1. Google Play Store apps and reviews

Las aplicaciones móviles están en todas partes. Son fáciles de crear y pueden resultar muy lucrativas. Debido a estos dos factores, se están desarrollando cada vez más aplicaciones. En este ejercicio, haremos un análisis completo del mercado de aplicaciones de Android comparando más de diez mil aplicaciones en Google Play en diferentes categorías. Buscaremos información valiosa en los datos para diseñar estrategias que impulsen el crecimiento y la retención.

Tenemos dos fuentes de datos:

- `apps.csv` : contiene todos los detalles de las aplicaciones en Google Play. Hay 13 características que describen una aplicación determinada.
- `user_reviews.csv` : contiene 100 reseñas para cada aplicación, reviews. El texto de cada reseña se ha procesado previamente y se le atribuyen tres características nuevas: Sentimiento (positivo, negativo o neutral), Polaridad del sentimiento y Subjetividad del sentimiento..

In [57]:
```
# Importa las librerías de pandas y matplotlib
import pandas as pd
import matplotlib.pyplot as plt

# Importa el datase apps.csv
apps = pd.read_csv('C:/Users/Isaac/Desktop/IHD/EBAC DT/CIENCIA DE DATOS/M25 DS/Avance de Proyecto parte 1 y 2/apps.csv')
apps.head()
```

Out[57]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool	ART_AND_DESIGN	4.7	87510	8.7	5,000,000+	Free	0	Everyone	Art & Design	August 1,	1.2.4	4.0.3 and

In [59]:
```
# Borra todos los duplicados del dataset
apps_dupli = len(apps) - len(apps.drop_duplicates())
apps_dupli
### No hay datos duplicados ###
```

Out[59]: 0

In [60]:
```
duplicados = apps.duplicated()
print('valores duplicados', apps[duplicados])
```

```
valores duplicados Empty DataFrame
Columns: [App, Category, Rating, Reviews, Size, Installs, Type, Price, Content Rating, Genres, Last Updated, Current Ver, Andro
id Ver]
Index: []
```

In [61]:
```
# Imprime el total de apps que se van a analizar que quedan (dimensión)
#print(apps['App'])
apps_total = len(apps['App'])
print('Total number of apps in the dataset = ', apps_total)
```

```
Total number of apps in the dataset =  9659
```

In [62]:
```
# Imprime la estádistica descriptiva de resumen
print(apps.describe())
```

```
            Rating       Reviews         Size
count  8196.000000  9.659000e+03  8432.000000
mean      4.173243  2.165926e+05    20.395327
std       0.536625  1.831320e+06    21.827509
min       1.000000  0.000000e+00     0.000000
```

In [63]:
```
# Vamos a echar un vistazo al DataFrame final

apps
```

Out[63]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.0	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art	ART_AND_DESIGN	4.3	967	2.8	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

2. Data cleaning

Las cuatro variables con las que trabajaremos con más frecuencia de ahora en adelante son *Installs*, *Size*, *Rating* y *Price*. La función `info()` nos dice que las columnas *Installs* y *Price* son de tipo `object`, no son de tipo `int` o `float` como esperaríamos. Esto se debe a que la columna contiene algunos caracteres más que solo [0,9] dígitos. Idealmente, queremos que estas columnas fueran puramente numéricas

Por lo tanto, ahora necesitamos limpiar nuestros datos. Específicamente, los caracteres especiales `,` y `+` que se encuentran en la columna *Installs* y `$` que esta en la columna *Price*.

Aquí un link donde podrás ver un poco más a detalle que es una función lambda

In [64]:
```
# Lista de caracteres a eliminar
chars_to_remove = [',','+','$']
# Lista de las columnas a limpiar
cols_to_clean = ['Installs','Price']

# Loop para cada columna
for col in cols_to_clean:
    # Loop para cada caracter especial
    for char in chars_to_remove:
        # Reemplaza con una función lambda el caracter especial por un texto vacío ('')
        apps[col] = apps[col].apply(lambda x: x.replace(char,''))
    # Convierte la columna a tipo flotante (float)
    apps[col] = apps[col].astype(float)
```

In [100]:
```
apps.head(10)
```

Out[100]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10000.0	Free	0.0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000.0	Free	0.0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5000000.0	Free	0.0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.0	50000000.0	Free	0.0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8	100000.0	Free	0.0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up
5	Paper flowers instructions	ART_AND_DESIGN	4.4	167	5.6	50000.0	Free	0.0	Everyone	Art & Design	March 26, 2017	1	2.3 and up
6	Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	3.8	178	19.0	50000.0	Free	0.0	Everyone	Art & Design	April 26, 2018	1.1	4.0.3 and up
7	Infinite Painter	ART_AND_DESIGN	4.1	36815	29.0	1000000.0	Free	0.0	Everyone	Art & Design	June 14, 2018	6.1.61.1	4.2 and up

3. Exploring App's categories

Con más de mil millones de usuarios activos en 190 países de todo el mundo, Google Play sigue siendo una importante plataforma de distribución para crear una audiencia global. Para que las empresas muestren sus aplicaciones a los usuarios, es importante hacerlas más rápida y fácilmente visibles en Google Play. Para mejorar la experiencia de búsqueda general, Google ha introducido el concepto de agrupar aplicaciones en categorías.

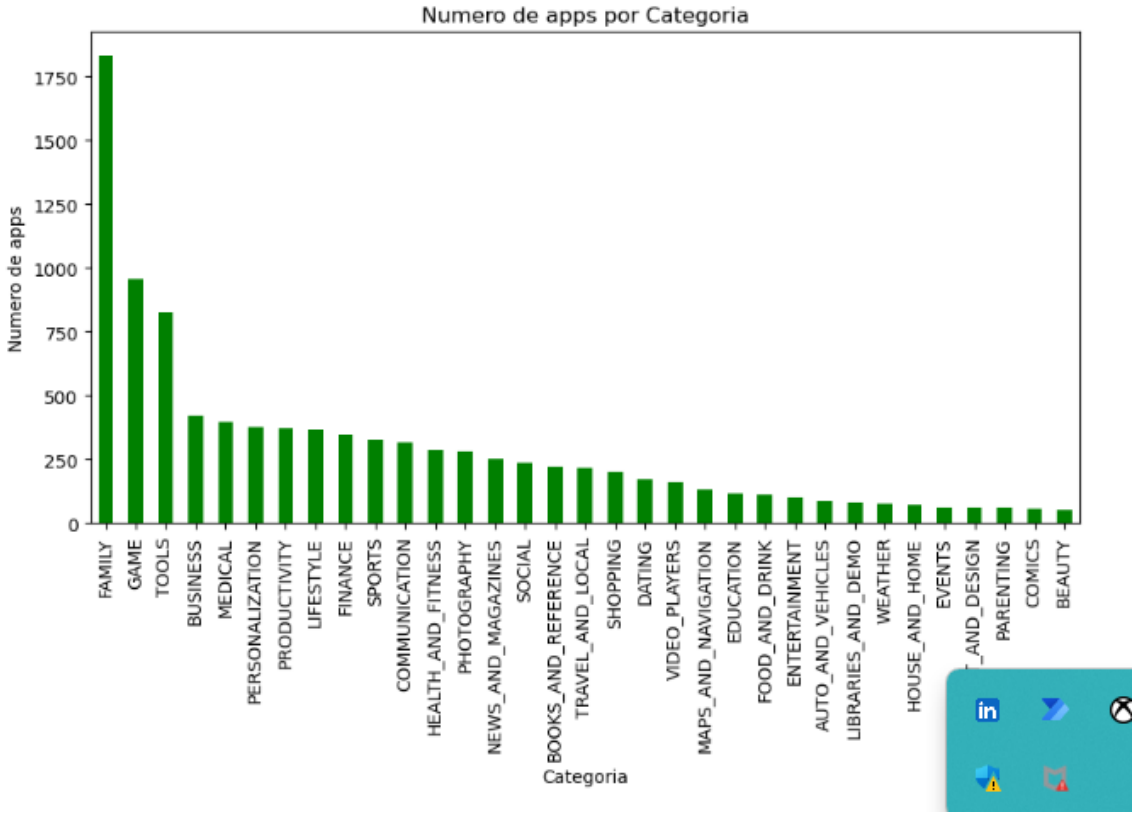
Esto nos lleva a las siguientes preguntas:

- ¿Qué categoría tiene la mayor participación de aplicaciones (activas) en el mercado?
- ¿Alguna categoría específica domina el mercado?
- ¿Qué categorías tienen la menor cantidad de aplicaciones?

Vamos a responder estas preguntas aquí `33` categorías unicas estan presentas en nuestro dataset. Las apps de *Family* y *Game* tienen la mayor prevalencia del mercado. Curiosamente, *Tools*, *Business* y *Medical* también están en el top.

In [80]:
```
# Imprime el total de categorías únicas
num_categories = apps['Category'].nunique()
print('Number of categories = ', num_categories)

# Cuenta el número de aplicaciones en cada Categoría y ordena de manera descendente
num_apps_in_category = apps['Category'].value_counts()

# Muestra el resultado en una gráfica de barras
num_apps_in_category.plot(kind = 'bar', figsize = (10, 5), color = 'green')
plt.xlabel('Categoria')
plt.ylabel('Numero de apps')
plt.title('Numero de apps por Categoria')
plt.show()
```

```
Number of categories =  33
```

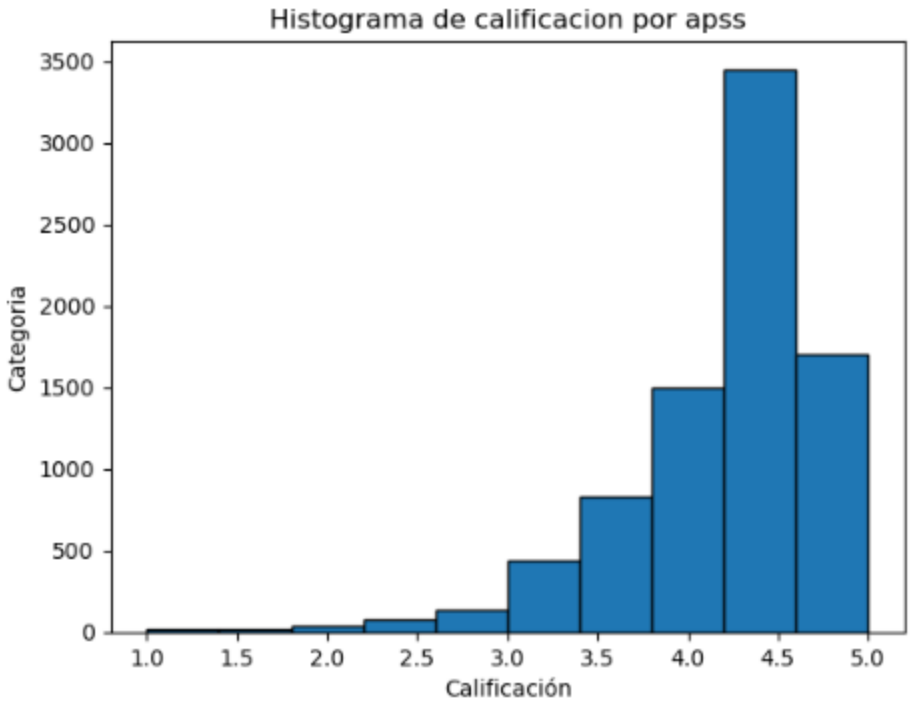


4. Ratings Distribution

Después de analizar la participación de mercado para cada categoría de las aplicaciones, veamos cómo se posicionan de acuerdo a las calificaciones (en una escala del 1 al 5) las cuales afectan la imagen de la marca general de la empresa. Las calificaciones son un indicador clave de rendimiento de una aplicación.

In [98]:
```
# Calcular el promedio de calificación de las apps
avg_app_rating = apps['Rating'].mean()
print('Average app rating = ', avg_app_rating)

# Calcula el promedio de calificación por categoría
prom_apps_in_category = apps.groupby('Category')['Rating'].mean()

# Visualiza en un histograma el comportamiento del Rating
plt.hist(apps['Rating'], bins = 10, edgecolor = 'black')
plt.title('Histograma de calificacion por apss')
plt.xlabel('Calificación')
plt.ylabel('Categoria')
plt.show()
```

```
Average app rating =  4.173243045387994
```



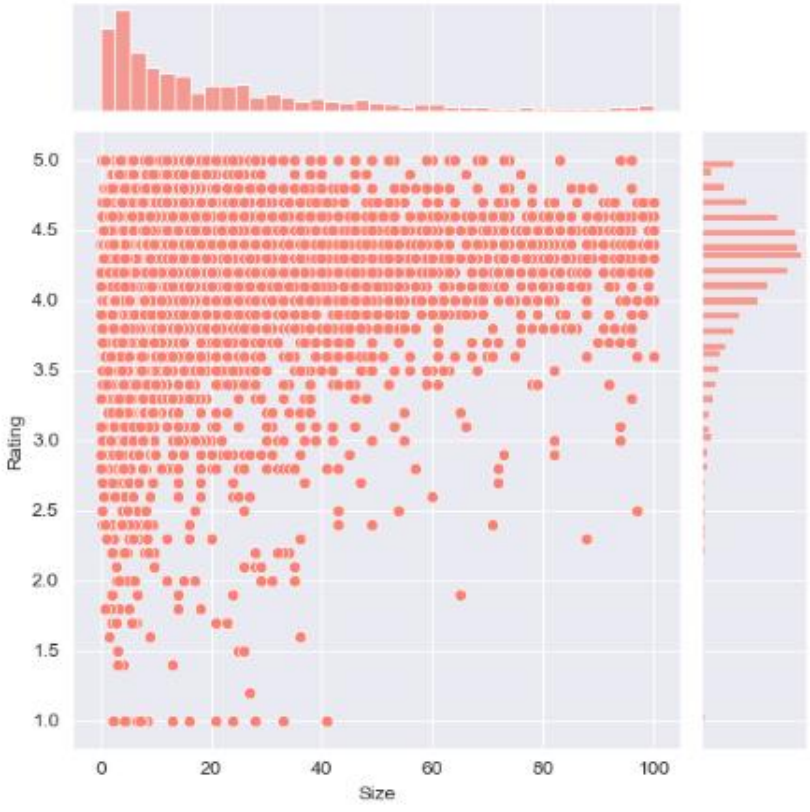
5. Size and Price

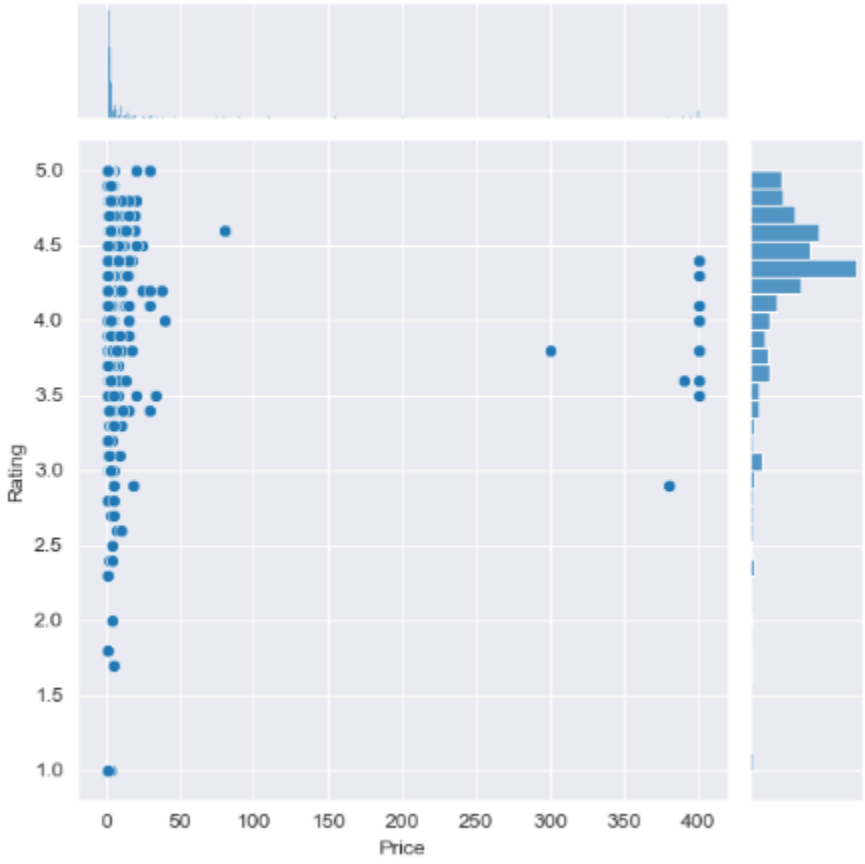
Examinemos ahora el tamaño y el precio de la aplicación. En cuanto al tamaño, si la aplicación móvil es demasiado grande, puede ser difícil y/o costoso p
los usuarios descargarla. Los tiempos de descarga prolongados pueden desanimar a los usuarios incluso antes de que experimenten su aplicación móvil.
Además, el dispositivo de cada usuario tiene una cantidad limitada de espacio en disco. Por el precio, algunos usuarios esperan que sus aplicaciones sea
gratuitas o económicas. Estos problemas se agravan si el mercado objetivo es en países en vías de desarrollo; especialmente debido a las velocidades d
Internet, el poder adquisitivo, los tipos de cambio, etc.

How can we effectively come up with strategies to size and price our app?

- ¿El tamaño de una aplicación afecta su calificación?
- ¿Los usuarios realmente se preocupan por las aplicaciones pesadas del sistema o prefieren las aplicaciones ligeras?
- ¿El precio de una aplicación afecta su calificación?
- ¿Los usuarios siempre prefieren las aplicaciones gratuitas a las de paga?

In [126]:
```
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")

sns.set_style("darkgrid")

# Filtra filas donde los valores de 'Rating' y 'Size' no sean nulos
apps_with_size_and_rating_present = apps['Rating'].notna() & apps['Size'].notna()

# Filtra las categorías con al menos 250 apps
large_categories =apps.groupby('Category').filter(lambda x: len(x) >= 250)

# Gráfica size vs. rating
plt1 = sns.jointplot(x = large_categories['Size'], y = large_categories['Rating'], color = 'salmon')

# Selecciona las apps de paga 'Type' = 'Paid'
paid_apps = apps[apps['Type'] == 'Paid']


# Grafica price vs. rating de las paplicaciones de paga
plt2 = sns.jointplot(x = paid_apps['Price'], y = paid_apps['Rating'])
```





6. Relation between Category & Price

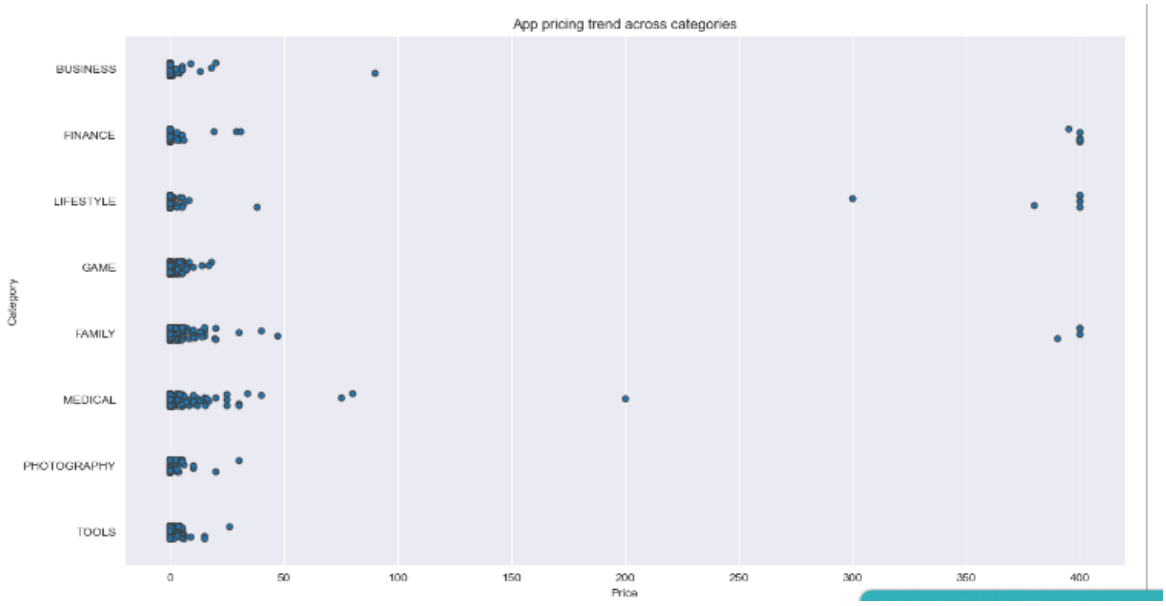
Así que ahora viene la parte difícil. ¿Cómo se supone que las empresas y los desarrolladores cubran sus cuotas de fin de mes? ¿Qué estrategias de monetización pueden utilizar las empresas para maximizar las ganancias? Los costos de las aplicaciones se basan en gran medida en las características, la complejidad y la plataforma. Hay muchos factores a considerar al seleccionar la estrategia de precios adecuada para las aplicaciones moviles. Es importante considerar la disposición de su cliente a pagar por la aplicación. Un precio elevado puede hacer que los clientes no se vean atraídos por descargarlaque ocurra la descarga o pueden eliminar una aplicación que han descargado después de recibir demasiados anuncios o simplemente no obtener el valor que esperaban de su dinero.

Las diferentes categorías exigen diferentes rangos de precios. Algunas aplicaciones que son simples y se usan a diario, como la aplicación de calculadora, probablemente deberían mantenerse gratuitas. Sin embargo, tendría sentido cobrar por una aplicación médica altamente especializada que diagnostica a pacientes diabéticos, así que vamos a descubrir y encontrar la respuesta

In [160]:
```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
fig.set_size_inches(15, 8)

# Lista de categorías populares
popular_app_cats = apps[apps.Category.isin(['GAME', 'FAMILY', 'PHOTOGRAPHY',
                                            'MEDICAL', 'TOOLS', 'FINANCE',
                                            'LIFESTYLE','BUSINESS'])]

# Examina la tendencia de precio graficando el Precio por Categoría
ax = sns.stripplot(x = popular_app_cats['Price'], y = popular_app_cats['Category'], jitter=True, linewidth=1)
ax.set_title('App pricing trend across categories')

# Selecciona las apps con un precio mayor a 200
apps_above_200 = apps[apps['Price'] > 200]
apps_above_200[['Category', 'App', 'Price']]
```

Out[160]:

	Category	App	Price
3327	FAMILY	most expensive app (H)	399.99
3465	LIFESTYLE	💎 I'm rich	399.99
3469	LIFESTYLE	I'm Rich - Trump Edition	400.00
4396	LIFESTYLE	I am rich	399.99
4398	FAMILY	I am Rich Plus	399.99
4399	LIFESTYLE	I am rich VIP	299.99
4400	FINANCE	I Am Rich Premium	399.99
4401	LIFESTYLE	I am extremely Rich	379.99
4402	FINANCE	I am Rich!	399.99
4403	FINANCE	I am rich(premium)	399.99
4406	FAMILY	I Am Rich Pro	399.99
4408	FINANCE	I am rich (Most expensive app)	399.99
4410	FAMILY	I Am Rich	389.99
4413	FINANCE	I am Rich	399.99
4417	FINANCE	I AM RICH PRO PLUS	399.99
8763	FINANCE	Eu Sou Rico	394.99



7. Paid apps vs Free apps

Para las aplicaciones de Play Store en la actualidad, existen cinco tipos de estrategias de precios: gratis, "freemium", de pago, "paymium" y de suscripción. Centrémonos solo en aplicaciones gratuitas y de pago.

Algunas características de las aplicaciones gratuitas son:

- Libres de descarga.
- La principal fuente de ingresos a menudo proviene de la publicidad.
- Por lo general son creadaa por empresas que tienen otros productos y la aplicación sirve como una extensión de esos productos.
- Puede servir como una herramienta para la retención de clientes, la comunicación y el servicio al cliente.

Algunas características de las aplicaciones de paga son:

- Tienen un tiempo de servicio de prueba gratuito, esto para que el usuario pueda conocerla.
- Ofrecen un servicio de mayor especialidad.

¿Además de esto que otras características diferencías a las aplicaciones de pago las aplicaciones gratuitas?

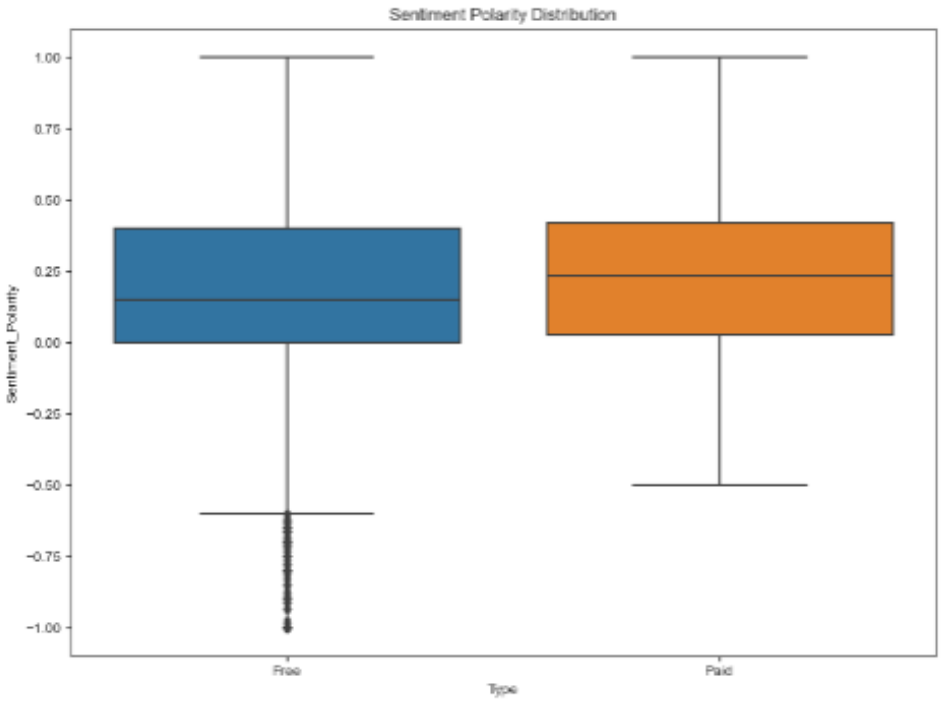
- Gratuita
  - Ofrecen una versión básica gratuita y una versión premium con más características a un costo.
  - Pueden ofrecer modelos de suscripción.
  - Ofrecen compras dentro de la aplicación.
  - Tienden a tener un mayor número de descargas debido a que no tienen costo
- Paga
  - Pagos unicos, mensuales, anuales
  - Sin anuncios
  - Pueden tener menos descargas debido a que tienen un costo
  - Soporte

8. Sentiment analysis

La minería de datos de reseñas de usuarios para determinar cómo se sienten las personas acerca de su producto, marca o servicio se puede realizar mediante una técnica llamada análisis de sentimientos. Las reseñas de los usuarios de las aplicaciones se pueden analizar para identificar si el estado de ánimo es positivo, negativo o neutral con respecto a esa aplicación. Por ejemplo, las palabras positivas en la revisión de una aplicación pueden incluir palabras como "asombroso", "amigable", "bueno", "excelente" y "amor". Las palabras negativas pueden ser palabras como 'malware', 'odio', 'problema', 'reembolso' e 'incompetente'.

¿Qué podemos decir acerca del analisis de sentimiento de las aplicaciones?

In [175]:
```
# Carga el archivo user_reviews.csv
reviews_df = pd.read_csv('C:/Users/Isaac/Desktop/IHD/EBAC DT/CIENCIA DE DATOS/M25 DS/Avance de Proyecto parte 1 y 2/user_reviews
reviews_df
# Une los dos DataFrames (join)
merged_df = pd.merge(apps, reviews_df, on = 'App')
#merged_df.info()
# Elimina los valores nulos (NA) de las columnas Sentiment y Review
merged_df = merged_df.dropna(subset = ['Sentiment', 'Review'])

# Grafica la polaridad de sentimientos para apps gratuitas y de paga
sns.set_style('ticks')
fig, ax = plt.subplots()
fig.set_size_inches(11, 8)

ax = sns.boxplot(x = 'Type', y = 'Sentiment_Polarity', data = merged_df)
ax.set_title('Sentiment Polarity Distribution')
```



9. Conclusion

En este cuaderno, analizamos más de diez mil aplicaciones de Google Play Store. Podemos usar nuestros hallazgos para poder encontrar información valiosa por si alguna vez deseamos crear una aplicación nosotros mismos. Espero que hayas disfrutado el curso!!! :)

Atte. Ivan Alducin