("Cantidad de Colaboradores")
plt.ylabel("Puesto")
plt.show()
```

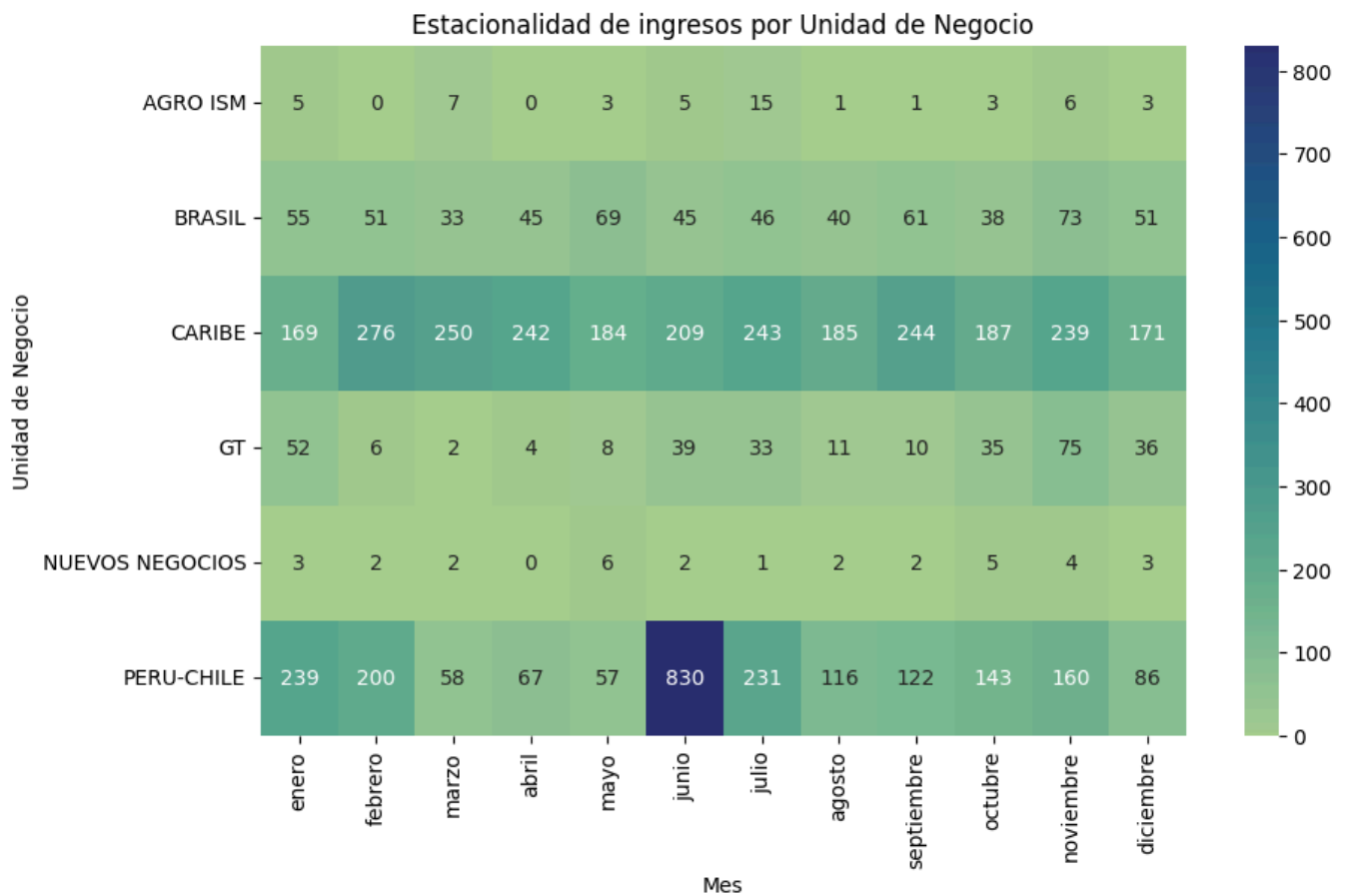


Se observa que el **puesto con mayores ingresos es el de Ejecutivo de Ventas**, lo cual podría reflejar la estacionalidad de los ingresos. Esto se debe a que los Ejecutivos de Ventas suelen tener **periodos de permanencia cortos** y presentan **alta rotación**, dado que son puestos enfocados en campañas y no tanto permanentes.

```
In [83]: ingresos_unidad_mes = (
    ACTIVOS_FEB_24
    .groupby(["UNIDAD DE NEGOCIO", "Mes_ingreso"])
    .size()
    .unstack(fill_value=0)
)

plt.figure(figsize=(10,6))
sns.heatmap(ingresos_unidad_mes, annot=True, fmt="d", cmap="crest")
plt.title("Estacionalidad de ingresos por Unidad de Negocio")
plt.xlabel("Mes")
plt.ylabel("Unidad de Negocio")
plt.show()
```

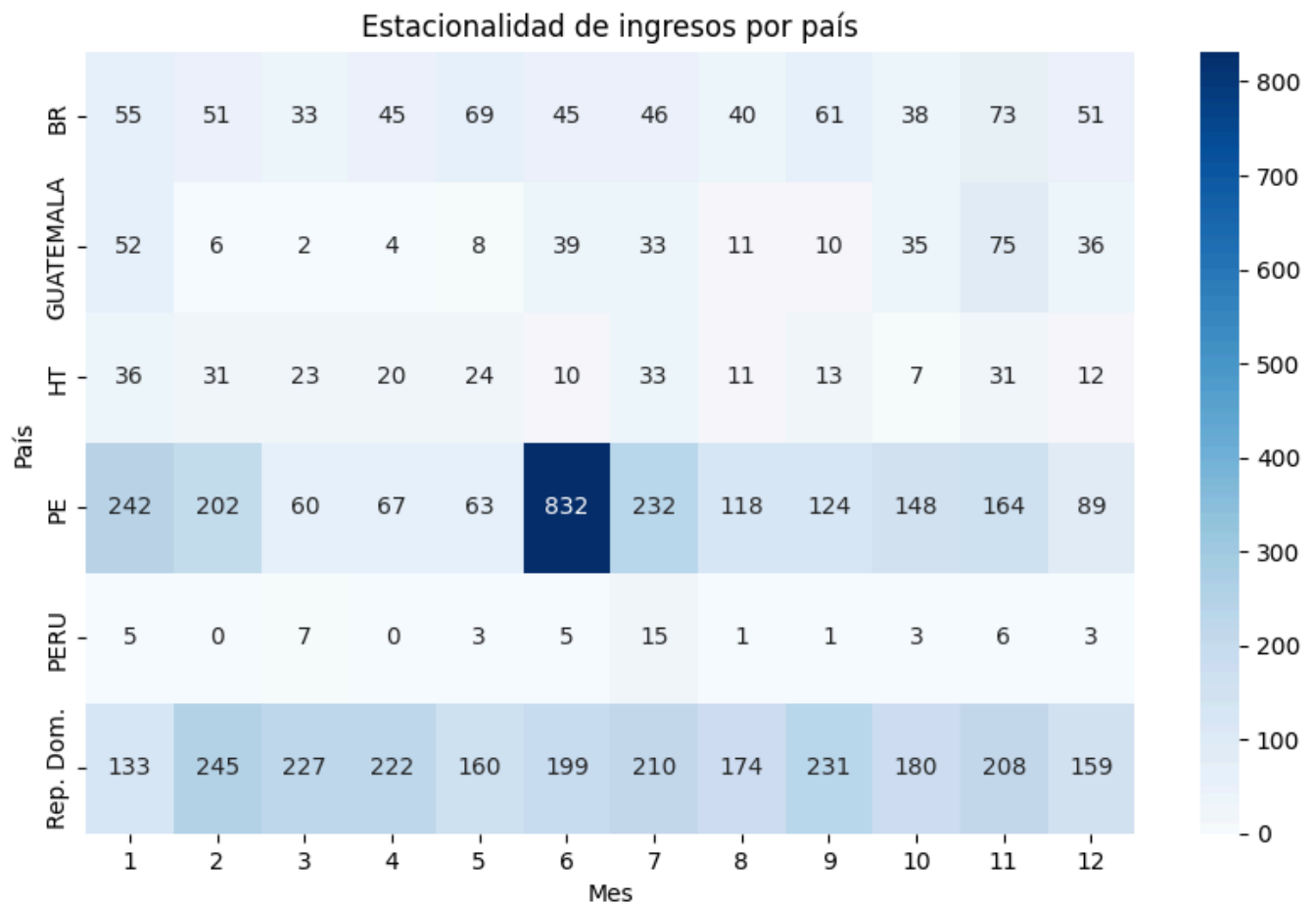
```
/var/folders/7l/8rl5yq1j1g9684gbgh80kqq80000gn/T/ipykernel_65621/3877741772.py:3: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
    .groupby(["UNIDAD DE NEGOCIO", "Mes_ingreso"])
```



```
In [84]: ACTIVOS_FEB_24["Mes"] = ACTIVOS_FEB_24["FECHA DE INGRESO (DD/MM/YYYY)"].dt.month

season_country = (
    ACTIVOS_FEB_24.groupby(["pais", "Mes"])
    .size()
    .unstack(fill_value=0)
)

plt.figure(figsize=(10,6))
sns.heatmap(season_country, cmap="Blues", annot=True, fmt="d")
plt.title("Estacionalidad de ingresos por país")
plt.xlabel("Mes")
plt.ylabel("País")
plt.show()
```



En los heatmaps se observa que la **estacionalidad de ingresos se presenta principalmente en Perú durante el mes de junio**. Este es el único mes que muestra una estacionalidad marcada; en los demás países, los ingresos se distribuyen de manera más equilibrada, lo que no representa un riesgo significativo para el negocio.

Sin embargo, en Perú, el mes de junio podría implicar ciertos riesgos:

- **Si se incorporan nuevos colaboradores**, se pierde el conocimiento adquirido en campañas anteriores, lo que genera mayores gastos en aprendizaje y desarrollo.
- **Si se cuenta con el mismo personal**, se aprovecha la experiencia previa, lo que reduce la necesidad de capacitaciones intensivas. No obstante, cuando los colaboradores son jóvenes y los puestos son de entrada, existe el riesgo de que el trabajo resulte monótono, lo que podría afectar la efectividad de la campaña.

Por ello, se recomienda **mantener un porcentaje de colaboradores con experiencia**, de modo que se reduzcan los costos de capacitación y se maximice la eficiencia de cada campaña.

```
In [85]: # Extraer mes numérico y nombre de mes
ACTIVOS_FEB_24["Mes"] = ACTIVOS_FEB_24["FECHA DE INGRESO (DD/MM/YYYY)"].dt.month
ACTIVOS_FEB_24["Mes_nombre"] = ACTIVOS_FEB_24["FECHA DE INGRESO (DD/MM/YYYY)"].dt.month_name()

# Convertir mes a minúsculas para que coincida con el reindex
ACTIVOS_FEB_24["Mes_nombre"] = ACTIVOS_FEB_24["Mes_nombre"].str.lower()

orden_meses = ["enero", "febrero", "marzo", "abril", "mayo", "junio",
               "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"]

genero_mes = (
    ACTIVOS_FEB_24.groupby(["Mes_nombre", "GENERO (F/M)"])
    .size()
    .unstack(fill_value=0)
```



```

        .reindex(orden_meses)
    )

    genero_mes.plot(kind="bar", figsize=(11,6))
    plt.title("📊 Ingresos por género a lo largo del año")
    plt.ylabel("Cantidad de ingresos")
    plt.xlabel("Mes")
    plt.xticks(rotation=45)
    plt.legend(title="Género")
    plt.tight_layout()
    plt.show()

```

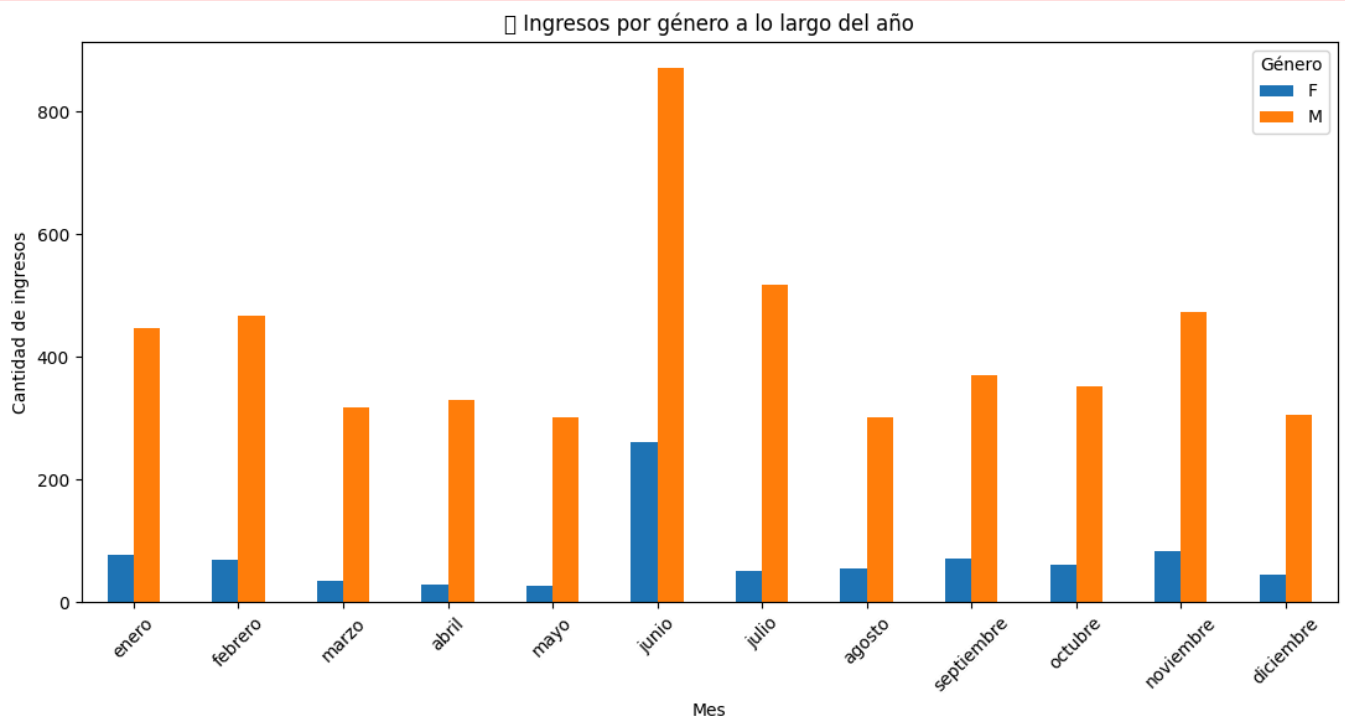
/var/folders/7l/8rl5yq1j1g9684gbgh80kqq80000gn/T/ipykernel\_65621/2706672040.py:24: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from current font.

```

    plt.tight_layout()
/Users/rodrigoluisyauliblas/Library/Python/3.12/lib/python/site-packages/IPython/core/pylabtools.py:170: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from current font.

```

```
fig.canvas.print_figure(bytes_io, **kw)
```



En los ingresos por puestos se observa que en algunos meses los ingresos varían mientras que en mujeres decrece y en hombres se mantiene igual, ambos tienen como estacionalidad al mes de junio.

## 4.7 Riesgo ingreso por puesto

```

In [87]: # Contar ingresos por puesto por mes
ingreso_puesto_mes = (
    ACTIVOS_FEB_24.groupby(["Mes_nombre", "POSICION / PUESTO / CARGO"])
    .size()
    .unstack(fill_value=0)
)

# Calcular volatilidad (riesgo)
riesgo_ingresos = ingreso_puesto_mes.std().sort_values(ascending=False)

print("\n🔥 Ranking de riesgo por volatilidad de ingresos:\n")
print(riesgo_ingresos.head(10))

```

## 📌 Ranking de riesgo por volatilidad de ingresos:

POSICION / PUESTO / CARGO	
EJECUTIVO DE VENTAS	148.568829
SUPERVISOR DE VENTAS	24.008048
AUXILIAR DE DISTRIBUCION	12.499697
OPERARIO ROTATIVO DE LINEA	11.405408
Vendedor	10.344080
EJECUTIVO(A) DE VENTAS	8.833047
AUXILIAR DE ALMACEN MONTACARGUISTA	7.833011
ALMACENERO	7.113794
CHOFER DISTRIBUIDOR	6.919450
AUXILIAR DE LINEA	6.297546
dtype: float64	

- El **puesto con mayor volatilidad de ingresos** es el **Ejecutivo de Ventas**, con un valor de 148.57, lo que indica que este rol es altamente sensible a cambios en las campañas, estacionalidad o desempeño individual. Esto implica que las estrategias de ingresos dependen en gran medida de este puesto.
- Los puestos de **Supervisor de Ventas** y **Auxiliar de Distribución** muestran volatilidad considerable, pero mucho menor que la de los Ejecutivos de Ventas, con valores de 24.01 y 12.50 respectivamente, lo que sugiere que estos roles también impactan los ingresos, aunque de manera más estable.
- Los puestos operativos como **Operario Rotativo de Línea, Vendedor, Almacenero, Auxiliar de Línea y Chofer Distribuidor** presentan volatilidad baja (entre 6.3 y 11.4), indicando que estos roles generan ingresos más estables y predecibles.
- En general, **la mayor concentración de riesgo se encuentra en los roles comerciales**, especialmente en ventas, mientras que los roles operativos aportan estabilidad al negocio. Esto sugiere que cualquier planificación financiera o de recursos humanos debe **considerar medidas para mitigar el riesgo en los puestos de ventas**, como capacitación, rotación estratégica de personal y campañas multitarifa para balancear la volatilidad.

```
In [46]: # Mes numérico y nombre en minúsculas ordenable
ACTIVOS_FEB_24["Mes_num"] = ACTIVOS_FEB_24["FECHA DE INGRESO (DD/MM/YYYY)"].dt.month
ACTIVOS_FEB_24["Mes_nombre"] = ACTIVOS_FEB_24["FECHA DE INGRESO (DD/MM/YYYY)"].dt.month_name

orden_meses = ["enero", "febrero", "marzo", "abril", "mayo", "junio",
               "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"]
ACTIVOS_FEB_24["Mes_nombre"] = pd.Categorical(ACTIVOS_FEB_24["Mes_nombre"], categories=orden_meses)

# Tabla: ingresos por puesto por mes (rows: Mes, cols: Puesto)
ingreso_puesto_mes = (
    ACTIVOS_FEB_24
    .groupby(["Mes_num", "POSICION / PUESTO / CARGO"])
    .size()
    .unstack(fill_value=0)
    .reindex(range(1,13), fill_value=0) # asegurar 12 meses
)

# Calcular volatilidad (std) y media mensual para cada puesto
riesgo_puestos = pd.DataFrame({
    "std_ingresos": ingreso_puesto_mes.std(axis=0),
    "media_ingresos": ingreso_puesto_mes.mean(axis=0),
    "headcount_total": ACTIVOS_FEB_24["POSICION / PUESTO / CARGO"].value_counts()
```

```
})
```

```
# Ordenar por volatilidad descendente y tomar top 15
```

```
top15_riesgo = riesgo_puestos.sort_values("std_ingresos", ascending=False).head(15)
```

```
print("Top 15 puestos por volatilidad (riesgo):")
```

```
print(top15_riesgo[["std_ingresos","media_ingresos","headcount_total"]])
```

Top 15 puestos por volatilidad (riesgo):

	std_ingresos	media_ingresos	\
POSICION / PUESTO / CARGO			
EJECUTIVO DE VENTAS	148.568829	69.166667	
SUPERVISOR DE VENTAS	24.008048	12.250000	
AUXILIAR DE DISTRIBUCION	12.499697	39.333333	
OPERARIO ROTATIVO DE LINEA	11.405408	14.083333	
Vendedor	10.344080	10.500000	
EJECUTIVO(A) DE VENTAS	8.833047	4.750000	
AUXILIAR DE ALMACEN MONTACARGUISTA	7.833011	4.083333	
ALMACENERO	7.113794	3.666667	
CHOFER DISTRIBUIDOR	6.919450	27.666667	
AUXILIAR DE LINEA	6.297546	8.250000	
PROMOTOR DE VENDAS	5.743903	5.916667	
ASISTENTE CONTABLE	5.407626	3.833333	
ASESOR DE VENTAS SENIOR	5.367551	13.583333	
SUPERVISOR DE DISTRIBUCION	5.187397	3.000000	
JEFE DE VENTAS	5.149287	2.833333	

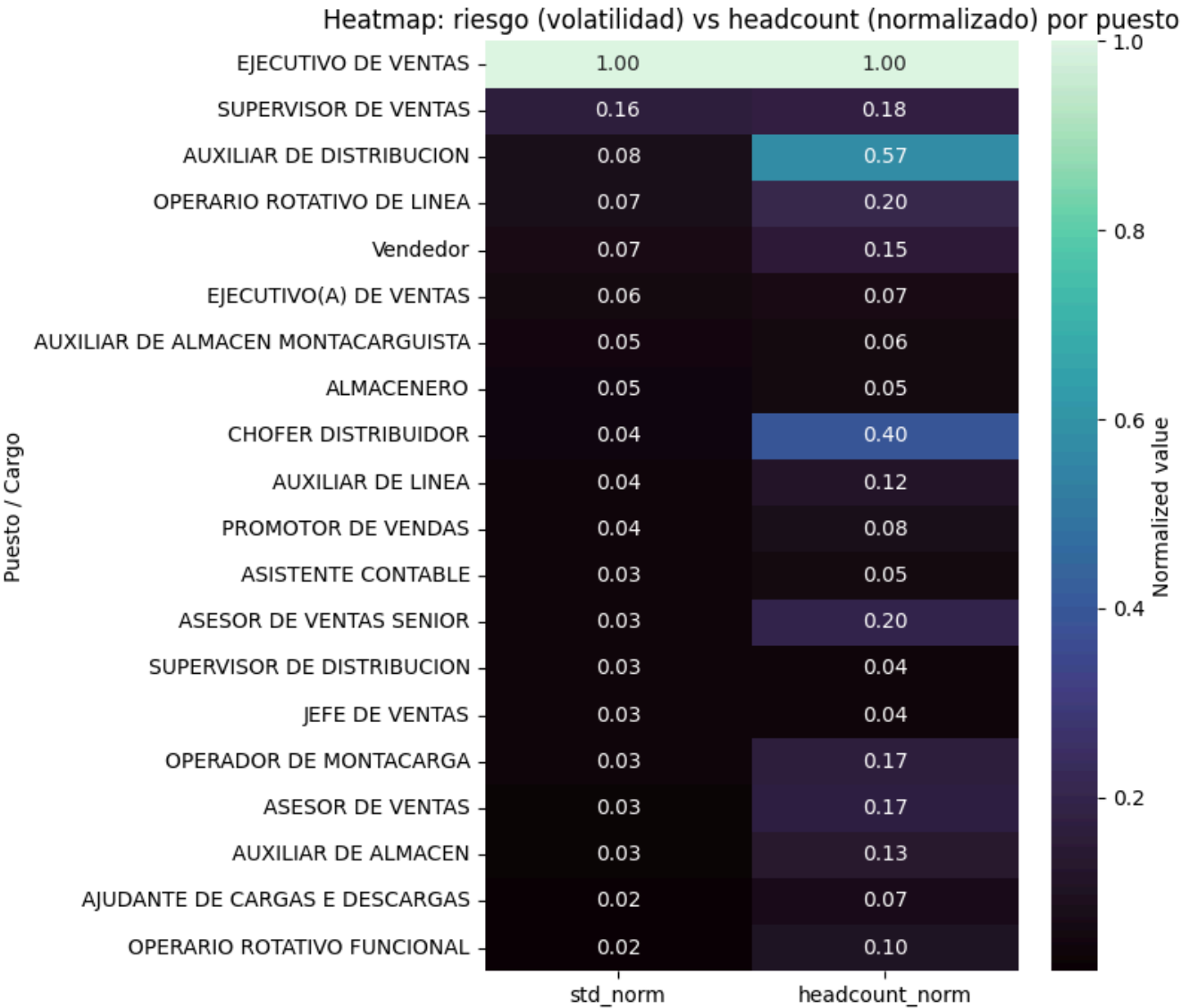
	headcount_total
POSICION / PUESTO / CARGO	
EJECUTIVO DE VENTAS	830
SUPERVISOR DE VENTAS	147
AUXILIAR DE DISTRIBUCION	472
OPERARIO ROTATIVO DE LINEA	169
Vendedor	126
EJECUTIVO(A) DE VENTAS	57
AUXILIAR DE ALMACEN MONTACARGUISTA	49
ALMACENERO	44
CHOFER DISTRIBUIDOR	332
AUXILIAR DE LINEA	99
PROMOTOR DE VENDAS	71
ASISTENTE CONTABLE	46
ASESOR DE VENTAS SENIOR	163
SUPERVISOR DE DISTRIBUCION	36
JEFE DE VENTAS	34

```
In [47]: # Normalizar para visual (0-1) – evita distorsión por escalas
df_heat = riesgo_puestos.copy()
df_heat["std_norm"] = (df_heat["std_ingresos"] - df_heat["std_ingresos"].min()) / (d
df_heat["headcount_norm"] = (df_heat["headcount_total"] - df_heat["headcount_total"]

# Selecciona puestos para el heatmap (top 30 por headcount o por riesgo, según prefir
puestos_sel = pd.concat([
    df_heat.sort_values("std_ingresos", ascending=False).head(15),
    df_heat.sort_values("headcount_total", ascending=False).head(15)
]).index.unique()

heatmap_df = df_heat.loc[puestos_sel, ["std_norm", "headcount_norm"]].sort_values("st

plt.figure(figsize=(8, max(6, len(heatmap_df)*0.35)))
sns.heatmap(heatmap_df, annot=True, fmt=".2f", cmap="mako", cbar_kws={"label": "Norma
plt.title("Heatmap: riesgo (volatilidad) vs headcount (normalizado) por puesto")
plt.ylabel("Puesto / Cargo")
plt.xlabel("")
plt.tight_layout()
plt.show()
```



Puesto crítico: EJECUTIVO DE VENTAS

- Tiene el mayor riesgo (1.00) y el mayor headcount (1.00).
- Esto indica que no solo es el rol más numeroso, sino también el más volátil.
- Requiere atención prioritaria en estrategias de retención, formación y estabilidad contractual.

#### Puestos con alta dotación pero baja volatilidad

- AUXILIAR DE DISTRIBUCIÓN: headcount alto (0.57) pero riesgo bajo (0.08).
- CHOFER DISTRIBUIDOR: headcount medio (0.40) y riesgo muy bajo (0.04).
- Estos roles muestran estabilidad pese a su volumen, lo que sugiere procesos bien establecidos o menor presión laboral.

#### Puestos con baja dotación y baja volatilidad

- Ejemplos: ASISTENTE CONTABLE, JEFE DE VENTAS, SUPERVISOR DE DISTRIBUCIÓN.
- Aunque no representan un riesgo inmediato, podrían ser sensibles a cambios individuales por su baja cantidad.

#### Puestos con riesgo moderado y headcount medio

- SUPERVISOR DE VENTAS: riesgo (0.16), headcount (0.18).
- OPERADOR DE MONTACARGA y ASESOR DE VENTAS: ambos con headcount (0.17) y riesgo (0.03).
- Estos roles podrían estar en transición o tener desafíos específicos que afectan su estabilidad.

```
In [88]: hoy = pd.to_datetime("2025-12-07") # puedes cambiar la fecha de referencia

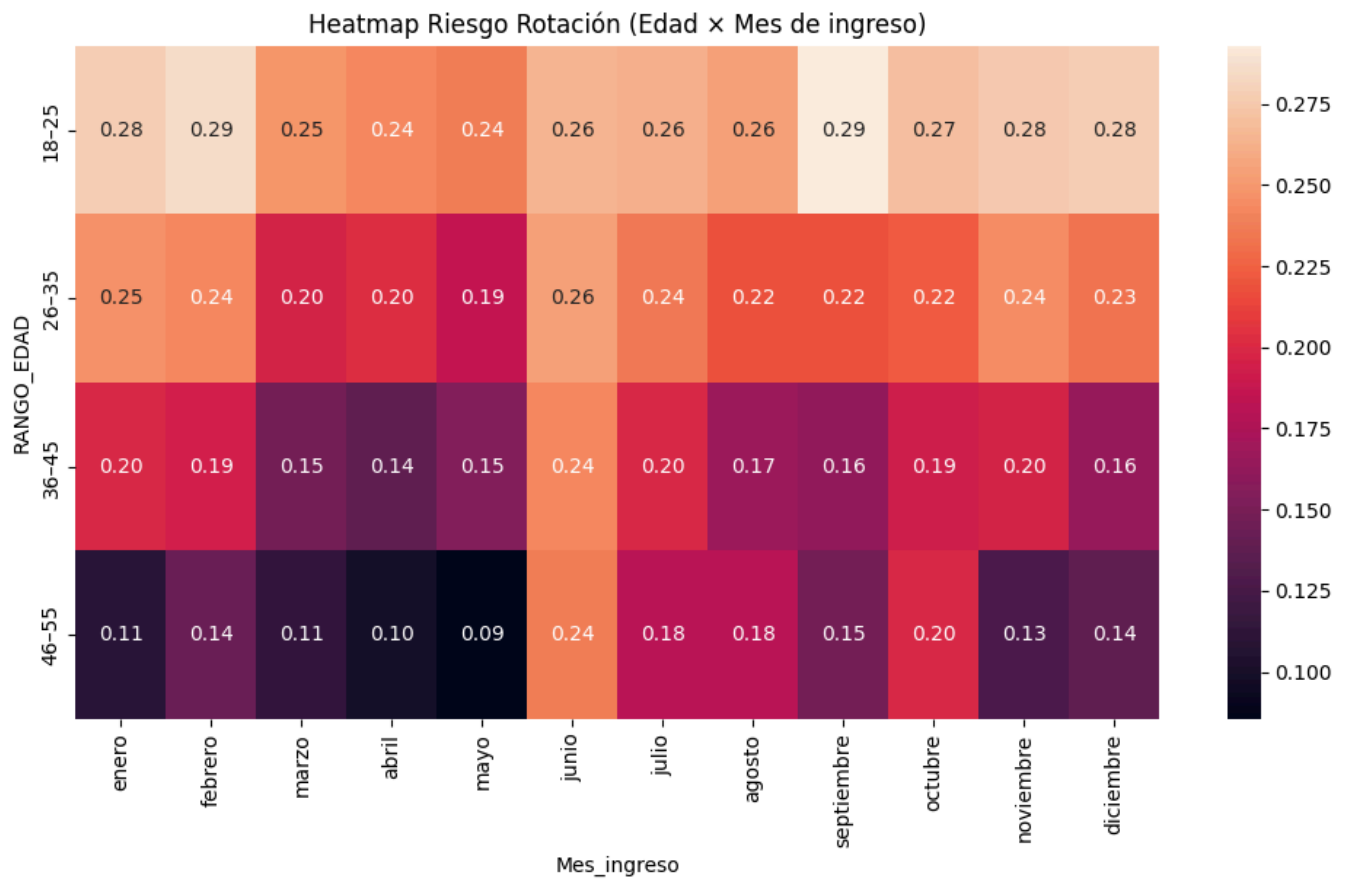
ACTIVOS_FEB_24["antiguedad_anios"] = ((hoy - ACTIVOS_FEB_24["FECHA DE INGRESO (DD/MM/AA)"]
                                         / pd.Timedelta(days=365)).round(2))

ACTIVOS_FEB_24["riesgo_rotacion"] = 1 / (ACTIVOS_FEB_24["antiguedad_anios"] + 1)

pivot = ACTIVOS_FEB_24.pivot_table(
    values="riesgo_rotacion",
    index="RANGO_EDAD",
    columns="Mes_ingreso",
    aggfunc="mean"
)
pivot

plt.figure(figsize=(12,6))
sns.heatmap(pivot, annot=True, fmt=".2f")
plt.title("Heatmap Riesgo Rotación (Edad x Mes de ingreso)")
plt.show()
```

```
/var/folders/7l/8rl5yq1j1g9684gbgh80kqq80000gn/T/ipykernel_65621/2352330151.py:8: FutureWarning: The default value of observed=False is deprecated and will change to observed=True in a future version of pandas. Specify observed=False to silence this warning and retain the current behavior
  pivot = ACTIVOS_FEB_24.pivot_table(
```



#### Mayor riesgo en jóvenes (18-25 años)

- Este grupo presenta consistentemente los valores más altos de riesgo de rotación, con cifras entre 0.24 y 0.29.
- El riesgo más alto se observa en febrero (0.29) y septiembre (0.29).
- Esto sugiere que los empleados más jóvenes son más propensos a dejar la empresa, posiblemente por menor estabilidad laboral, búsqueda de oportunidades o falta de compromiso inicial.

#### Riesgo moderado en adultos jóvenes (26-35 años)

- Los valores oscilan entre 0.19 y 0.26, mostrando una tendencia más estable.
- El mes de mayo (0.19) tiene el riesgo más bajo, mientras que junio (0.26) es el más alto.
- Este grupo podría estar más consolidado en sus carreras, pero aún muestra cierta movilidad.

#### Riesgo bajo en adultos medios (36-45 años)

- Los valores van de 0.14 a 0.24, con los más bajos en abril (0.14) y mayo (0.15).
- Este grupo parece tener mayor estabilidad laboral, posiblemente por compromisos familiares o mayor antigüedad.

#### Menor riesgo en adultos mayores (46-55 años)

- Este grupo tiene los valores más bajos, entre 0.09 y 0.24.
- El riesgo mínimo se da en mayo (0.09), y el máximo en junio (0.24).
- La rotación en este grupo es baja, lo que puede reflejar mayor lealtad o cercanía a la jubilación.

#### Meses críticos: febrero, junio y septiembre

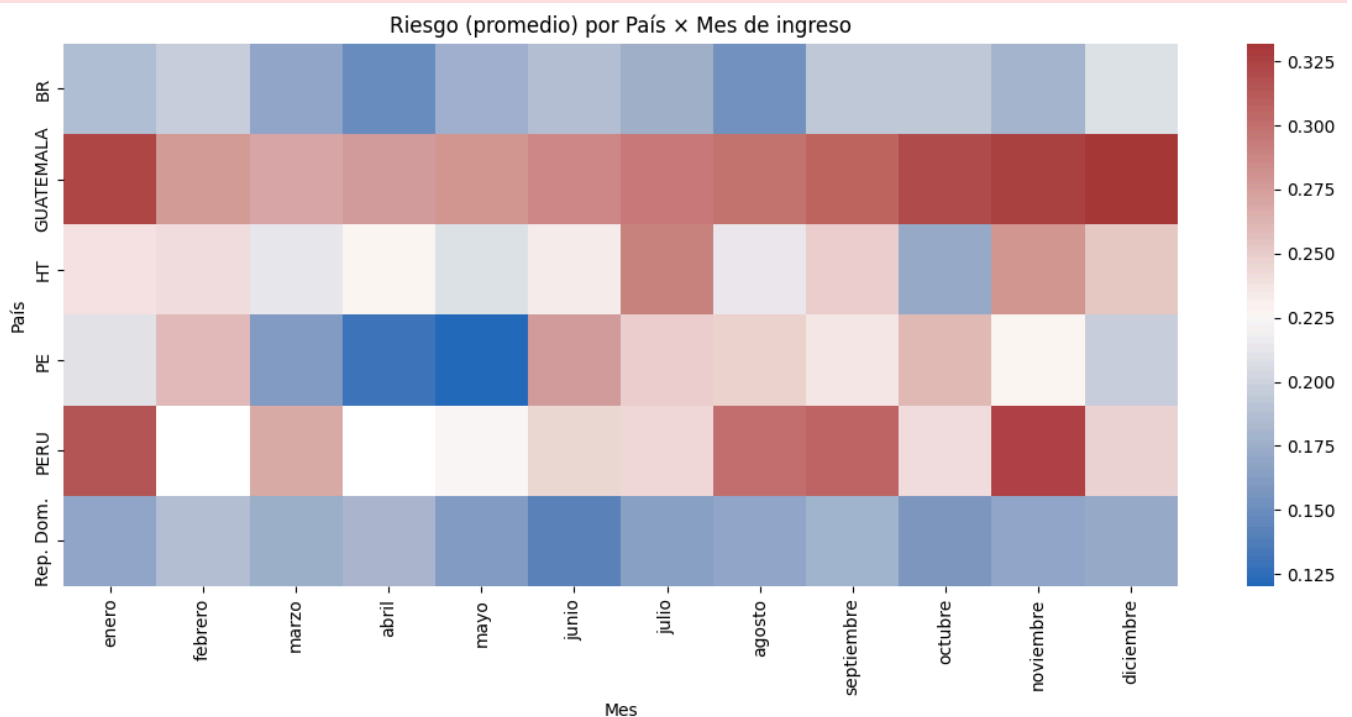
- Estos meses muestran picos de riesgo en varios grupos, especialmente en los más jóvenes.

- Podrían coincidir con ciclos de contratación, finalización de contratos o eventos estacionales que afectan la permanencia.

```
In [89]: # ejemplo: riesgo por mes y país (heatmap compacto)
rp = ACTIVOS_FEB_24.groupby(["pais", "Mes_ingreso"])["riesgo_rotacion"].mean().unstack()
plt.figure(figsize=(12,6))
sns.heatmap(rp, annot=False, cmap="vlag")
plt.title("Riesgo (promedio) por País x Mes de ingreso")
plt.xlabel("Mes")
plt.ylabel("País")
plt.tight_layout()
plt.show()
```

/var/folders/7l/8rl5yq1j1g9684gbgh80kqq80000gn/T/ipykernel\_65621/1524943446.py:2: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
rp = ACTIVOS_FEB_24.groupby(["pais", "Mes_ingreso"])["riesgo_rotacion"].mean().unstack(fill_value=0).reindex(columns=orden_meses)
```



#### Países con mayor riesgo promedio

- GUATEMALA y HT (Haití) presentan los valores más altos en varios meses, alcanzando hasta 0.325, lo que indica una mayor vulnerabilidad o inestabilidad en la permanencia de empleados.
- Estos países podrían estar enfrentando desafíos estructurales como condiciones laborales menos favorables, menor engagement o factores externos como contexto económico o político.

#### Países con riesgo intermedio

- Rep. Dom. y PE (Perú) muestran valores moderados, con fluctuaciones entre 0.20 y 0.30.
- Aunque no están en el rango más alto, sí presentan meses críticos como junio, septiembre y diciembre, donde el riesgo se intensifica.

#### Países con menor riesgo

- BR (Brasil) y PERU (posiblemente una duplicación de PE) tienen los valores más bajos, cercanos a 0.125, especialmente en los primeros meses del año.
- Esto podría reflejar mejores prácticas de onboarding, mayor estabilidad o políticas de retención más efectivas.

#### Meses críticos a nivel regional

- Junio, septiembre y diciembre son recurrentemente altos en varios países.
- Estos meses podrían coincidir con ciclos de contratación, cierres de proyectos, evaluaciones de desempeño o decisiones presupuestarias que afectan la continuidad laboral.

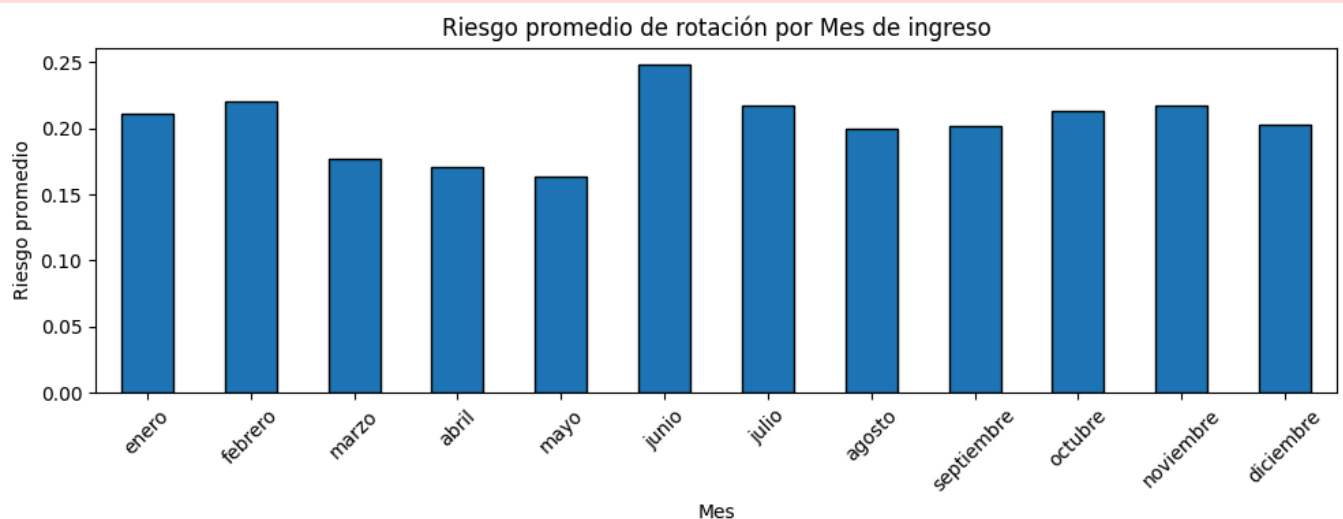
```
In [52]: # Ranking global por mes de ingreso
riesgo_por_mes = ACTIVOS_FEB_24.groupby("Mes_ingreso")["riesgo_rotacion"].mean().reindex(orden_meses)
riesgo_por_mes = riesgo_por_mes.fillna(0)

plt.figure(figsize=(10,4))
riesgo_por_mes.plot(kind="bar", edgecolor="black")
plt.title("Riesgo promedio de rotación por Mes de ingreso")
plt.ylabel("Riesgo promedio")
plt.xlabel("Mes")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Mostrar tabla ordenada (de mayor a menor riesgo)
print("Ranking meses por riesgo (mayor a menor):")
print(riesgo_por_mes.sort_values(ascending=False))
```

/var/folders/7l/8rl5yq1j1g9684gbgh80kqq80000gn/T/ipykernel\_65621/166050678.py:2: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
riesgo_por_mes = ACTIVOS_FEB_24.groupby("Mes_ingreso")["riesgo_rotacion"].mean().reindex(orden_meses)
```





Ranking meses por riesgo (mayor a menor):

Mes\_ingreso

junio	0.248475
febrero	0.219936
julio	0.216955
noviembre	0.216798
octubre	0.212911
enero	0.210901
diciembre	0.203293
septiembre	0.201788
agosto	0.200091
marzo	0.177202
abril	0.170888
mayo	0.163862

Name: riesgo\_rotacion, dtype: float64

## Conclusiones de Riesgo de Rotación

### 1. Meses con mayor riesgo de rotación

- Junio (0.248) lidera el ranking como el mes con mayor riesgo, seguido por febrero (0.220) y julio (0.217).
- Estos meses podrían coincidir con ciclos de contratación masiva, prácticas profesionales, o ingresos post-vacacionales, donde los vínculos laborales aún no están consolidados.
- También podrían reflejar contrataciones por proyectos temporales o estacionales que terminan rápidamente.

### 2. Meses de riesgo intermedio

- Noviembre, octubre, enero y diciembre se ubican en un rango de riesgo entre 0.203 y 0.217.
- Estos meses suelen estar cerca del cierre anual, lo que puede implicar ajustes de personal, finalización de contratos o decisiones estratégicas que afectan la permanencia.

### 3. Meses con menor riesgo de rotación

- Marzo, abril y mayo presentan los valores más bajos, especialmente mayo (0.164).
- Esto sugiere que los ingresos en estos meses tienden a ser más estables, posiblemente por una mejor planificación de recursos humanos o mayor alineación con objetivos anuales.



## Implicaciones estratégicas

- Para retención: reforzar el onboarding y seguimiento en junio, febrero y julio.
- Para planificación: priorizar contrataciones en marzo a mayo para mejorar la estabilidad.
- Para análisis adicional: cruzar estos datos con edad y país (como en los heatmaps anteriores) puede revelar patrones más profundos.