

# ANÁLISIS DE DATOS

CON PYTHON

2024

Daniel Pablo Kresisch



Alarmix

# ÍNDICE

**Introducción**

**01**

**Hipótesis**

**02**

**Librerías y dataset**

**03**

**Variables**

**04**

**Limpieza del dataset**

**05**

**Gráficos**

**06**

**Conclusiones**

**07**

# INTRODUCCIÓN

En este proyecto se trabajará con la base de datos Alarmix (descargada de la plataforma Kaggle) que se dedica a la venta de productos de seguridad en España.

Se analizarán los datos para transformarlos en información significativa y accionable. Además, se usará Python y las librerías como Pandas, NumPy, Seaborn y Matplotlib que permiten manipular, limpiar, analizar y visualizar los datos de manera eficiente.

En este análisis, se aprovecharán al máximo estas capacidades para obtener resultados claros y fundamentados que respalden la toma de decisiones basada en datos.

# HIPÓTESIS

La hipótesis de este trabajo sostiene que la mayoría de las ventas se concentran durante las épocas de vacaciones, principalmente en destinos costeros que los turistas eligen como lugar de descanso y recreación.

Estos lugares, debido a la alta cantidad de visitantes, experimentan un incremento significativo en la actividad comercial durante estas temporadas.

Sin embargo, es precisamente en estos momentos de mayor dinamismo turístico y comercial cuando también se registra un aumento considerable en los índices de inseguridad y miedos, afectando tanto a los residentes locales como a los turistas.

# LIBRERIAS Y DATASET

Se importaron las librerías de Python que se van utilizar, se muestra el dataset con los primero 5 registros y se agregó un código para que Python no muestre errores.

```
# Importación de las librerias
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Para que Python no muestre errores
import warnings
warnings.filterwarnings('ignore')
```

```
# Lectura del dataset y se ven los primeros 5 registros
df = pd.read_excel ("Alarmas.xlsx")
df.head()
```

	Motivo del estado	Estado de Sector Mobile	Empleado	Instalador Actual	Fecha de Reserva	Fecha de Venta	Precio de Venta	Numero de Cuenta	Instalacion	Precio de Compra	Campaña	Cartera cliente	Zip Code (Installation)	Rama Padre	ID Hardware	Nombre Hardware	Region de venta
0	Programado	Completado	105	105	2024-03-14	2024-03-17	99.0	10103858	165186	39.7	Inhouse/Direct sales	Sector Alarm HOME	38616	TE1	12001	I. Canarias-Kit Proteccion SAS	Andalucía
1	Programado	Completado	108	204	2024-03-16	2024-03-17	99.0	10103906	165249	46.9	Inhouse/Direct sales	Sector Alarm HOME	28300	MD1	10068	KIT Proteccion Plus SAS	Andalucía
2	Programado	Completado	60	60	2024-03-16	2024-03-17	99.0	10103907	165250	44.9	Inhouse/Direct sales	Sector Alarm HOME	8192	BN1	10067	Kit Proteccion SAS	Andalucía
3	Programado	Completado	99	99	2024-03-17	2024-03-17	99.0	10103908	165251	44.9	Inhouse/Direct sales	Sector Alarm HOME	29651	MA1	10068	KIT Proteccion Plus SAS	Andalucía
4	Programado	Completado	96	96	2024-03-17	2024-03-17	397.0	10103915	165258	49.9	Inhouse/Direct sales	Sector Alarm HOME	18640	GR1	10068	KIT Proteccion Plus SAS	Andalucía

# VARIABLES

Para comprender mejor los datos es necesario conocer la descripción de cada variable del dataset:

- **Motivo del estado:** Refleja si la venta fue programada o aprobada.
- **Estado de Sector Mobile:** Muestra si el pedido fue completado.
- **Empleado:** Menciona el empleado que atendió al cliente.
- **Instalador Actual:** La persona que se va encargar de la instalación de la alarma.
- **Fecha de Reserva:** La fecha cuando se reservó la compra de la alarma.
- **Fecha de Venta:** La fecha que se concretó la venta.
- **Precio de Venta:** El precio cuando se concretó la venta.
- **Número de Cuenta:** El número de cuenta del cliente.
- **Instalación:** El código de instalación.
- **Precio de Compra:** El precio del producto en el momento de compra.
- **Campaña:** La campaña que se realizó para la venta.
- **Cartera de cliente:** El tipo de cliente que compró el producto.
- **Zip Code (Installation):** El lugar donde se realizó la instalación.
- **Rama Padre:** El término técnico para identificar el tipo de producto que se vendió.
- **ID Hardware:** El número de identificación de cada producto.
- **Nombre Hardware:** El nombre de cada hardware que se vendió.
- **Región de venta:** La región de España donde son los compradores.

Se verificó qué tipo de datos es cada variable, cuántos registros y atributos existen, y cuáles son todas las variables.

```
# Tipos de datos de cada variable  
print(df.dtypes)
```

```
Motivo del estado          object  
Estado de Sector Mobile   object  
Empleado                  int64  
Instalador Actual         int64  
Fecha de Reserva          datetime64[ns]  
Fecha de Venta            datetime64[ns]  
Precio de Venta           float64  
Numero de Cuenta          int64  
Instalacion                int64  
Precio de Compra           float64  
Campaña                   object  
Cartera de cliente        object  
Zip Code (Installation)   int64  
Rama Padre                 object  
Nombre Hardware            object  
Region de venta            object  
dtype: object
```

```
# Registros (filas) y atributos (columnas) que presenta el dataset
```

```
df.shape
```

```
(10567, 16)
```

```
# Variables del dataset
```

```
df.columns
```

```
Index(['Motivo del estado', 'Estado de Sector Mobile', 'Empleado',  
       'Instalador Actual', 'Fecha de Reserva', 'Fecha de Venta',  
       'Precio de Venta', 'Numero de Cuenta', 'Instalacion',  
       'Precio de Compra', 'Campaña', 'Cartera de cliente',  
       'Zip Code (Installation)', 'Rama Padre', 'Nombre Hardware',  
       'Region de venta'],  
      dtype='object')
```

# LIMPIEZA DEL DATASET

Se convirtió una variable en índice, se verificó que no haya datos nulos y se mostró las estadísticas descriptivas de las variables numéricas.

```
# Verificación de valores nulos
print(df.isnull().sum())
```

```
Motivo del estado      0
Estado de Sector Mobile 0
Empleado               0
Instalador Actual       0
Fecha de Reserva        0
Fecha de Venta          0
Precio de Venta         0
Numero de Cuenta        0
Instalacion              0
Precio de Compra         0
Campaña                 0
Cartera de cliente       0
Zip Code (Installation) 0
Rama Padre               0
Nombre Hardware          0
Region de venta          0
dtype: int64
```

```
# Estadísticas descriptivas de las variables numéricas
print(df.describe(include=[np.number]))
```

	Empleado	Instalador Actual	Precio de Venta	Numero de Cuenta
count	10567.000000	10567.000000	10567.000000	1.056700e+04
mean	174.971610	159.893536	213.337438	1.009509e+07
std	141.356578	136.124191	186.738968	6.950715e+03
min	1.000000	1.000000	0.000000	1.000028e+07
25%	65.000000	60.000000	99.000000	1.009069e+07
50%	133.000000	125.000000	149.000000	1.009609e+07
75%	251.000000	227.000000	250.000000	1.010038e+07
max	675.000000	734.000000	2435.000000	1.010392e+07

	Instalacion	Precio de Compra	Zip Code (Installation)
count	10567.000000	10567.000000	10567.000000
mean	154430.475159	45.308872	25377.751396
std	7071.412450	4.861179	10635.198029
min	123926.000000	0.000000	3015.000000
25%	148568.500000	42.000000	18100.000000
50%	155261.000000	44.900000	29570.000000
75%	160740.500000	49.900000	30107.000000
max	165258.000000	88.400000	50420.000000

```
# Conversión de una columna en index
df = df.set_index('ID Hardware')
df
```

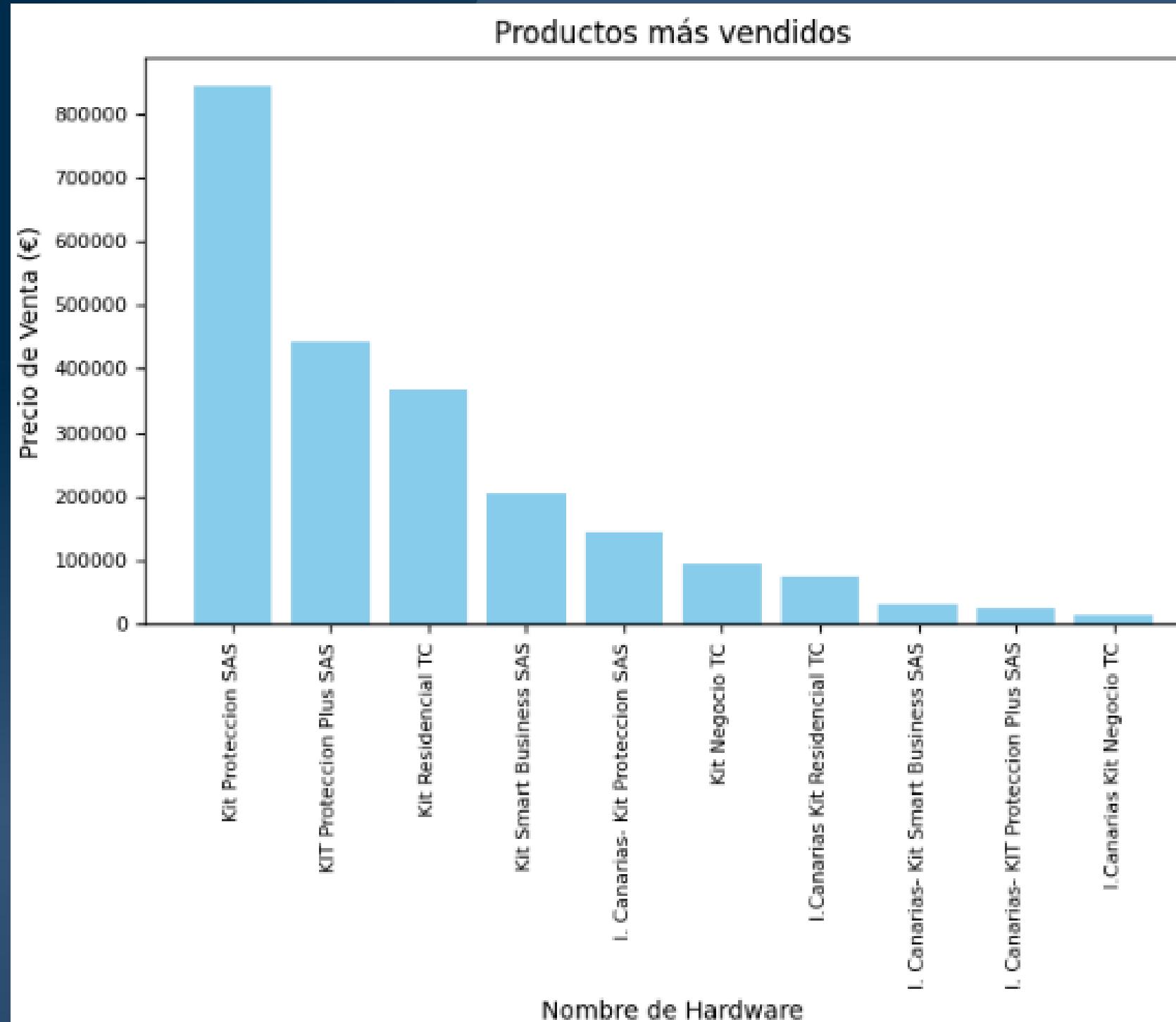
ID Hardware	Motivo del estado	Estado de Sector Mobile	Empleado	Instalador Actual	Fecha de Reserva	Fecha de Venta	Precio de Venta	Numero de Cuenta	Instalacion	Precio de Compra	Campaña	Cartera de cliente	Zip Code (Installation)	Rama Padre	Nombre Hardware	Region de venta
12001	Programado	Completado	105	105	2024-03-14	2024-03-17	99.0	10103858	165186	39.7	Inhouse/Direct sales	Sector Alarm HOME	38616	TE1	I. Canarias- Kit Proteccion SAS	Andalucía
10068	Programado	Completado	108	204	2024-03-16	2024-03-17	99.0	10103906	165249	46.9	Inhouse/Direct sales	Sector Alarm HOME	28300	MD1	KIT Proteccion Plus SAS	Andalucía
10067	Programado	Completado	60	60	2024-03-16	2024-03-17	99.0	10103907	165250	44.9	Inhouse/Direct sales	Sector Alarm HOME	8192	BN1	Kit Proteccion SAS	Andalucía
10068	Programado	Completado	99	99	2024-03-17	2024-03-17	99.0	10103908	165251	44.9	Inhouse/Direct sales	Sector Alarm HOME	29651	MA1	KIT Proteccion Plus SAS	Andalucía
10068	Programado	Completado	96	96	2024-03-17	2024-03-17	397.0	10103915	165258	49.9	Inhouse/Direct sales	Sector Alarm HOME	18640	GR1	KIT Proteccion Plus SAS	Andalucía
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11000	Aprobado	Completado	140	140	2022-01-02	2022-01-03	299.0	10084721	140851	39.9	Direct Sales	Sector Alarm HOME	29620	MA1	Kit Residencial TC	Islas Baleares

# GRÁFICOS

Para comprender mejor los datos se realizaron 7 gráficos con Python, utilizando las librerías Matplotlib y Seaborn.



# PRODUCTOS MÁS VENDIDOS



Se observa que el producto más vendido es Kit Protección SAS, seguido por Kit Protección Plus SAS y Kit Residencial TC, por lo que sería bueno potenciar las ventas de estos últimos dos para que puedan superar o igualarse al producto estrella.

```
# Filtro de las columnas que se van a utilizar en el gráfico
df_filtered = df[['Nombre Hardware', 'Precio de Venta']]

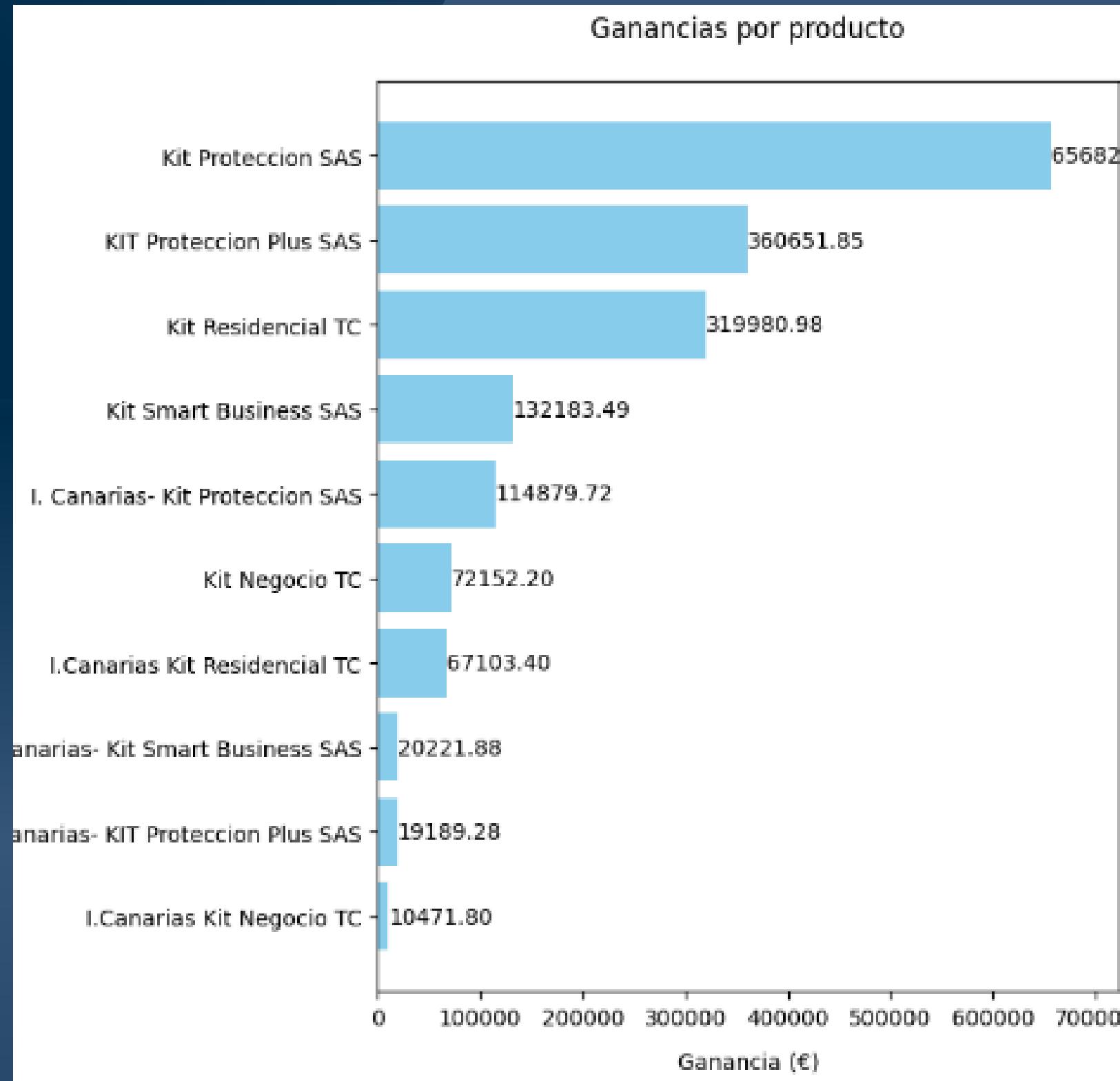
# Se agrupa por 'Nombre Hardware' y se suma los precios (si hay duplicados), luego se ordena de mayor a menor
df_grouped = df_filtered.groupby('Nombre Hardware').sum().sort_values(by='Precio de Venta', ascending=False)

# Selección el top 10
df_top10 = df_grouped.head(10)

# Creación del gráfico con sus ajustes
plt.figure(figsize=(7, 6))
plt.bar(df_top10.index, df_top10['Precio de Venta'], color='skyblue')
plt.xticks(rotation=90, fontsize=8)
plt.yticks(fontsize=8)
plt.title('Productos más vendidos', fontsize=12)
plt.xlabel('Nombre de Hardware', fontsize=10)
plt.ylabel('Precio de Venta (€)', fontsize=10)

# Se muestra el gráfico
plt.tight_layout()
plt.show()
```

# GANANCIAS POR PRODUCTO



En este gráfico se observan las ganancias que hubo por cada producto, lo que es clave para la toma de decisiones y para evaluar el precio de venta y compra de cada producto.

```
# Cálculo de la ganancia
df['Ganancia'] = df['Precio de Venta'] - df['Precio de Compra']

# Filtro del top 10 de nombre hardware por ganancia
top_10_hardware = df.groupby('Nombre Hardware')['Ganancia'].sum().nlargest(10)

# Creación de un DataFrame para el top 10
df_top10 = top_10_hardware.reset_index()

# Se ordena el DataFrame de mayor a menor
df_top10 = df_top10.sort_values(by='Ganancia', ascending=True)

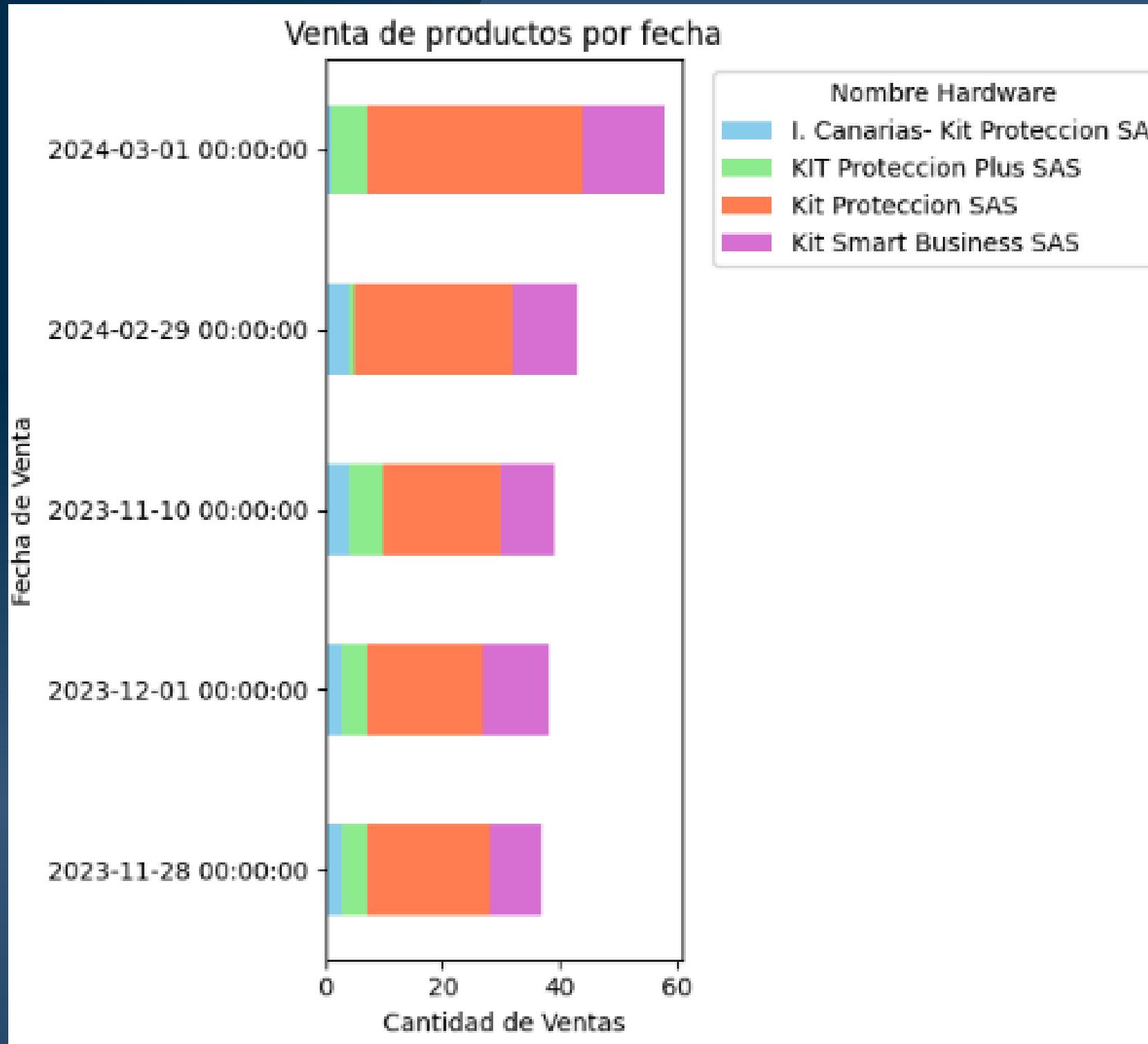
# Creación de un gráfico de barras para ver la ganancia
plt.figure(figsize=(8, 7))
bars = plt.barh(df_top10['Nombre Hardware'], df_top10['Ganancia'], color='skyblue')

# Ajustes del gráfico
plt.title('Ganancias por producto', fontsize=12, pad=20)
plt.xlabel('Ganancia (€)', fontsize=10, labelpad=10)
plt.ylabel('Nombre de Hardware', fontsize=10, labelpad=10)
plt.xlim(0, df_top10['Ganancia'].max() * 1.1)

# Valores en las barras
for bar in bars:
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2, f'{bar.get_width():.2f}', va='center', fontsize=10)

# Se muestra el gráfico
plt.tight_layout()
plt.show()
```

# VENTA DE PRODUCTOS POR FECHA



Se observan los productos más vendidos durante la época de mayor ventas, y el producto estrella fue el más vendido. En relación a las fechas, se identificó que las mayores ventas fueron en febrero y marzo (fecha de mayor turismo) y luego, previo a las fiestas de diciembre.

```
## Filtro de las columnas que se van utilizar
nombre_hardware = df['Nombre Hardware']
fecha_venta = pd.to_datetime(df['Fecha de Venta'])

## Top 5 de hardware más vendidos
top5.hardware = df.groupby('Nombre Hardware').size().nlargest(5)

## Filtro del dataframe con el top 5 hardware
df_top5.hardware = df[df['Nombre Hardware'].isin(top5.hardware.index)]

## Top 5 de fechas con más ventas
top5_fechas = df_top5.hardware.groupby('Fecha de Venta').size().nlargest(5)

## Filtro del dataframe con las 5 fechas más importantes
df_top5_fechas = df_top5.hardware[df_top5.hardware['Fecha de Venta'].isin(top5_fechas.index)]

## Se agrupa por 'Fecha de Venta' y 'Nombre Hardware' para sumar las ventas de cada hardware por fecha
ventas_agrupadas = df_top5_fechas.groupby(['Fecha de Venta', 'Nombre Hardware']).size().unstack(fill_value=0)

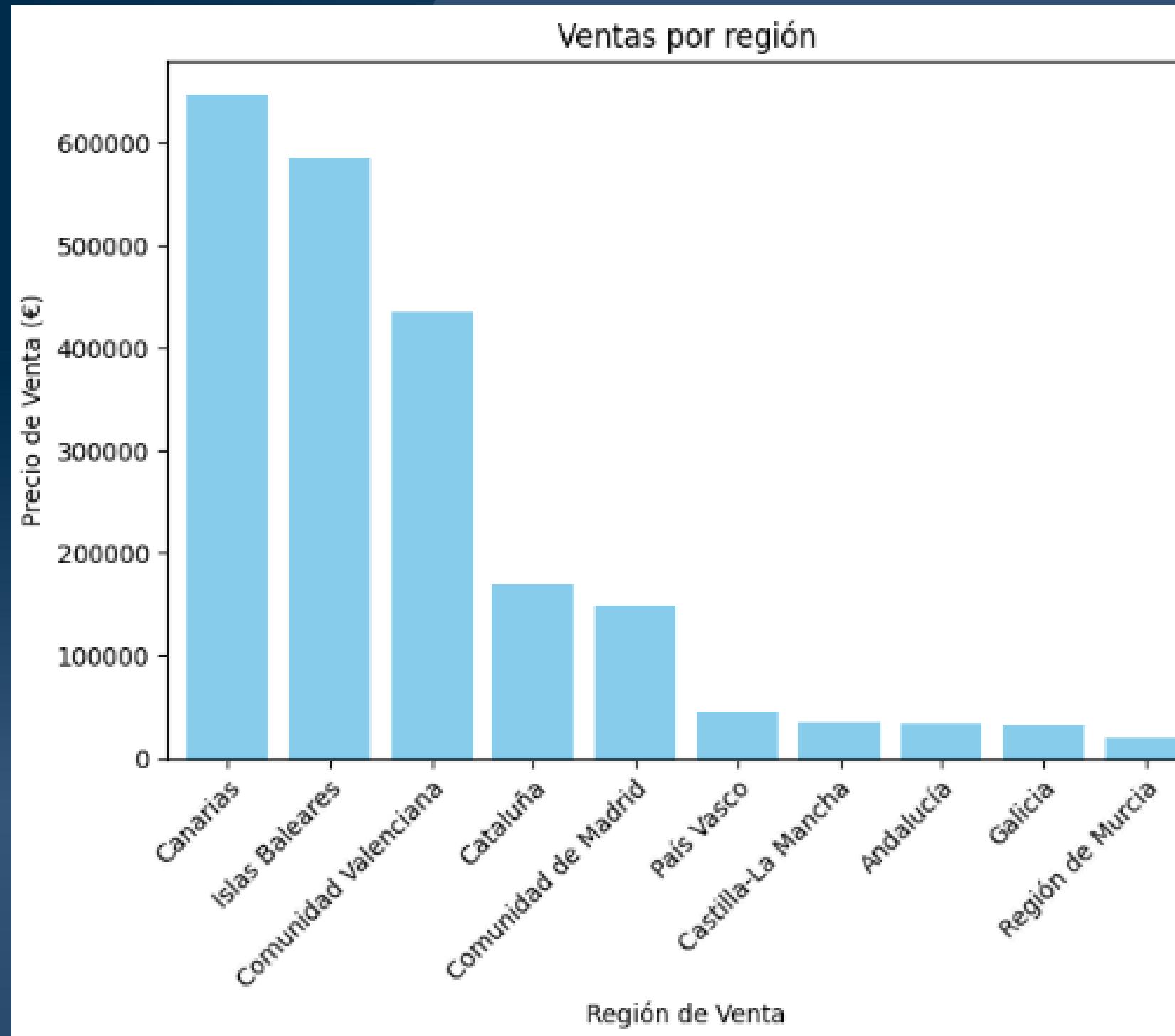
## Suma de las ventas por cada fecha y se ordena por la cantidad ventas
ventas_agrupadas['Total Ventas'] = ventas_agrupadas.sum(axis=1)
ventas_agrupadas = ventas_agrupadas.sort_values(by='Total Ventas', ascending=True)

## Se elimina la columna de 'Total Ventas' antes de graficar
ventas_agrupadas = ventas_agrupadas.drop(columns=['Total Ventas'])

## Creación del gráfico de barras apiladas (horizontal) junto con los ajustes
ventas_agrupadas.plot(kind='barh', stacked=True, figsize=(7, 6), color=['skyblue', 'lightgreen', 'coral', 'orchid', 'lightgrey'])
plt.title('Venta de productos por fecha', fontsize=12)
plt.xlabel('Cantidad de Ventas', fontsize=10)
plt.ylabel('Fecha de Venta', fontsize=10)
plt.legend(title='Nombre Hardware', bbox_to_anchor=(1.05, 1), loc='upper left')

## Se muestra el gráfico
plt.tight_layout()
plt.show()
```

# VENTAS POR REGIÓN



En este gráfico se observa que las regiones donde hubo mayores ventas fueron los lugares para vacacionar (Canarias, Islas Baleares y Comunidad Valenciana), y las dos grandes regiones (Cataluña y Madrid).

```
# Filtro de las columnas que se van a utilizar en el gráfico
df_filtered = df[['Region de venta', 'Precio de Venta']]

# Se agrupa por 'Region de ventas' y se suman los precios (si hay duplicados), luego se ordena de mayor a menor
df_grouped = df_filtered.groupby('Region de venta').sum().sort_values(by='Precio de Venta', ascending=False)

# Selección del top 10
df_top10 = df_grouped.head(10)

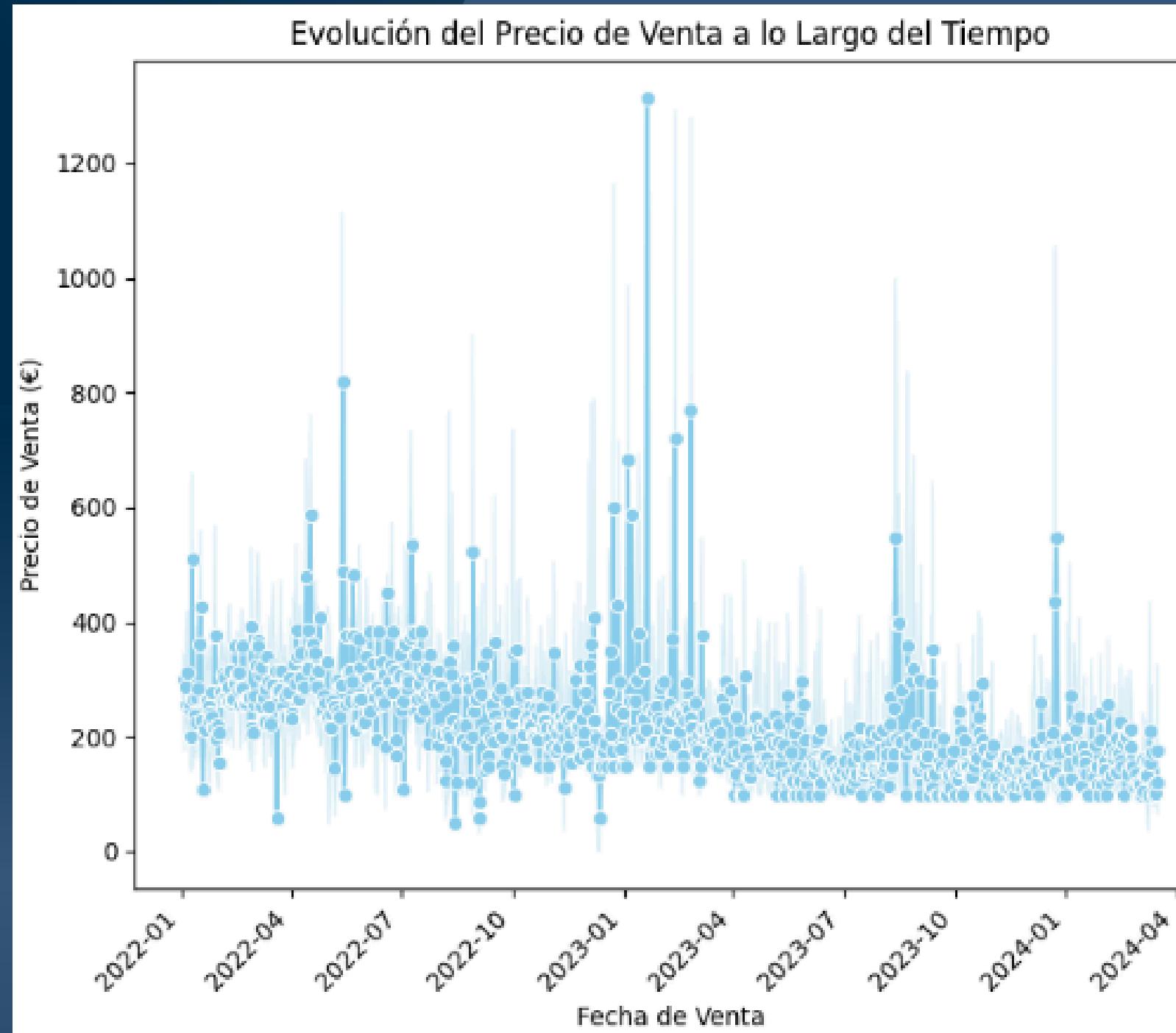
# Creación del gráfico de barras
plt.figure(figsize=(7, 6))
plt.bar(df_top10.index, df_top10['Precio de Venta'], color='skyblue')

# Ajustes del eje X
plt.xticks(rotation=45, ha='right', fontsize=10, rotation_mode='anchor')
plt.gca().margins(x=0.02)
plt.yticks(fontsize=10)

# Personalización del gráfico
plt.title('Ventas por región', fontsize=12)
plt.xlabel('Región de Venta', fontsize=10)
plt.ylabel('Precio de Venta (€)', fontsize=10)

# Se muestra el gráfico
plt.tight_layout()
plt.show()
```

# PRECIO DE VENTA A LO LARGO DEL TIEMPO



Se muestra la evolución histórica del precio de venta en relación a las fechas. Se puede observar que la mayoría de los precios varían entre 100 y 400 Euros.

```
# Confirmación que la columna de fechas esté con en formato correcto
df['Fecha de Venta'] = pd.to_datetime(df['Fecha de Venta'], errors='coerce')

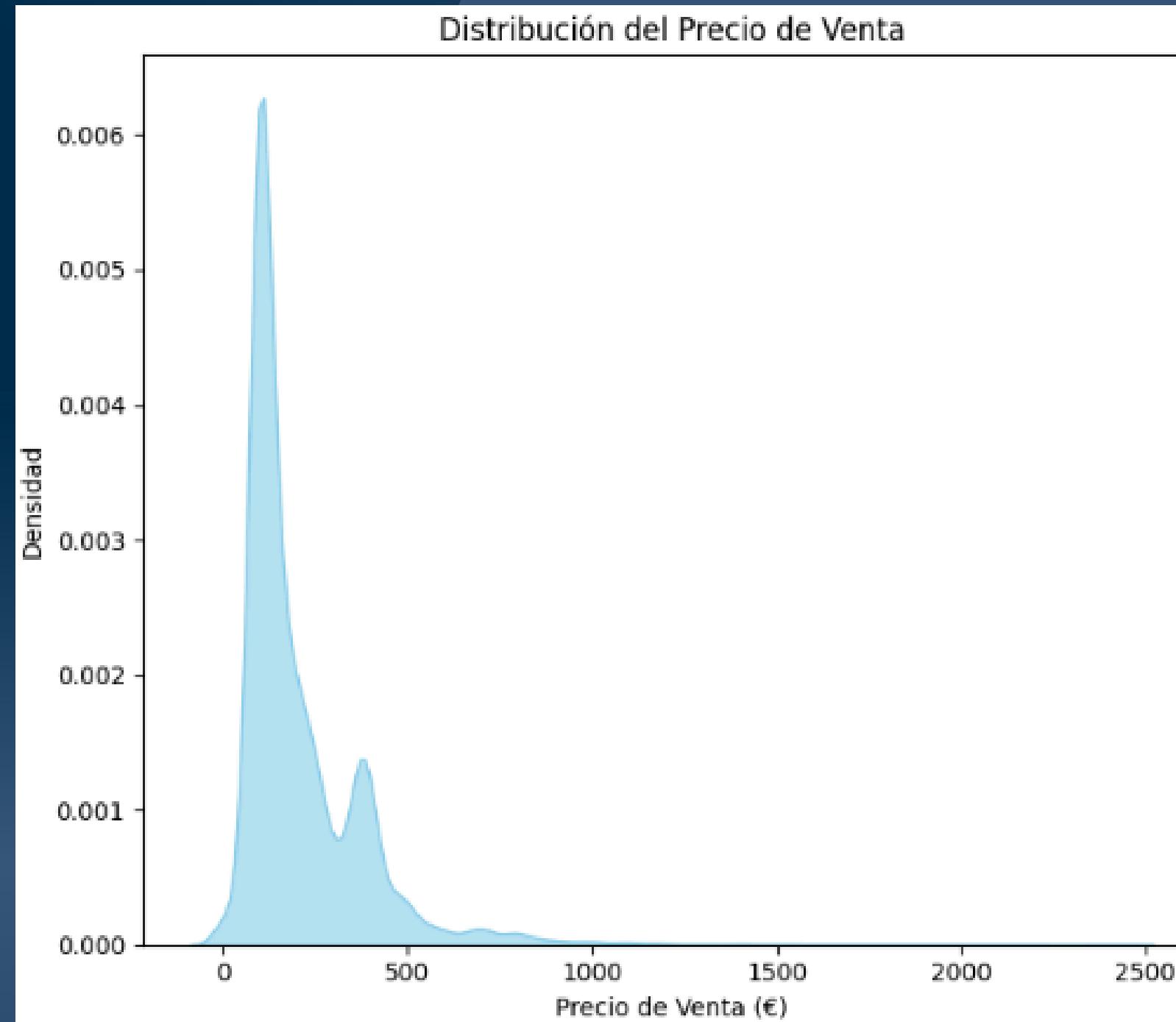
# Se ordenan los datos por fecha para ver la evolución en el tiempo
df_sorted = df.sort_values(by='Fecha de Venta')

# Creación del gráfico de líneas con Seaborn
plt.figure(figsize=(7, 6))
sns.lineplot(x='Fecha de Venta', y='Precio de Venta', data=df_sorted, marker='o', color='skyblue')

# Ajustes del gráfico
plt.title('Evolución del Precio de Venta a lo Largo del Tiempo', fontsize=12)
plt.xlabel('Fecha de Venta', fontsize=10)
plt.ylabel('Precio de Venta (€)', fontsize=10)

# Se muestra el gráfico
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

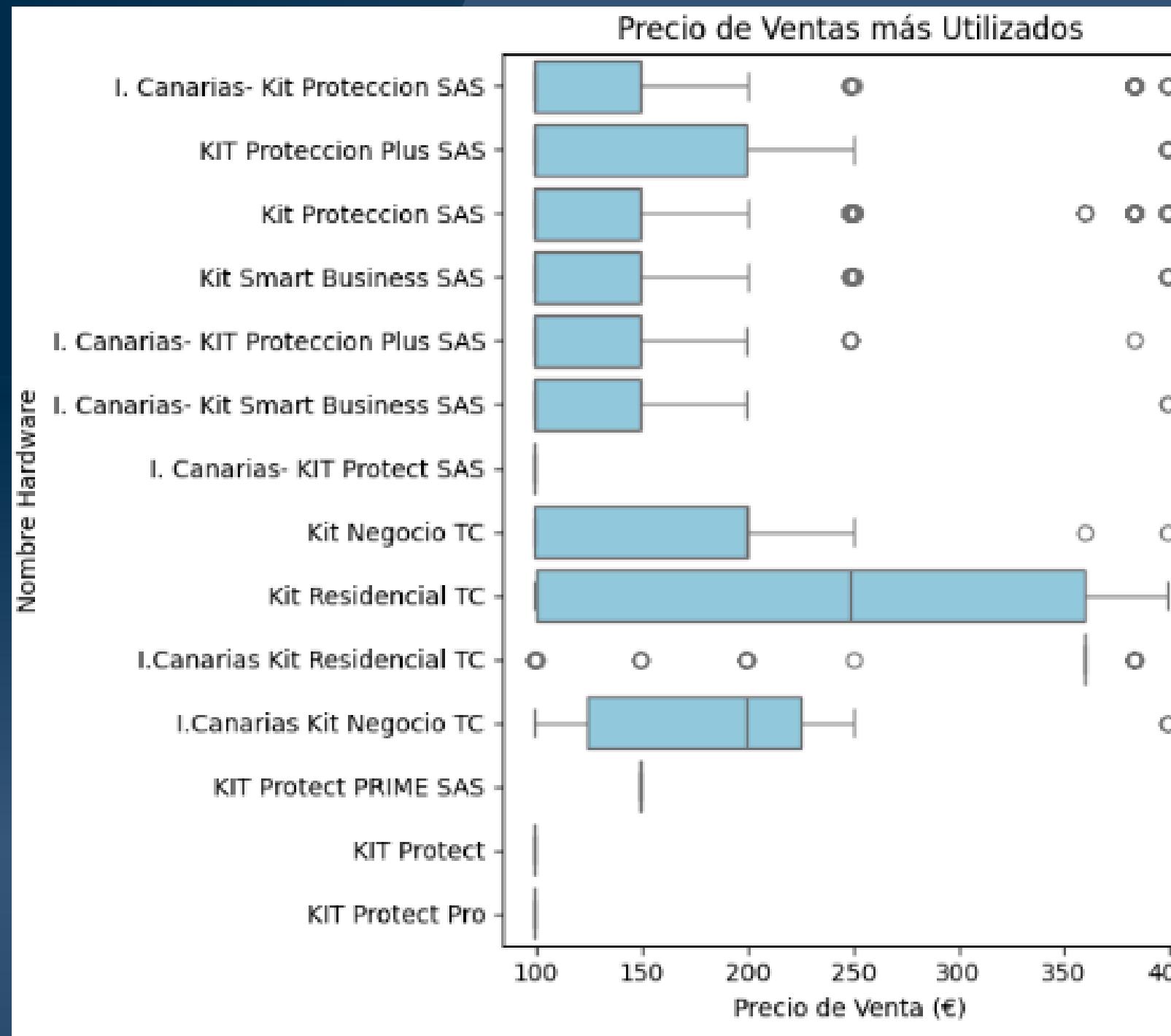
# DISTRIBUCIÓN DEL PRECIO DE VENTA



En este gráfico se observa el rango de precio de las mayores ventas que, al igual que el gráfico anterior, varía entre los 100 y 400 Euros.

```
# Creación del gráfico de densidad para la columna 'Precio de Venta'  
plt.figure(figsize=(7, 6))  
sns.kdeplot(data=df, x='Precio de Venta', fill=True, color='skyblue', alpha=0.6)  
  
# Ajustes del gráfico  
plt.title('Distribución del Precio de Venta', fontsize=12)  
plt.xlabel('Precio de Venta (€)', fontsize=10)  
plt.ylabel('Densidad', fontsize=10)  
  
# Se muestra el gráfico  
plt.tight_layout()  
plt.show()
```

# PRECIO DE VENTAS MÁS UTILIZADOS



Se utilizó este gráfico para ver la comparación entre el rango de precio que más se vendió y los productos más vendidos. Con el boxplot se puede observar cuáles son la media, mediana y moda para identificar los precios que suelen comprar los clientes.

```
# Filtro con los 10 precios más utilizados
top_10_used_prices = df['Precio de Venta'].value_counts().nlargest(10).index
df_filtered_prices = df[df['Precio de Venta'].isin(top_10_used_prices)]

# Creación de un boxplot para la columna 'Precio de Venta'
plt.figure(figsize=(7, 6))
sns.boxplot(data=df_filtered_prices, x='Precio de Venta', y='Nombre Hardware', color='skyblue')

# Personalización del gráfico
plt.title('Precio de Ventas más Utilizados', fontsize=12)
plt.xlabel('Precio de Venta (€)', fontsize=10)
plt.ylabel('Nombre Hardware', fontsize=10)

# Se muestra el gráfico
plt.tight_layout()
plt.show()
```

# CONCLUSIONES

Con toda la exploración y el análisis de los datos y los gráficos, se identificó que existen dos períodos de mayor venta:

- Entre febrero y marzo: debido a las vacaciones de los países debajo del Ecuador, como en el caso de Sudamérica que suelen viajar para esa época a los lugares turísticos.
- Entre noviembre y principios de diciembre: debido a las festividades.

Además, se descubrió que los lugares con mayores ventas son justamente los destinos que se eligen para ir de vacaciones.

En conclusión, sería bueno enfocarse y reforzar la presencia en los lugares turísticos que la gente utiliza para irse de vacaciones, ya que es el momento donde las personas quieren sentirse más seguras fuera de su hogar.

# MUCHAS GRACIAS



[danielkresisch.com.ar](http://danielkresisch.com.ar)



[danielkresisch@gmail.com](mailto:danielkresisch@gmail.com)



[linkedin.com/in/daniel-kresisch/](https://linkedin.com/in/daniel-kresisch/)



[github.com/danykre](https://github.com/danykre)

