Facultad de Económicas, Universidad Pontificia Comillas (ICADE)

“DE PRADA A PRE-AMADA: DEMOCRATIZACIÓN DE LA MODA DE LUJO A TRAVÉS DE LA SEGUNDA MANO”.

Trabajo de Fin de Grado de Business Analytics

Clave: 202004387
Rocío Medina Llamas

MADRID | Febrero 2025

Contenido

Abstract

El mercado de moda de lujo de segunda mano crece impulsado por la sostenibilidad, accesibilidad y exclusividad, aunque su adopción sigue condicionada por factores psicológicos y culturales. Este estudio analiza los principales impulsores y barreras, aplicando la Theory of Planned Behavior (TPB) e incorporando variables como consumo de arrastre, consumo esnob, control conductual percibido, singularidad, sostenibilidad, riesgo y creatividad emocional.

A través de un enfoque mixto, se han realizado un estudio exploratorio y otro confirmatorio, cuyos resultados muestran que las actitudes y el control conductual percibido determinan la intención de compra, mientras que las normas subjetivas no son significativas. Además, la singularidad, la creatividad y la calidad-precio fortalecen las actitudes positivas, mientras que el riesgo afecta negativamente el control percibido.

El estudio identifica oportunidades para mejorar la percepción de autenticidad, reducir la incertidumbre del consumidor y reforzar la sostenibilidad, aportando estrategias para fomentar la confianza en el lujo de segunda mano.

Palabras clave: Moda de lujo, segunda mano, *Theory of Planned Behavior, Snob consumption, Bandwagon consumption*, consumo de esnob, consumo de arrastre, intención de compra, *Perceived Behavioral Control*, democratización del lujo, valor de reventa, factores motivacionales en el lujo, economía circular, *pre-loved*.

1. Introducción

A. Propósito general, objetivos del estudio y metodología

El mercado de lujo de segunda mano ha crecido significativamente en los últimos años, impulsado por cambios en el comportamiento del consumidor y una mayor conciencia sobre la sostenibilidad. No obstante, persisten barreras en su percepción, lo que puede influir en la intención de compra. Este estudio tiene como propósito analizar los factores que determinan dicha intención, considerando tanto motivaciones como barreras percibidas por los consumidores.

Para ello se empleó una metodología mixta, combinando un análisis exploratorio y un análisis confirmatorio. Las muestras de ambos estudios se obtuvieron mediante un muestreo no probabilístico, utilizando redes sociales como canal de distribución. En la fase exploratoria, se utilizó una encuesta de preguntas abiertas para analizar las percepciones del lujo, diferenciando entre consumidores y no consumidores. Se formularon preguntas clave para comprender las asociaciones emocionales, barreras y motivaciones que influyen en este mercado y se aplicaron técnicas de aprendizaje no supervisado como *text mining* y análisis de *topic modeling* (LDA) para identificar patrones en las respuestas.

En la fase confirmatoria, se diseñó una encuesta estructurada con escalas Likert para medir variables de un modelo teórico basado en la *Theory of Planned Behavior* (TPB) y otras variables identificadas en la literatura. En este sentido, el estudio tuvo los siguientes objetivos específicos:

- Analizar el impacto de los componentes de la TPB en la intención de compra.
- Evaluar la influencia de factores simbólicos y emocionales como la singularidad, el consumo de arrastre, el consumo snob y la creatividad.
- Examinar cómo la sostenibilidad, el valor de reventa y el riesgo afectan la disposición a comprar lujo de segunda mano.
- Identificar barreras y proponer estrategias para aumentar la confianza en este sector.

Los datos del estudio confirmatorio fueron analizados mediante aprendizaje supervisado, empleando modelos de regresión múltiple en Python, permitiendo evaluar la influencia de cada variable sobre la intención de compra.

Este estudio no solo busca comprender los factores que impulsan la intención de compra en el mercado de lujo de segunda mano, sino también proporcionar *insights* prácticos para marcas y plataformas de reventa, basados en datos cualitativos y cuantitativos. A partir de los hallazgos, se pretende mejorar la percepción del lujo de segunda mano y fomentar estrategias comerciales alineadas con las expectativas de los consumidores.

B. Estructura del trabajo

Este Trabajo de Fin de Grado sigue una estructura académica, iniciándose con una revisión de la literatura sobre el lujo, el mercado de moda de lujo de segunda mano y teorías conductuales como la Theory of Planned Behavior (TPB). Se analizan informes de consultoras como Bain y Boston Consulting Group, así como estudios de compañías del sector como Altagamma y Vestiaire Collective.

A continuación, se profundiza en las características y motivaciones del consumo de lujo, abordando conceptos como bandwagon consumption (consumo de arrastre), snob consumption (consumo snob) y el carácter hedonista y simbólico del lujo. También se examina el fenómeno de la segunda mano, diferenciándolo del vintage, para destacar su papel en la democratización del lujo.

A partir de esta base, se desarrolla un modelo teórico, validado mediante un análisis exploratorio y otro confirmatorio, ambos basados en encuestas. Los datos se procesan con Python y técnicas estadísticas como regresión múltiple y topic modeling (LDA).

Finalmente, se presentan conclusiones, evaluando la aplicabilidad del modelo y proponiendo recomendaciones para futuras investigaciones y empresas del sector. Se incluyen anexos con las encuestas y el código utilizado en los análisis.

2. Revisión de la literatura: marco teórico

A. Introducción al lujo

i. Panorama del mercado del lujo

El mercado de artículos personales de lujo, que, según Statista, facturó alrededor de 353 miles de millones de dólares en 2022, es uno de los sectores con crecimiento más rápido de la economía mundial (Stolz, 2022). De acuerdo con Bain&Co, el mercado de lujo en general se puede dividir en nueve categorías: coches de lujo, artículos personales de lujo, hostelería de lujo, vinos y destilados de alta gama, comida gourmet, muebles y artículos de hogar de lujo, arte, yates y jets privados, y por último, cruceros de lujo. Los segmentos de coches, hostelería y artículos personales de lujo representan el 80% de este mercado.

El crecimiento del mercado de artículos personales de lujo ha crecido de forma estable desde hace tres décadas, y a pesar de la caída de esta tasa de crecimiento en el año 2020 por los efectos de la pandemia del Covid-19, en 2021 ya se habían recuperado

prácticamente los niveles de facturación pre-covid en los sectores más relevantes del mercado de lujo en general (D'Arpizio, Levato, Gault, De Montogolfier, & Jaroudi, 2021).

Después de la pandemia, hay ciertos hábitos de consumo que han cambiado. Uno de ellos es que el gasto en lujo ha virado de experiencias a productos tangibles, detectable en la recuperación más rápida del conjunto de categorías de productos de lujo, frente al de experiencias; en concreto, el mercado de artículos personales de lujo. (D'Arpizio, Levato, Gault, De Montogolfier, & Jaroudi, 2021). Otro patrón de consumo en auge tras la pandemia, es la preferencia por canales de distribución directos de las marcas de lujo sobre los canales de venta al por mayor o multimarca, con un crecimiento acumulado del 11% entre 2019 y 2021, frente a la caída del 7% del canal indirecto. Dentro de los canales directos de distribución, el *retail* online de las casas de lujo se espera que llegue a alcanzar alrededor de un 30% del mercado global de lujo, superando a las tiendas físicas mono-marca (D'Arpizio, Levato, Gault, De Montogolfier, & Jaroudi, 2021). La tendencia de consumo más relevante para este estudio, en la que profundizaremos posteriormente, es el crecimiento del mercado de segunda mano de productos de lujo, a un ritmo incluso más acelerado que el del mercado de artículos personales de lujo.

En cuanto a las tendencias demográficas en el consumo de lujo, la geografía y generación de los consumidores resultan de interés. Tradicionalmente, en cuanto a geografía se refiere, los mercados líderes en el lujo han sido Europa y América. Sin embargo, en el 2020, Asia se convirtió en la región dominante, ya que la pandemia disparó el consumo de lujo en mercados locales, sobre todo en China. La cuota de Asia sobre el mercado de artículos personales de lujo mundial fue del 39% en 2021, y el consumo de lujo local alcanzó el 90% sobre el consumo total de bienes personales de lujo en 2021 (D'Arpizio, Levato, Gault, De Montogolfier, & Jaroudi, 2021). Por otro lado, las generaciones más jóvenes (generaciones Y y Z) conforman ya más de la mitad de la cuota del mercado de artículos personales de lujo, y se espera que superen el 70% de las compras del sector en 2025 (Cinco Días, 2022).

ii. Características clave de los productos de lujo

El término “lujo” y sus implicaciones han sido ampliamente definidos por diversos autores, incorporando matices diferentes. Sin embargo, los productos de lujo tienen ciertas características clave que ayudan a definir el término con más claridad. La literatura

anterior sobre la psicología del lujo alude a atributos como la manufactura (parcial o completa) del producto, la oferta de un servicio exclusivo y la excelencia del producto, etc. para determinar un criterio de clasificación de productos o servicios considerándolos lujo o no. Para ello, otros autores emplean rasgos como la máxima calidad, sofisticación y rareza (aludiendo a la escasez) refiriéndose a estilos de vida adinerados, cómodos y opulentos.

En síntesis, no solo las características básicas como la exclusividad, la escasez y el nivel elevado de precio y calidad definen a los productos de lujo, sino que también se han de tener en cuenta factores subyacentes a lo económico o tangible - como pueden ser la estética, la herencia y la historia – ya que rara vez los atributos del producto son suficiente para justificar el precio y mantener el éxito y el estatus del lujo (Turunen & Leipämaa-Leskinen, 2015).

Sin embargo, estas propiedades resultan cada vez menos definitivas, pues se está desdibujando la imagen tradicional del lujo a través de diferentes mecanismos de democratización, haciendo más accesible al público en general un concepto que tradicionalmente estaba reservado a élites y altas esferas. Algunos de ellos pueden ser el crecimiento de la segunda mano como canal de distribución de artículos de lujo, las colaboraciones de marcas de *mass-market* con casas de lujo o la producción de ciertos artículos a precios asequibles y con notable exposición de marca que pasan a convertirse en objetos de culto, dejando atrás cualidades como la exclusividad, la escasez y la calidad excelente.

iii. Motivadores tradicionales detrás de la compra de artículos de lujo

De todos modos, es innegable que, psicológicamente hablando hay ciertos factores motivacionales identificados por estudios anteriores que afectan de forma significativa a la intención de consumo de lujo.

Son un número considerable de autores los que han abordado el lujo desde un enfoque conductual. Uno de los principales estudios es el de Leibenstein, en el que se introducen los conceptos de *"conspicuous consumption"* y *"bandwagon consumption"*, a los que nos referiremos como "consumo ostentoso" y "consumo de arrastre" respectivamente de ahora en adelante. Para explicar el concepto de consumo ostentoso, hemos de introducir

primero los términos consumo de arrastre y consumo esnob. Por un lado, el consumo de arrastre es *"el grado en el que la demanda de un producto aumenta debido al hecho de que otros también están consumiendo el mismo producto"* (Leibenstein, 1950, pág. 189). Por el contrario, el consumo esnob, se define como "*el grado en el que la demanda de un producto se reduce debido al hecho de que otros también están consumiendo el mismo producto"* (Leibenstein, 1950, pág. 189)*, "tenemos, pues, en el efecto esnob una relación opuesta pero completamente simétrica con el efecto bandwagon"* (Leibenstein, 1950, pág. 199). Así, cuando hablamos de consumo ostentoso, nos referimos a *"la tendencia de los consumidores a comprar productos de lujo para mostrar su riqueza y alcanzar un estatus social superior"* (Kessous & Valette-Florence, 2019, p.315) o *"la práctica de utilizar productos para señalar aspiraciones de estatus social a otros consumidores"* (Eastman, Goldsmith, & Flynn, 1999, pág. 42). Es decir, el ser humano se ve motivado a comprar productos de lujo, no solo por sus cualidades, sino también por alcanzar un estatus social superior al que tiene, ya sea queriendo diferenciarse de otros consumidores (consumo esnob) o buscando parecerse a ellos (consumo de arrastre).

Otros autores también se han referido a las orientaciones de compra general, diferenciando entre una orientación utilitaria, basada en los atributos intrínsecos del producto o servicio, *"refiriéndose a un tipo de compra eficiente u orientado a tareas o metas"* (Turunen & Pöyry, 2019, p. 550) siendo una *"adquisición de productos y/o información de una manera eficiente y puede verse como un reflejo de un resultado de compra más orientado a la tarea, cognitivo y no emocional"* (Jones, Reynolds, & Arnold, 2006, pág. 974)*;* y una orientación hedonista *"basada en el valor de entretenimiento y de experiencia de la compra"* (Turunen & Pöyry, 2019, p. 550) lo cual *"refleja el valor recibido de los aspectos multisensoriales, fantásticos y emotivos de la experiencia de compra"* (Jones, Reynolds, & Arnold, 2006, pág. 974). A estas, se añade la orientación social u ostentosa aludiendo al aspecto simbólico y recreativo de la compra (Turunen & Pöyry, 2019, p. 550) (Shim, 1996). Dentro de este planteamiento, podríamos decir que los consumidores de lujo recurren a las perspectivas hedonista y social u ostentosa a la hora de comprar, pasando a tercer plano la perspectiva utilitarista del producto. Así, en muchas ocasiones, el comprar un bolso de lujo o ir a cenar a un restaurante de lujo, no es una decisión basada en cubrir una necesidad física (tener un lugar donde llevar las cosas) o fisiológica (alimentarse), sino una necesidad social de pertenecer a un estatus que conoce y se puede permitir cierto bolso o restaurante, además de la perspectiva hedonista

del entretenimiento, experiencia o placer sensorial que supone esa acción de compra o consumo. Según autores como (Berthon, Pitt, Parent, & Berthon, 2009) estas tres dimensiones o perspectivas de compra son dependientes del contexto, y están sujetas a cambios en el tiempo.

B. La segunda mano

i. Diferencias entre segunda mano y vintage

Los términos vintage y segunda mano, pese a la creencia común y a la confusión frecuente, son conceptos claramente diferenciados. Primeramente, el término vintage hace referencia a "*piezas singulares y auténticas que representan el estilo de una era en concreto*" (Gerval, 2008). Para (Cervellon, Carey, & Harms, 2012), esta época debe ser entre las décadas de 1920 y 1980. Otros autores completan esta definición especificando que son productos que han sido poseídos anteriormente (*pre-owned*), pero no necesariamente usados (Sihvonen & Turunen, 2016). Así, entran en juego ciertos factores que aportan valor como puede ser su historia, o darle una nueva vida a ese objeto (Turunen & Leipämaa-Leskinen, 2015). Por otro lado, al ser artículos de otra época, la posibilidad de incremento de valor en cuanto a su reventa es otro factor a tener en cuenta (Sarial-Abi, Vohs, Hamilton, & Ulqinaku, 2016). Otras motivaciones para comprar artículos vintage pueden ser la nostalgia o la necesidad del consumidor de singularidad, que (Sihvonen & Turunen, 2016) denominan *Consumers' Need For Uniqueness* (CNFU).

Por el contrario, un artículo de segunda mano se define como aquel que ha sido previamente poseído y usado, independientemente de la vida del objeto (Cervellon, Carey, & Harms, 2012) (Turunen & Leipämaa-Leskinen, 2015). Centraremos este estudio los artículos de moda de lujo de segunda mano, por lo que hablaremos de productos previamente poseídos y usados, que pueden pertenecer a cualquier época. A este tipo de artículos también se les denomina "pre-amados" (*pre-loved*), agregando el componente del amor, haciendo referencia a que su anterior dueño amó y cuidó el producto.

ii. Mercado de lujo de segunda mano

El mercado de lujo de segunda mano, tal y como se presenta en el informe de BCG en colaboración con Vestiaire Collecctive por Abtan, et al. (2019), ha estado

tradicionalmente opacado por el lujo de primera mano. Sin embargo, en los últimos años, este segmento ha experimentado un crecimiento sustancial, tomando rápidamente cada vez más protagonismo. De acuerdo con Bain&Co., el mercado de lujo de segunda mano alcanzó los 33 mil millones de euros (€33B) en el año 2021, gracias al crecimiento acelerado de la demanda y a el incremento de la competitividad en la oferta. Entre 2017 y 2021, este mercado creció un 53% más que el mercado de artículos de lujo de primera mano (D'Arpizio, Levato, Gault, De Montogolfier, & Jaroudi, 2021) , y según Statista, se espera que llegue a alcanzar los 48 mil millones de euros en 2027 (Statista, 2023).

En cuanto a los principales consumidores de este mercado, en el ámbito generacional, el informe de BCG y Vestiare collective indica que "*en términos de diferencias generacionales, los consumidores de lujo más jóvenes son los mayores participantes en el mercado de segunda mano, con el 54% de la Generación Z y el 48% de los consumidores de lujo millennials*" (Abtan, et al., 2019, p. 2).

Una de las causas primordiales de este crecimiento acelerado, según Abtan et al. (2019), es el precio accesible de sus productos, que permite a los consumidores mantener una renta disponible más alta a la que tendrían en caso de adquirir este tipo de productos de primera mano.

Otro factor contribuyente puede ser el incremento de la competitividad de la oferta, que se materializa en diferentes aspectos. El primero de ellos podría ser la proliferación de plataformas de *marketplace online* efectivas y profesionales. La oferta ya no se concentra en tiendas físicas de segunda mano especializadas, sino en plataformas online. El éxito del comercio online ha arrasado en este mercado, elevando su potencial y sus ingresos. En efecto, BCG señala que "*gran parte del crecimiento de la reventa provendrá de las plataformas en línea, que actualmente generan alrededor del 25% de las ventas del mercado mundial de artículos de segunda mano*" (Abtan, et al., 2019, p. 2). Actualmente hay numerosas plataformas dedicadas a la venta de artículos de segunda mano, en las que se venden también productos de lujo como e-Bay, Depop, Vinted o Wallapop; o dedicadas específicamente a poner en contacto a vendedores y compradores de artículos de lujo de segunda mano como Vestiaire Collective, Rebag y The RealReal, que cuentan con millones de usuarios.

Sin embargo, una de las principales barreras del *e-commerce* en el caso de la segunda mano, es la verificación de autenticidad de los productos y la gestión de malas

experiencias, devoluciones o incluso estafas. Como veremos posteriormente, uno de los elementos que más influyen en la compra de prendas y accesorios de lujo de segunda mano es la autenticidad y el buen estado del producto (Sihvonen & Turunen, 2016). Comprar un artículo que pueda ser una imitación, supone un factor de riesgo en la inversión en lujo de segunda mano, como afirman Turunen y Leipämaa-Leskinen (2015). Por tanto, otro factor contribuyente al crecimiento puede ser el hecho de que la mayoría de las plataformas mencionadas han implementado políticas de certificación de la autenticidad de los productos antes de que puedan ser publicados en su interfaz, para aumentar su competitividad y calidad de servicio. Algunas plataformas como Vestiaire Collective, implementan medidas estrictas para garantizar la autenticidad de los productos. Solicitan a los vendedores números de serie, imágenes de etiquetas y otros elementos verificables. Además, cuentan con un equipo de expertos en control de autenticidad en diversas áreas, como moda urbana y gemología. Complementan este proceso con inteligencia artificial y algoritmos avanzados para mejorar la precisión de las inspecciones. Gracias a estas estrategias y su firme postura contra las falsificaciones, han logrado fortalecer la confianza del consumidor, reduciendo barreras del crecimiento del lujo de segunda mano.

Otro factor relevante es el carácter sostenible asociado a la segunda mano. Es sabido que la industria de la moda es una de las más contaminantes, llenando nuestro planeta de residuos textiles y químicos que surgen tanto del proceso de producción de las prendas, como del desecho masivo de ropa en todo el mundo. El éxito del modelo de negocio *fast-fashion* ha contribuido considerablemente al deterioro del medio ambiente, propulsado por el consumismo excesivo y con ello, el desecho de millones prendas, casi cada temporada, en todo el mundo. En este panorama, la moda de lujo en sí ya se propone como una alternativa genérica al problema, ya que un producto de calidad y de diseño atemporal, debería tener una durabilidad en el sentido físico y estético mucho superior a la de una prenda de *fast-fashion*. De acuerdo con Carrigan et al. (2013, pág. 1296), "*hasta cierto punto, las marcas de moda de lujo representan la antítesis de la moda desechable, ya que muchos clientes aprecian su herencia y calidad, y el servicio postventa que prolonga la vida útil de la compra*". Por otro lado, la segunda mano también supone una vía de apoyo a un consumo de moda más sostenible, ya que promueve el adquirir productos que ya han sido poseídos, y que probablemente, si nadie los comprase, serían

desechados, fomentando la reutilización de recursos, contribuyendo a la reducción del consumo de energía y minimizando el impacto contaminante de producir nuevas prendas.

Por tanto, no es de extrañar que la fusión de la compra de artículos de lujo y el canal de segunda mano sea la idónea para promover estilos de vida sostenibles, que apuesten por reformar el sistema de consumo de moda.

iii. Segunda mano como mecanismo de democratización del lujo

El término democratización en el ámbito del lujo es un concepto que ha ganado relevancia en los últimos años. Ha sido definido por numerosos autores con matices diferentes y aplicado a diversos campos.

Sin embargo, la mayor parte de estos autores parten de una base común que es la definición de su raíz, democracia. Según la RAE, la palabra democracia tiene significados como "*participación de todos los miembros de un grupo o de una asociación en la toma de decisiones*" y "*forma de sociedad que reconoce y respeta como valores esenciales la libertad y la igualdad de todos los ciudadanos ante la ley*" (Real Academia de la Lengua Española, 2023). De ambas definiciones, los aspectos más importantes a la hora de delimitar el concepto de democratización del lujo son la participación de todos los miembros de un grupo y la igualdad de todos ellos.

La definición que tomaremos como base para este estudio es la que explica la democratización desde el punto de vista del mercado del lujo como "*el proceso de amplificación de la disponibilidad y el acceso del público en general a productos de lujo, y su consecuente reducción percibida en distinción, exclusividad y auto diferenciación*" (Shukla & Rosendo-Ríos, 2022).

El fenómeno de democratización ha afectado al mercado del lujo de forma significativa desde la última década, años en los que la clase media ha aumentado de forma exponencial (Canals, 2019). Este sector poblacional ha desarrollado un carácter aspiracional hacia las clases altas y los productos y servicios propios de las mismas, invirtiendo en artículos y experiencias de lujo en ocasiones especiales o cuando sus ingresos cubren sus necesidades básicas. Así, aspectos como la digitalización y la segunda

mano han propiciado incrementar la accesibilidad de este grupo socioeconómico a los productos de alta gama, contribuyendo a la democratización del lujo.

El hecho de que más personas tengan acceso a productos que antes estaban reservados a una minoría poblacional afecta a numerosas dimensiones del mercado del lujo.

Primeramente, impacta directamente en la base conductual y psicológica sobre la que se sustenta la industria del lujo, reduciendo la sensación de exclusividad. Como hemos comentado anteriormente, el interés por consumir productos premium se ve motivado por dos tipos de conducta: *bandwagon consumption* (o de arrastre) y *snob consumption* (esnob). La primera de ellas se ve fomentada cuantas más personas deseen y compren artículos de lujo, mientras que la segunda sigue el patrón opuesto, ya que cuanto más conocido y deseado sea un producto, menos probable es que ciertos individuos lo consuman y quieran ser asociados con ello.

Otra de los impactos de la democratización del lujo sobre esta industria es la adaptación de la oferta de las empresas a este fenómeno, mediante la cual las firmas de lujo se ven motivadas a desarrollar líneas más accesibles al público tanto en cuestiones de precio como de distribución. Así, casas como Armani cuentan con líneas de prendas básicas como camisetas y accesorios, e incluso submarcas (como Armani Jeans, ahora incluida en Armani Exchange) a precios más asequibles para poder acercarse a la clase media que quiere consumir sus productos, pero no se puede permitir su oferta habitual. Este acercamiento a la clase media supone un incremento de la base de clientes de las casas de lujo, haciendo crecer el mercado global del lujo.

Un método alternativo que tienen las firmas de moda de lujo para adaptar su oferta sin crear una línea adicional con las colaboraciones con empresas de moda accesible, denominadas '*mass market*', para crear colecciones con unidades y duración limitadas, de productos a precios accesibles y con una calidad inferior a la usual de la firma. Esta estrategia, con el nombre *masstige* – mezcla de los términos masa y prestigio en inglés - acerca a las casas de lujo tanto al público que desea poder comprar prendas de la marca, como a un público que hasta el momento desconocía la marca y tiene un primer contacto con ella a un precio más asequible.

También ha fomentado la democratización del lujo el auge del modelo de negocio "sharing economy" o economía colaborativa, que hace referencia a "*plataformas intermediarias que permiten el intercambio entre pares y crean oportunidades entre los*

consumidores y/o las organizaciones para comprar, alquilar y/o comerciar con activos no utilizados o infrautilizados" (Christodoulides, Athwal, Boukis, & Semaan, 2021). Este modelo de negocio ha afectado a todo tipo de industrias, desde el transporte – donde triunfa el *carsharing* - hasta la moda de lujo, y ha sido adoptado de forma destacable por las generaciones más jóvenes.

Un ejemplo de economía colaborativa en la moda de lujo es el éxito de plataformas de alquiler de prendas y accesorios de lujo, en formato de suscripción mensual que da acceso a una selección de artículos de alta gama durante un periodo de tiempo determinado, con la opción de poder comprarlos, como la plataforma española Efímero.

Otra forma, y la que da razón de ser de este estudio es la compra de productos de lujo por canales de segunda mano. Comprar prendas y accesorios 'pre-amados' permite al consumidor identificarse con marcas de lujo a un precio accesible, y de forma sostenible, fomentando la originalidad en cuanto a la selección de productos. Esto facilita adquirir productos con historia o vida anterior, dándoles una relevancia emocional superior a la de la compra de lujo en la era de la democratización.

Para las firmas de lujo, la segunda mano como mecanismo de democratización de lujo supone un arma de doble filo. Por un lado, puede servir para que un potencial consumidor comience a crear relación con la marca, formando asociaciones positivas y emocionales con ella. Así, un cliente que ha podido comprobar la calidad de un bolso de una marca de lujo, es muy probable que decida adquirir uno de primera mano en el futuro cuando tenga los medios necesarios. Sin embargo, dado que la mayor parte de la distribución de artículos de segunda mano de lujo se da a través de canales indirectos, las marcas de lujo no llegan a ver beneficios económicos de la popularidad de sus productos de segunda mano. Es por ello, que muchas de estas empresas están comenzando a ofrecer a sus consumidores un canal directo de consumo de prendas y accesorios de la marca de segunda mano, pudiendo así asegurar la autenticidad de los productos y lucrarse de forma directa de esta tendencia de mercado. Además, este tipo de iniciativas también refuerza el compromiso de las compañías de moda de lujo con la sostenibilidad y la economía circular, destacando propuestas como Gucci *Vault* y Miumiu *Upcycled*, que toman piezas vintage de la compañía, las restauran y re-diseñan para darles una nueva vida sin comprometer el medioambiente.

C. Factores motivacionales para comprar segunda mano de lujo

i. Theory of Planned Behavior (TPB)

La Teoría del Comportamiento Planeado o *Theory of Planned Behavior*, a la que nos referiremos de ahora en adelante como TPB, fue desarrollada hace más de tres décadas y sin embargo sigue siendo de vital importancia para comprender los patrones de comportamiento de los consumidores. Esta teoría se centra en los factores que influyen sobre la intención que tiene un individuo de tomar un determinado comportamiento (Azjen, 1991). En este estudio, emplearemos la TPB como base para el análisis de los factores motivacionales detrás de la intención de compra de moda de lujo de segunda mano.

Para contextualizar esta teoría, debemos aclarar primero el concepto de intención, ya que se encuentra en el corazón de esta doctrina. *"Se supone que las intenciones capturan los factores motivacionales que influyen en un comportamiento; son indicaciones de cuánto están dispuestas a esforzarse las personas (...) para realizar el comportamiento. Como regla general, cuanto más fuerte sea la intención de participar en un comportamiento, más probable debe ser su rendimiento"* (Azjen, 1991, p. 181). De acuerdo con Ajzen, existen tres factores determinantes que componen la TBP e influyen en la intención de tomar un comportamiento.

El primero de ellos, son las actitudes (*attitudes*), que hacen referencia a *"el grado en que una persona tiene una evaluación o valoración favorable o desfavorable de la conducta en cuestión",* (Azjen, 1991, p. 188). Es decir, las actitudes se refieren a si el individuo asocia un determinado comportamiento a connotaciones positivas o negativas. Por tanto, tiene sentido establecer que puede existir una correlación positiva entre la actitud de un individuo hacia un producto y su intención de compra (Jiang, Yang, & Jun, 2013). Así, en nuestro caso, nos centraremos en las actitudes de los individuos sobre la compra de prendas y accesorios de lujo de segunda mano.

El segundo factor es denominado normas subjetivas (*subjective norms*), que hacer referencia a *"la presión social percibida para realizar o no un comportamiento"* (Azjen, 1991, p. 188). En otras palabras, son aquellas normas sociales de comportamiento que un individuo tiende a seguir, afirmando que al individuo el importa (en mayor o menor grado) lo que la sociedad piense de sus comportamientos. Si un comportamiento está

altamente influenciado por normas subjetivas, su impacto en la intención de realizarlo dependerá de si estas lo perciben de forma positiva o negativa.

Por último, el tercer factor que compone la TPB es el Control Conductual Percibido (Perceived Behavioral Control), al que nos referiremos como PBC. Este concepto pretende explicar *"la facilidad o dificultad percibida para realizar el comportamiento y se supone que refleja la experiencia pasada, así como los impedimentos y obstáculos anticipados"* (Azjen, 1991, p. 188). Otros autores se han referido a este concepto anteriormente como factores facilitadores, contexto de oportunidad, recursos o control de acción (Azjen, 1991).

Ajzen propone el siguiente modelo de influencia sobre el comportamiento, que emplearemos como base en este estudio para la construcción de nuestro propio modelo.

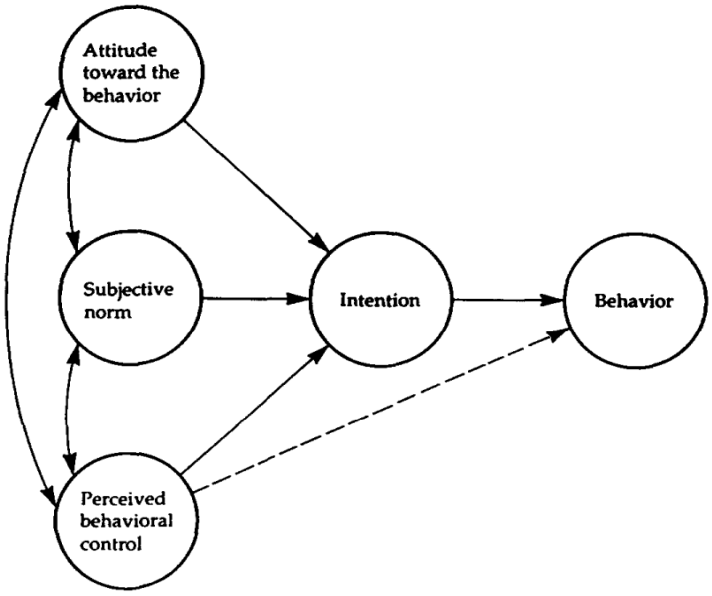


Ilustración 1: Esquema de los componentes de la TPB (Azjen, 1991). (Líneas continuas: relación directa; líneas discontinuas: relación indirecta).

Se proponen hipótesis de influencia directa e indirecta (flechas continuas y discontinuas respectivamente) sobre la intención y el comportamiento. Se sugiere que los tres componentes primordiales de la TPB influyen de forma directa en la intención de tener un determinado comportamiento, que a su vez influye directamente en el comportamiento. También tiene influencia sobre el comportamiento, aunque de forma indirecta el PBC. Por último, se establecen relaciones bilaterales de influencia directa entre los tres componentes básicos (actitudes, normas subjetivas y PBC).

Por tanto, de la TPB extraeremos como fundamentos para nuestro modelo, los conceptos de actitud, normas subjetivas y PBC, además de su influencia en la intención, resultando en las siguientes hipótesis.

H0: Las normas subjetivas influyen de forma positiva y significativa sobre la intención de compra de prendas de lujo de segunda mano.

H2: El PBC influye de forma positiva y significativa sobre la intención de compra de prendas de lujo de segunda mano.

H3: Las actitudes hacia el producto influye de forma positiva y significativa sobre la intención de compra de prendas de lujo de segunda mano.

ii. Factores identificados por estudios anteriores

Pese a que muchos autores han estudiado los factores motivacionales que afectan al comportamiento del consumidor para comprar lujo, son relativamente escasos los que toman la perspectiva de segunda mano de lujo. Tras haber examinado la literatura existente, vamos a analizar los principales factores identificados por estudios anteriores, agrupados por temáticas.

Singularidad

La singularidad es un factor popular entre autores cuando se estudia el lujo, ya sea de primera o de segunda mano, pues una de las características principales de los artículos de lujo es la exclusividad. La exclusividad implica singularidad, sobre todo en el sentido de escasez de unidades, o en la dificultad de acceder a determinados productos, como el bolso Birkin de Hermès.

En su estudio sobre los factores motivacionales para comprar lujo de segunda mano, Turunen y Leipämaa-Leskinen (2015) identificaron una variable relacionada con la singularidad a la que denominaron "*unique find*" (hallazgo único). Este concepto hace referencia a la representación sobre uno mismo que hace el poseer un determinado producto escaso y difícil de encontrar. En el caso de lujo de segunda mano, hay ediciones limitadas o versiones de productos que hoy en día están descatalogados, y que solo se puede encontrar un número limitado de unidades en todo el mundo. En cualquier caso, el carácter singular de estos productos ayuda al comprador a distanciarse de la norma, del

público general. De esta manera, se llega a la hipótesis de que lo que Kessous y Valette-Florence (2019) determinan CNFU (*Consumer Need For Uniqueness*), factor que influye de manera positiva en el concepto de consumo esnob, ya que favorece el consumo de productos únicos y reduce la intención de comprar productos con alta demanda. Esta perspectiva del producto conduce a considerar muchos de los artículos de segunda mano de lujo unos tesoros, en vez de la connotación negativa de basura o antigüedad que se le ha otorgado tradicionalmente.

H1.1: El consumo de arrastre influye de forma positiva y significativa sobre las normas subjetivas.

H1.2: El consumo esnob influye de forma positiva y significativa sobre las normas subjetivas.

H7: La singularidad influye de forma positiva y significativa sobre las actitudes hacia el producto.

H15: La singularidad influye de manera positiva, significativa e indirecta sobre el consumo esnob.

Conocimiento de marca

La popularidad de una marca y la consideración social que tenga nuestro entorno sobre ella es algo muy relevante a la hora de determinar la intención de comprar un artículo de lujo de segunda mano. Así, si una firma está en boca de todos, resultará más atractivo buscar y comprarla de segunda mano desde la perspectiva del consumo de arrastre.

En uno de sus estudios, Turunen y Pöyry (2019), identifican este factor y lo denominan "*Brand consciousness*" (consciencia de marca), estableciendo que los consumidores que buscan una marca en concreto ven el canal de segunda mano como cualquier otro canal de distribución para llegar hasta la marca deseada. De hecho, determinan que los consumidores de lujo de segunda mano que van buscando una marca concreta consideran esta decisión de compra como sabia, ya que es una opción más asequible económicamente hablando para adquirir productos de la firma. También indican en el mismo estudio que los consumidores con consciencia de marca suelen tener también conocimiento sobre ediciones limitadas, productos estacionales y diseños de invitados o colaboraciones, que

son más deseados por su escasez y singularidad (Turunen & Pöyry, 2019). Por otro lado, en este mismo estudio también se habla de la lealtad a la marca como factor motivacional, declarando que "*Los informantes describieron cómo habían buscado un producto específico*" (Turunen & Pöyry, 2019), refiriéndose a un modelo en concreto de una marca determinada.

H4: El conocimiento de marca influye de forma positiva y significativa sobre las actitudes hacia el producto.

H12.1: El conocimiento de la marca influye de forma positiva, significativa e indirecta sobre las normas subjetivas, a través del consumo de arrastre.

H12.2: El conocimiento de la marca influye de forma negativa, significativa e indirecta sobre las normas subjetivas a través del consumo esnob.

Calidad-precio

Uno de los factores más relevantes a la hora de identificar las motivaciones para comprar lujo de segunda mano es el binomio de calidad-precio. Al tratarse de artículos de lujo, la calidad y precio esperados son elevados, sin embargo, el canal de distribución de segunda mano contribuye a reducir el valor del precio estimado, aunque también puede suponer una reducción de la calidad del producto (por su potencial deterioro en su anterior vida).

Este bloque de factores abarca tanto la motivación económica, como la búsqueda de la mejor calidad de un producto al precio más competitivo. Dentro de la primera perspectiva, podemos hacer referencia a cómo, de acuerdo con Stolz (2022), "*el precio más bajo de los productos de segunda mano es una razón mencionada con frecuencia por la que los consumidores compran productos de segunda mano en lugar de nuevos*". Una de las causas es la consideración generalizada de que los productos de lujo hoy en día tienen unos precios desorbitados. Ejemplo de ello son los incrementos de precio continuados de marcas como Chanel en algunos sus productos más emblemáticos, que en base a los datos publicados por Fashion United, ha aumentado su valor desde 2020 en casi un 60%. Estas políticas de precios motivan cada vez más a adquirir estos artículos por otros canales como la segunda mano, que puedan ofrecer precios más atractivos para los consumidores. Sin embargo, hay que tener en cuenta que el hecho de que un artículo de lujo sea de

segunda mano no siempre disminuye su valor, sino que en piezas de edición especial o de alta demanda, el valor de reventa puede llegar a multiplicarse.

En la perspectiva de la búsqueda de optimización del binomio calidad-precio, podemos centrarnos en "*esos significados que se asocian a la caza de gangas y a hacer buenos tratos con respecto a las decisiones financieras y de precio (...) para obtener el mejor valor por su dinero*" (Turunen & Leipämaa-Leskinen, 2015, p. 61). El consumidor, busca adquirir artículos de gran calidad con el menor deterioro y al mejor precio posibles. Otros autores hacen referencia en sus estudios, a cómo participantes de la muestra declaraban sentir que estaban ahorrando dinero al comprar las prendas de lujo que deseaban a través del canal de segunda mano, otorgándole a la compra un fundamento racional (Turunen & Pöyry, 2019).

H10: La calidad-precio influye de forma positiva y significativa sobre las actitudes hacia el producto.

Nostalgia

En lo que se refiere a la antigüedad del artículo, cuanto más antiguo, resulta más difícil de encontrar, haciéndolo más exclusivo y, finalmente, más deseable (Sihvonen & Turunen, 2016).

El componente de apelación a la nostalgia está presente sobre todo en compras de productos vintage, aunque también se puede hallar en la de los que previamente hemos definido de segunda mano (Stolz, 2022). Esto se debe a que los artículos vintage y de segunda mano, despiertan emociones asociadas a una época en concreto, generando un sentimiento de nostalgia en el consumidor que parece favorecer la compra (Stolz, 2022). También autores como Kessous y Valette-Florence (2019) hablan de las conexiones nostálgicas y el vínculo con la marca, estableciendo la nostalgia como antecedente al vínculo con el producto en cuestión y la marca fabricante. En ese mismo estudio, se validó la hipótesis de que "*las conexiones nostálgicas refuerzan la relación entre la búsqueda de estatus y el apego a la marca. Más específicamente, postulamos que este efecto moderador de las conexiones nostálgicas es mayor para los productos de lujo de segunda mano que para los productos de lujo de primera mano*" (Kessous & Valette-Florence, 2019, p. 325).

La nostalgia también incluye el apego emocional que siente el consumidor por un producto que tuvo un dueño previo. Turunen y Leipämaa-Leskinen lo denominan tesoro pre-amado ("*pre-loved treasure*"), que "*indica los fuertes compromisos emocionales que hay detrás de las posesiones de lujo de segunda mano*" (Turunen & Leipämaa-Leskinen, 2015, p. 61) y en este concepto toman mayor importancia la autenticidad y la vida anterior del producto, por encima de su valor económico. "*Se consideraba que la vida anterior del producto daba a las posesiones de lujo de segunda mano un carácter más distintivo que sus homólogos nuevos*" (Turunen & Leipämaa-Leskinen, 2015, p. 61).

H9: La nostalgia influye de forma positiva y significativa sobre las actitudes hacia el producto.

Sostenibilidad

Como explicamos anteriormente, los conceptos de lujo y de segunda mano están ligados por separado a la sostenibilidad, ya que suponen alternativas al consumo irresponsable e, por lo que la combinación de ambos también estará ligada a la responsabilidad medioambiental. Ya en 2015 Turunen y Leipämaa-Leskinen identificaron la relevancia de las consideraciones medioambientales en los hábitos de consumo como factor motivacional para comprar artículos de segunda mano de lujo. Definieron esta variable como *"los significados ecológicos y responsables que se atribuyen a la posesión y adquisición de artículos de lujo de segunda mano"* (Turunen & Leipämaa-Leskinen, 2015, p. 60). Otros autores ligan este concepto, no solo a estilos de vida basados en elecciones sostenibles, sino a una "*profunda crítica al materialismo y el consumismo en general*" (Joung & Park-Poaps, 2013). De esta manera se establece que la compra de estos productos provoca en el consumidor un sentimiento de satisfacción y orgullo, al estar tomando una clara posición en contra del consumismo excesivo e irresponsable (Turunen & Leipämaa-Leskinen, 2015). También alude a la sostenibilidad Stolz (2022), como factor intrínseco al comprador y que influye de manera positiva en la actitud sobre los productos de lujo de segunda mano.

H5: La sostenibilidad influye de forma positiva y significativa sobre las actitudes hacia el producto.

H13: La sostenibilidad influye de forma positiva, significativa e indirecta sobre el consumo de arrastre o *bandwagon consumption*.

Riesgo

Otro factor relevante, que hemos introducido anteriormente, es el nivel de riesgo. Los riesgos a los que se enfrenta el comprador se concentran en la autenticidad y el estado del producto, asociados a imitaciones o estafas que pueden suceder tanto en *retail* físico como online. Como defienden Turunen y Leipämaa-Leskinen, el "*riesgo de la inversión representa el cuestionamiento de la autenticidad de los productos, lo que puede llevar a riesgos financieros y reputacionales*" (Turunen & Leipämaa-Leskinen, 2015, p. 61). En muchas ocasiones, aunque informados, los consumidores de lujo de segunda mano no son expertos en el área, por lo que son susceptibles de ser engañados. Sin embargo, el comercio online de segunda mano de lujo trabaja para derribar esta barrera ofreciendo un servicio de autentificación previo a publicar productos en sus plataformas. Además, al tratarse de artículos de lujo, en los que el precio establecido será superior a otras prendas de segunda mano, el riesgo en la inversión será mayor. Aquí entra en juego el concepto de aversión a las pérdidas, suceso psicológico que lleva al ser humano a valorar menos las ganancias que las pérdidas de igual cuantía. En definitiva, el riesgo de que un artículo no sea auténtico o que las condiciones no sean las especificadas, especialmente en el comercio online, es un factor que influye de forma negativa en la intención de compra de artículos de lujo de segunda mano.

H11: El riesgo influye de forma negativa y significativa sobre las actitudes hacia el producto.

H14: El riesgo influye de manera negativa y significativa sobre el control percibido o PBC.

Valor de reventa

Un aspecto que ha ganado relevancia en los últimos años en los estudios sobre las motivaciones de compra de segunda mano de lujo es el valor de reventa. Tal y como mencionan Turunen y Pöyry, "*muchos de los informantes del estudio no se consideraban*

consumidores finales, sino que pensaban que probablemente pasarían el producto a otro usuario tomando así posteriormente el rol de vendedor" (Turunen & Pöyry, 2019, p. 552). Es decir, una proporción considerable de los compradores de lujo de segunda mano pretenden tomar posesión del articulo únicamente durante un periodo de tiempo, y después revenderlo. Pese a que este factor está relacionado con la calidad de los productos de lujo (que determina su alta durabilidad y atemporalidad en cuanto a la estética, permitiendo que tenga varios poseedores sin perjudicar mucho su estado) y con la familiaridad o conocimiento de marca (que permite conocer cuándo un producto es lo suficientemente escaso o demandado como para intuir que tendrá un precio de reventa predecible (Liu, Xia, & Lang, C. , 2023). Destaca la "*percepción de la compra como una inversión y la intención explícita de vender el producto en un punto posterior en el tiempo (independientemente de si reventa realmente sucede o no)*" (Turunen & Pöyry, 2019, p. 552). Así, hay artículos de lujo que multiplican su valor cada año. Ejemplo de ello son las ediciones limitadas de bolso "*baguette*" de la marca italiana Fendi, que en los últimos 20 años han triplicado su valor original, o los bolsos de la casa francesa Hermès, cuyo valore de reventa llega a superar los 45 mil euros.

Por otro lado, la reventa de productos de lujo se ve inevitablemente relacionada con la circularidad de la industria de la moda como propuesta para hacer más sostenible el sector. "*Entre todos los modelos circulares de consumo de moda, la reventa se considera una solución fundamental a los enormes retos medioambientales provocados por la sobreproducción y el sobreconsumo de productos de moda y textiles*." (Liu, Xia, & Lang, C. , 2023, pág. 2). De hecho, tal y como afirman Lui, C. et al. (2023), gracias al auge de plataformas de reventa online de productos de lujo, el lujo concretamente se ha convertido en un pilar fundamental dentro de la reventa y la moda circular.

Por todo ello, podemos inferir que el valor potencial de reventa afectará positivamente a la intención de compra, y más cuanto mayor sea este valor.

H6: El valor de reventa influye de forma positiva y significativa sobre las actitudes hacia el producto.

Creatividad y emoción

Otro factor que no hemos de olvidar en este estudio es al aspecto emocional y la apelación a la creatividad que implica la compra de segunda mano de cualquier tipo de productos, con respecto a la compra por canales de directos.

De acuerdo con Turunen y Pöyry (2019), los sujetos de la muestra de uno de sus estudios definían la experiencia de compra de segunda mano, sobre todo en tiendas físicas, como una "*exploración emocionante, ya que nadie puede saber de antemano lo que se puede encontrar*" (Turunen & Pöyry, 2019, p. 552), a diferencia de la compra de artículos de primera mano. Así, determinan que el factor de compra recreativa ("*recreational shopping*") hace referencia a la "*participación activa en la actividad de compras: búsqueda y búsqueda de tesoros, experimentar un ambiente único, el espíritu del pasado, en lugar de comprar necesariamente cualquier cosa*" (Turunen & Pöyry, 2019, p. 552), y establecen su influencia sobre la intención de compra de artículos de lujo de segunda mano. Por tanto, con este factor, la actividad de la compra queda ligada a un aspecto de experiencia y emoción del consumidor, además de un esfuerzo cuya recompensa es una compra satisfactoria. Esto también lo identificaron Turunen y Leipämaa-Leskinen (2015) en uno de sus estudios, con el nombre "*thrill of the Hunt*", que se traduce por emoción de la caza o de la búsqueda. También Stolz (2022), hace referencia a este aspecto con la variable a la que denomina motivación creativa ("*creative motivation*"), que hace referencia a cómo los consumidores valoran más prendas "retro" o con significación en el ámbito de la moda y cómo esto es uno de los motores para comprar marcas de lujo, y no solo la calidad o la estética de los productos.

H8: La creatividad y emoción influye de forma positiva y significativa sobre las actitudes hacia el producto.

iii. Síntesis, creación del modelo

Una vez conceptualizados los factores motivacionales identificados por otros autores y explicadas las relaciones entre ellos, es momento de construir el modelo teórico que evaluaremos en este estudio.

Para ello, nos basaremos primeramente en el esquema proporcionado por la TPB de Azjen, en el que las actitudes, las normas subjetivas y el PBC influyen directamente sobre la intención de compra. Además, añadiremos la influencia sobre las actitudes de los factores: Conocimiento de marca, Sostenibilidad, Valor de reventa, Singularidad,

Creatividad y emociones, Nostalgia, Calidad-precio y Riesgo. También, los factores Sostenibilidad y Conocimiento de marca influirán sobre las normas subjetivas de forma indirecta, a través del consumo esnob y de arrastre. Por último, el factor riesgo influirá también sobre el PBC en nuestro modelo. Para sintetizar todo ello, establecemos las siguientes hipótesis.

H0: Las normas subjetivas influyen de forma positiva y significativa sobre la intención de compra

H1.1: El consumo de arrastre o *bandwagon consumption* influye de forma positiva y significativa sobre las normas subjetivas.

H1.2: El consumo esnob o *snob consumption* influye de forma positiva y significativa sobre las normas subjetivas.

H2: El PBC influye de forma positiva y significativa sobre la intención de compra de prendas de lujo de segunda mano.

H3: Las actitudes hacia el producto influye de forma positiva y significativa sobre la intención de compra de prendas de lujo de segunda mano.

H4: El conocimiento de marca influye de forma positiva y significativa sobre las actitudes hacia el producto.

H5: La sostenibilidad influye de forma positiva y significativa sobre las actitudes hacia el producto.

H6: El valor de reventa influye de forma positiva y significativa sobre las actitudes hacia el producto.

H7: La singularidad influye de forma positiva y significativa sobre las actitudes hacia el producto.

H8: La creatividad y emoción influye de forma positiva y significativa sobre las actitudes hacia el producto.

H9: La nostalgia influye de forma positiva y significativa sobre las actitudes hacia el producto.

H10: La calidad-precio influye de forma positiva y significativa sobre las actitudes hacia el producto.

H11: El riesgo influye de forma negativa y significativa sobre las actitudes hacia el producto.

H12.1: El conocimiento de la marca influye de forma positiva, significativa e indirecta sobre las normas subjetivas, a través del consumo de arrastre o *bandwagon consumption*.

H12.2: El conocimiento de la marca influye de forma negativa, significativa e indirecta sobre las normas subjetivas a través del consumo esnob.

H13: La sostenibilidad influye de forma positiva, significativa e indirecta sobre las normas subjetivas a través del consumo de arrastre o *bandwagon consumption*.

H14: La singularidad influye de manera positiva, significativa e indirecta sobre las normas subjetivas a través del consumo esnob.

H15: El riesgo influye de forma negativa y significativa sobre el control percibido o PBC.

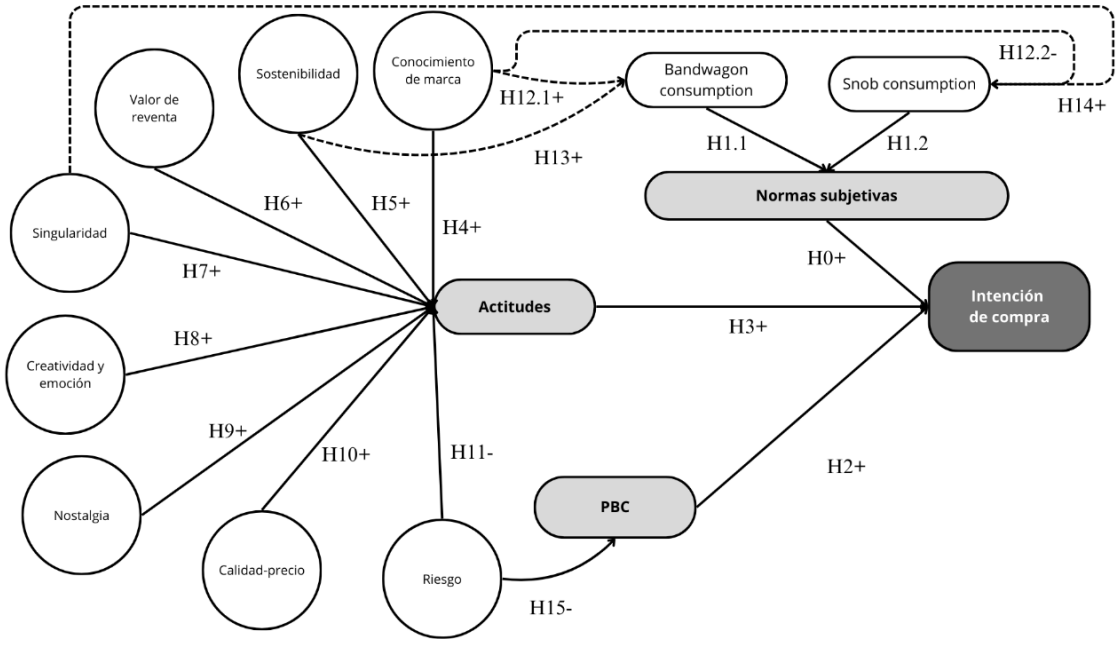


Ilustración 1: Esquema del modelo teórico propuesto, elaboración propia (líneas continuas: relación directa; líneas discontinuas: relación indirecta, interacción entre variables)

3. Investigación exploratoria

A. Encuestas abiertas

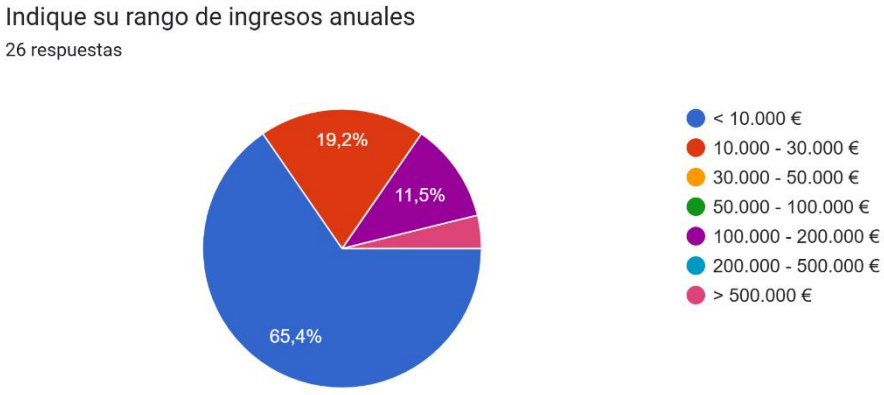
Para la investigación exploratoria, se diseñó una encuesta abierta en *Google Forms* con preguntas demográficas, una pregunta filtro, una opción múltiple y ocho preguntas abiertas. Las preguntas demográficas buscan caracterizar la muestra, mientras que la

pregunta filtro distingue entre consumidores y no consumidores de moda de lujo para analizar sus respuestas por separado.

Las preguntas abiertas exploran percepciones sobre el lujo, sus características, las emociones asociadas y las motivaciones de compra. Sus respuestas se analizarán mediante *text-mining* y *topic modeling* para identificar temas clave. La pregunta de opción múltiple evalúa la importancia de ciertos atributos en los productos de lujo. El objetivo es aportar información sobre el comportamiento del consumidor y complementar la literatura existente en este campo.

La muestra de encuestados es de 26 personas, de las cuales 15 eran mujeres y 11 hombres. Dentro de ellos, la distribución de edad se puede comprobar en el gráfico a continuación que eran primordialmente jóvenes de entre 18 y 25 años.

En la distribución de ingresos anuales hubo algo más de variedad, aunque la gran mayoría de los encuestados tenían una media de ingresos anuales inferior a los 10 mil euros.



En cuanto a las ocupaciones de los encuestados, encontramos que 17 de ellos son estudiantes (65,4% de la muestra), 4 son empleados por cuenta ajena del sector privado (15,4% de la muestra), un empleado por cuenta ajena del sector público, un autónomo,

un director o socio de empresa, un jubilado y un encuestado que respondió que tenía otra ocupación (representando cada uno de ellos un 3,8% de la muestra).

Por último, el 69% de los encuestados habían consumido artículos de moda de lujo con anterioridad a responder al formulario.

La primera pregunta de la encuesta es una pregunta filtro sobre si el encuestado ha consumido artículos de lujo con anterioridad (de cualquier tipo). En base a la respuesta a esta primera cuestión se tendrán en cuenta por separado las respuestas de ambos grupos para poder analizar si existen diferencias significativas en las asociaciones al concepto de lujo y los sentimientos relativos al mismo entre ambos grupos de encuestados.

A continuación, mostramos las *research questions* con las que se relacionan las preguntas de la encuesta, y el listado detallado de las preguntas se encuentra en el Anexo.

RQ1: Un mayor conocimiento de marcas de lujo está relacionado con asociaciones positivas hacia el concepto de lujo.

RQ2: Las personas que han consumido lujo con anterioridad conocen más marcas de lujo.

RQ3: ¿Qué significa el lujo para los encuestados? ¿Es compatible con el concepto de segunda mano?

RQ4: ¿Cuáles de las variables independientes del modelo teórico son consideradas como inherentes al concepto de lujo por los encuestados?

RQ5: ¿Qué sentimientos genera el concepto de lujo frente al consumir productos de lujo? ¿Existen diferencias entre los sentimientos de quienes han consumido lujo con anterioridad y quienes no?

RQ6: ¿Cuáles son las motivaciones principales para consumir productos de lujo? ¿Coinciden con las establecidas en el modelo teórico?

B. Análisis de los resultados con text-mining

Para el análisis de las respuestas de la encuesta, ha sido empleado el lenguaje de programación Python, y con él se han creado los modelos que se explican a continuación.

En este apartado nos centraremos en la creación y análisis del modelo de minería de texto y procesamiento de lenguaje natural (NLP) conocido como *Topic Modeling LDA (Latent*

Dirichlet Allocation). Este es un modelo de aprendizaje automático utilizado en *Topic Modeling*, una técnica de minería de texto que permite identificar temas latentes dentro de un conjunto de documentos. Este modelo asume que cada documento está compuesto por una mezcla de temas y que cada tema está formado por un conjunto de palabras con diferentes probabilidades de aparición. En el contexto de este estudio, LDA resulta especialmente útil para analizar las respuestas de las preguntas abiertas de la encuesta, extrayendo patrones temáticos recurrentes en las percepciones de los participantes sobre el lujo. Esto facilita la interpretación de grandes volúmenes de texto y aporta una visión estructurada de los principales conceptos mencionados.

A continuación, expondré brevemente los pasos seguidos para analizar las respuestas a la encuesta. El código empleado para el análisis se encuentra adjunto en la plataforma como anexo en este documento.

Para analizar las respuestas obtenidas en la encuesta, se ha aplicado un enfoque basado en técnicas de Procesamiento de Lenguaje Natural (NLP) con el objetivo de estructurar la información y extraer patrones significativos en la percepción del lujo. Dado que la encuesta distingue entre consumidores previos de moda de lujo y aquellos que no han adquirido estos productos, se ha segmentado el análisis en dos grupos: Grupo SÍ, conformado por encuestados con experiencia previa en el consumo de moda de lujo, y Grupo NO, compuesto por aquellos que no han consumido este tipo de productos. Esta segmentación ha permitido comparar los resultados y analizar diferencias en la percepción del lujo en función del nivel de familiaridad de los encuestados con estas marcas y productos.

El análisis comenzó dividiendo las respuestas en documentos individuales por pregunta, permitiendo un estudio separado según la naturaleza de cada una. Posteriormente, se llevó a cabo un proceso de limpieza y normalización de datos, asegurando coherencia en la escritura y facilitando su interpretación.

Dado que algunas preguntas no eran abiertas, no se pudieron analizar mediante el modelo Latent Dirichlet Allocation (LDA). En estos casos, se utilizó análisis de frecuencia y coocurrencias, permitiendo interpretar los datos sin forzar el uso de modelado de tópicos. Las preguntas cerradas (preguntas 1 y 6) se analizaron como opción múltiple, pues no contenían textos extensos. Para la primera pregunta, se corrigieron errores tipográficos, se eliminaron caracteres especiales y se estandarizaron las respuestas mediante un

diccionario de mapeo manual (por ejemplo, "luis vuitton" → "louis vuitton"). También se eliminaron términos irrelevantes y *stopwords*, y se ajustaron respuestas problemáticas que no podían procesarse correctamente.

Tras la normalización de datos, se realizó un análisis de frecuencia para determinar qué marcas y características eran más reconocidas. Se segmentaron respuestas en unidades individuales y se generó un *DataFrame* con la frecuencia de aparición de cada marca y característics. Posteriormente, los datos se visualizaron mediante gráficos de barras para facilitar su interpretación.

Además, se aplicó un análisis de coocurrencias para identificar qué marcas o características se mencionaban juntas en una misma respuesta. Se generaron combinaciones (parejas) de términos y se construyó una red de relaciones, donde cada nodo representaba una marca o factor y las conexiones reflejaban la frecuencia de mención conjunta. Se estableció un umbral de frecuencia mínima para evitar asociaciones poco representativas. Este análisis se replicó en los grupos SÍ y NO, comparando patrones entre consumidores y no consumidores de lujo.

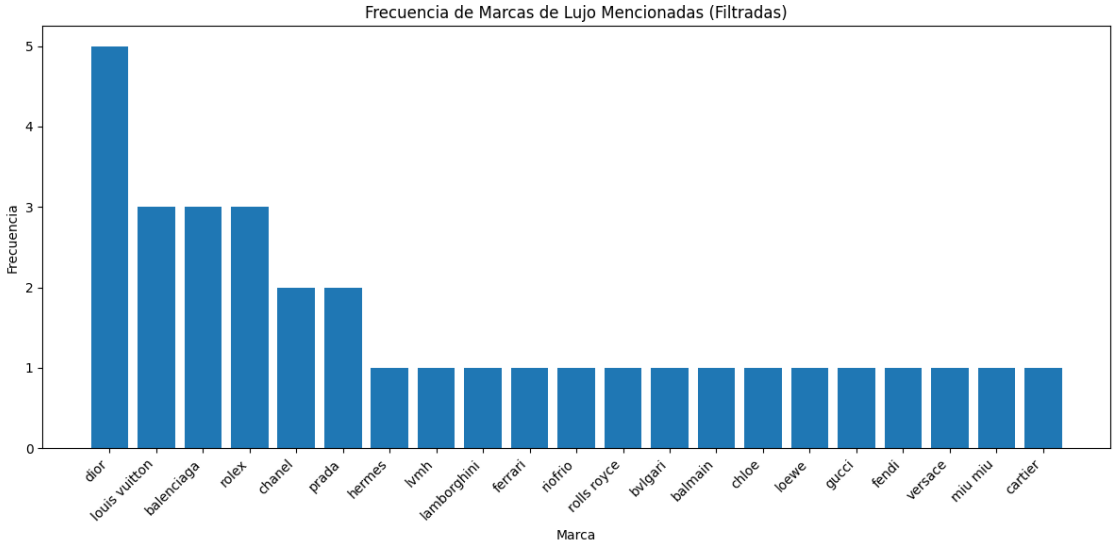
Para las preguntas abiertas, se utilizó el modelo LDA, permitiendo identificar temas predominantes en las respuestas. Se eliminaron *stopwords*, se normalizó el texto y se creó una matriz de documentos y términos. Se evaluó la coherencia semántica del modelo para definir en definitiva el número óptimo de tres temas por pregunta. Finalmente, los resultados se representaron mediante nubes de palabras, facilitando la identificación visual de los conceptos más mencionados en cada grupo.

Este enfoque permitió un análisis estructurado, combinando técnicas de procesamiento de datos y aprendizaje automático para extraer patrones significativos en la percepción del lujo y la segunda mano.

Ahora analizaremos los resultados de este análisis.

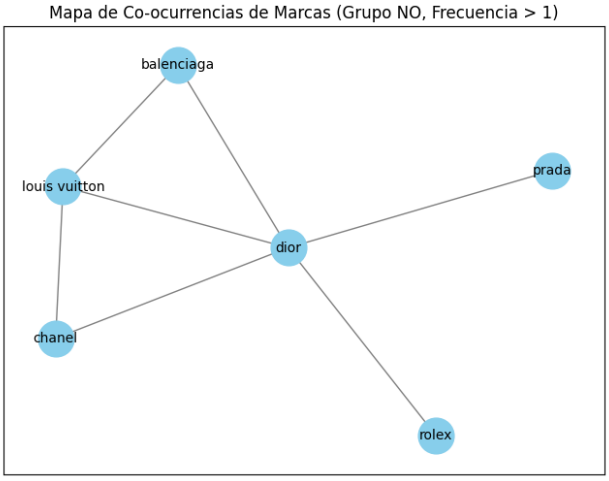
Pregunta 1: ¿Qué marcas de lujo conoces? Menciona al menos 3

Para esta pregunta, entre los encuestados que no habían consumido lujo con anterioridad, observamos que la marca mencionada con mayor frecuencia es Dior (5 menciones), seguida por Louis Vuitton, Balenciaga y Rolex (3 menciones cada una).



En cuanto a las coocurrencias, las marcas que más se mencionaron conjuntamente poreste grupo fueron:

```
Coocurrencias ordenadas por frecuencias (Grupo NO):
                          Par  Frecuencia
15       (dior, louis vuitton)           2
1          (balenciaga, dior)           2
4   (balenciaga, louis vuitton)           2
7              (chanel, dior)           2
10     (chanel, louis vuitton)           2
```



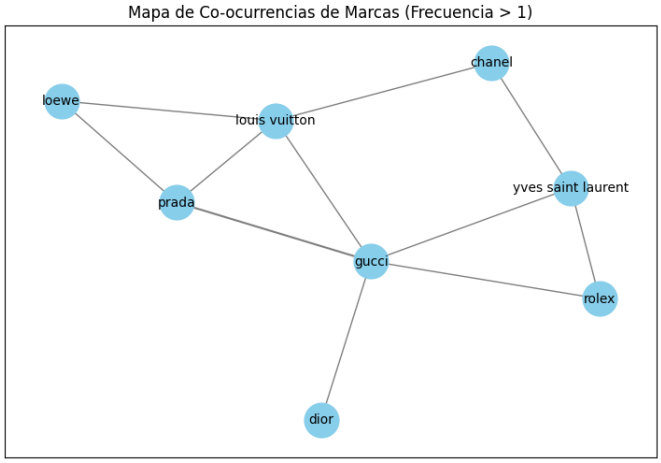
En el grupo de encuestados que sí habían consumido lujo antes, las marcas más mencionadas fueron Gucci y Louis Vuitton (7 menciones cada una), seguidas de Rolex y Loewe (6 menciones) y Chanel (5 menciones). Curiosamente, Balenciaga, que era una de las marcas más mencionadas entre quienes no habían consumido lujo, figura entre las menos mencionadas en este grupo. Además, el número de marcas mencionadas se ha visto incrementado considerablemente en relación con el grupo de no consumidores, sugiriendo que tienen un mayor conocimiento de marcas.

Las marcas mencionadas conjuntamente con mayor frecuencia fueron Gucci y Prada (probablemente por pertenecer al mismo grupo y contar con la misma directora creativa, cosa que los consumidores de lujo es posible que tengan presente).

Coocurrencias por orden de frecuencias:

	Par	Frecuencia
100	(gucci, prada)	3
0	(chanel, louis vuitton)	2
2	(chanel, yves saint laurent)	2
109	(louis vuitton, prada)	2
108	(gucci, louis vuitton)	2
..	...	...
39	(brunello cucinelli, vivienne westwood)	1
38	(brunello cucinelli, stellamcartney)	1
37	(brunello cucinelli, maxmara)	1
36	(brunello cucinelli, loro piana)	1
129	(chanel, loewe)	1

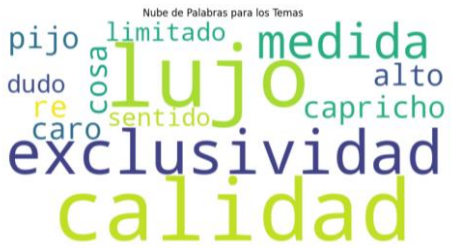
[130 rows x 2 columns]



Pregunta 2: ¿Qué significa para ti la palabra lujo?

En el grupo de no consumidores, los temas y la nube de palabras generados fueron los siguientes.

🔹 Temas generados:
Tema 1: 0.105*"exclusividad" + 0.105*"medida" + 0.104*"capricho" + 0.104*"lujo" + 0.027*"calidad"
Tema 2: 0.080*"pijo" + 0.080*"re" + 0.080*"cosa" + 0.080*"caro" + 0.080*"alto"
Tema 3: 0.088*"lujo" + 0.050*"limitado" + 0.050*"calidad" + 0.050*"sentido" + 0.050*"dudo"



Por lo que parece, el primer tema hace referencia a las características más objetivas (excepto capricho), el segundo tema a aspectos algo más subjetivos y el tercero a la duda ante el sentido de la calidad o de la exclusividad del lujo. En general, vemos como se

mencionan sobre todo la calidad, la exclusividad y el precio alto desde un tono de rechazo, denotado por palabras subjetivas como pijo o capricho.

En el caso del grupo SÍ, vemos que la connotación con la que se interpreta el término lujo es mucho más positiva, centrada en el valor y la calidad (primer tema generado), el carácter innecesario pero placentero para la vida del lujo (segundo tema) y la fabricación o manufactura como principales atributos del lujo (tercer tema).

```
Temas generados:
Tema 1: 0.140*"calidad" + 0.098*"alto" + 0.098*"caro" + 0.056*"valor" + 0.056*"exclusivo"
Tema 2: 0.063*"innecesario" + 0.063*"facilitar" + 0.063*"disfrutable" + 0.063*"vida" + 0.063*"elegancia"
Tema 3: 0.163*"exclusividad" + 0.127*"calidad" + 0.039*"excepcional" + 0.039*"fabricación" + 0.039*"manufactura"
```

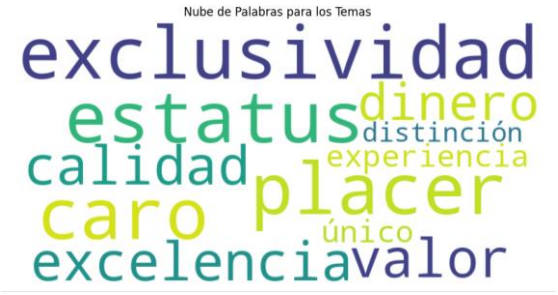
Pregunta 3: Definición de lujo en una sola palabra.

En el grupo de los no consumidores, encontramos temas centrados en la exclusividad o privilegio, el poder adquisitivo necesario para consumir lujo y la experiencia de consumo. El dinero es un factor muy comentado, tomando un enfoque subjetivo y sociológico al definir lujo.

```
Temas generados:
Tema 1: 0.712*"exclusivo" + 0.072*"adquisitivo" + 0.072*"experiencia" + 0.072*"dinero" + 0.072*"privilegio"
Tema 2: 0.362*"privilegio" + 0.362*"dinero" + 0.092*"exclusivo" + 0.092*"experiencia" + 0.092*"adquisitivo"
Tema 3: 0.362*"experiencia" + 0.362*"adquisitivo" + 0.092*"exclusivo" + 0.092*"dinero" + 0.092*"privilegio"
```

En el grupo de consumidores, sin embargo, además del enfoque sociológico (al que se le añaden términos como estatus o distinción), se habla también de placer, valor, excelencia y unicidad, dando al lujo una connotación superior y más subjetiva.

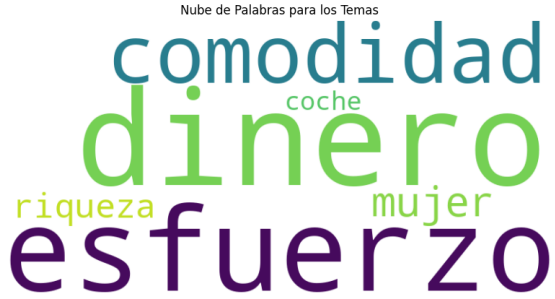
```
Temas generados:
Tema 1: 0.312*"exclusividad" + 0.218*"placer" + 0.124*"excelencia" + 0.124*"valor" + 0.032*"caro"
Tema 2: 0.241*"calidad" + 0.240*"estatus" + 0.137*"dinero" + 0.137*"experiencia" + 0.035*"exclusividad"
Tema 3: 0.383*"caro" + 0.153*"único" + 0.153*"distinción" + 0.039*"estatus" + 0.039*"placer"
```



Pregunta 4: ¿Con qué asocias tú el lujo?

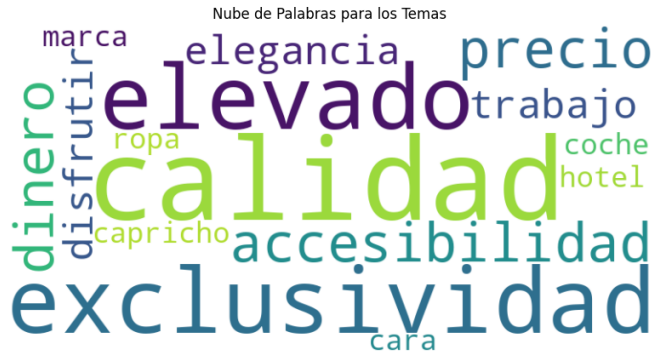
En el grupo de no consumidores de lujo, las respuestas a estas preguntas se centraron en el dinero y lo que permite conseguir (mujeres, coches, etc.). También se menciona el esfuerzo necesario para conseguir ese dinero y la comodidad de tener los recursos para ser consumidor de lujo.

```
Temas generados:
Tema 1: 0.169*"dinero" + 0.168*"esfuerzo" + 0.168*"comodidad" + 0.167*"riqueza" + 0.164*"mujer"
Tema 2: 0.223*"coche" + 0.223*"mujer" + 0.221*"comodidad" + 0.221*"esfuerzo" + 0.057*"dinero"
Tema 3: 0.554*"dinero" + 0.221*"riqueza" + 0.057*"esfuerzo" + 0.056*"comodidad" + 0.056*"mujer"
```



Entre los consumidores de lujo, uno de los principales temas se centró en la difícil accesibilidad debido al precio. También fueron relevantes temas enfocados a asociaciones positivas con productos de lujo como la elegancia o el disfrute. Por último, coincidiendo con el otro grupo, también se asocian con el lujo cosas que se pueden conseguir con dinero como hoteles, ropa, coches y cosas de marca.

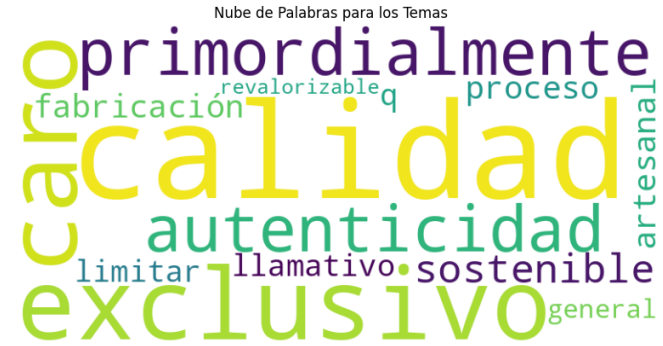
```
Temas generados:
Tema 1: 0.081*"calidad" + 0.080*"exclusividad" + 0.046*"elevado" + 0.046*"accesibilidad" + 0.046*"preci
Tema 2: 0.216*"dinero" + 0.116*"elegancia" + 0.067*"disfrutir" + 0.066*"trabajo" + 0.066*"capricho"
Tema 3: 0.054*"hotel" + 0.054*"ropa" + 0.053*"coche" + 0.053*"marca" + 0.053*"cara"
```



Pregunta 5: ¿Qué características debe tener, en tu opinión, un artículo de lujo?

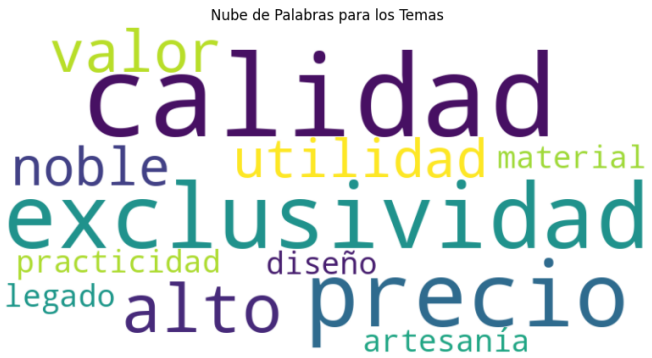
Entre los no consumidores de lujo, las características mencionadas principalmente fueron relativas a lo que tradicionalmente se asocia al lujo: calidad, exclusividad, autenticidad y precio alto. También surgieron temas centrados en el origen del producto, mencionando la fabricación, el proceso, la artesanía y la sostenibilidad. Por último, surge un tema con un carácter algo más específico, relacionando el lujo con lo llamativo y limitado, hablando también de revalorización.

```
Temas generados:
Tema 1: 0.252*"calidad" + 0.198*"exclusivo" + 0.084*"caro" + 0.084*"primordialmente" + 0.084*"autenticidad"
Tema 2: 0.130*"calidad" + 0.106*"sostenible" + 0.104*"fabricación" + 0.104*"artesanal" + 0.104*"proceso"
Tema 3: 0.129*"llamativo" + 0.129*"limitar" + 0.129*"q" + 0.129*"general" + 0.128*"revalorizable"
```



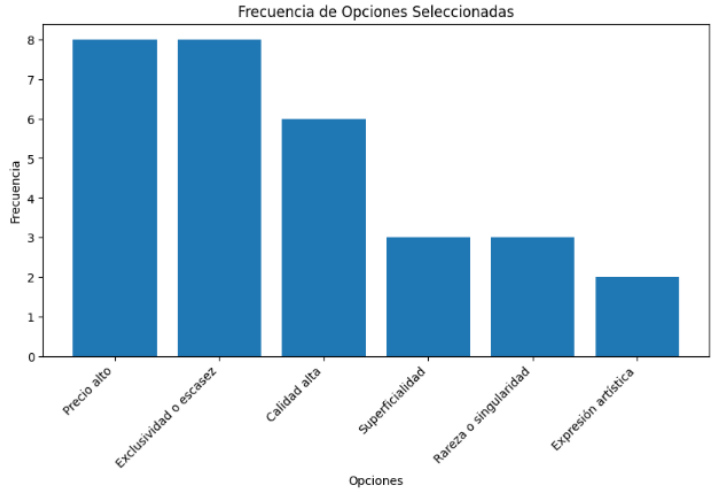
En cambio, los consumidores de lujo introdujeron características como legado, haciendo referencia a la historia de la marca o producto, e incluso practicidad y utilidad. Esto sugiere que los consumidores de lujo encuentran en estos productos un factor de utilidad que los no consumidores no identifican de primeras. En los temas extraídos se repiten atributos como calidad, exclusividad, materiales nobles, precio alto, valor, artesanía, etc. Por tanto, estas últimas podríamos decir que son las características comúnmente asociadas a productos de lujo, independientemente de si se es consumidor de lujo o no.

```
Temas generados:
Tema 1: 0.187*"calidad" + 0.177*"alto" + 0.153*"exclusividad" + 0.065*"precio" + 0.038*"valor"
Tema 2: 0.058*"utilidad" + 0.058*"noble" + 0.058*"practicidad" + 0.058*"diseño" + 0.058*"material"
Tema 3: 0.188*"calidad" + 0.064*"exclusividad" + 0.039*"precio" + 0.038*"artesanía" + 0.038*"legado"
```



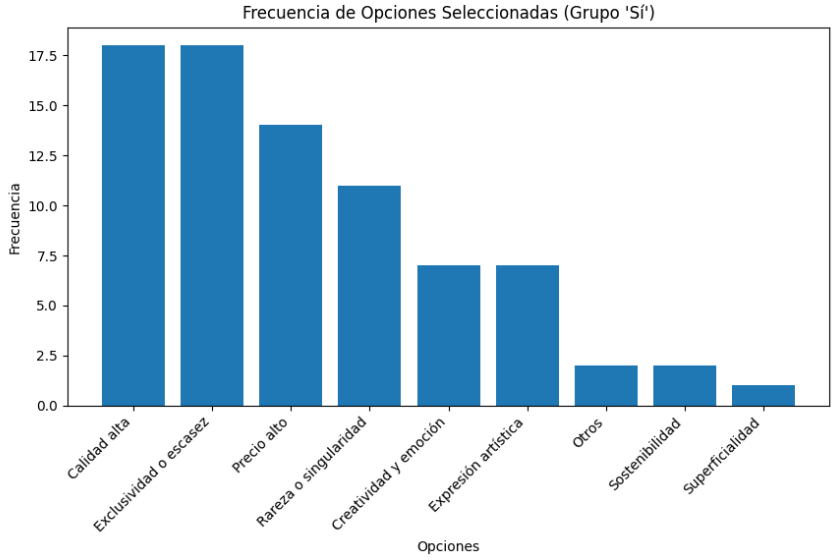
Pregunta 6: ¿Cuáles de estos factores consideras que están relacionados con el lujo?

En el grupo de no consumidores, los factores más mencionados (unánimes) fueron precio alto y exclusividad o escasez, y el menos mencionado fue expresión artística. Esto sugiere que los encuestados no consumidores, no consideran que exista una relación estrecha entre lujo y arte, y reconocen la exclusividad asociada al precio.



En cambio, en el grupo de consumidores, los factores mencionados por todos no incluían precio alto, sino calidad alta y exclusividad o escasez. Esto nos invita a pensar que el consumidor de lujo valora estos atributos por encima del precio del producto. El menos mencionado fue superficialidad, que sí que fue más mencionado en otro grupo. Sin embargo, otra característica menos mencionada en el grupo anterior, rareza o singularidad, es la cuarta más mencionada entre los consumidores de lujo.

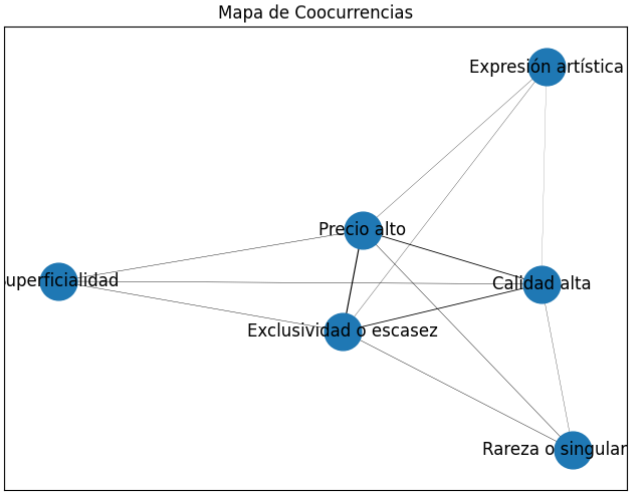
En general, esto nos invita a pensar que los no consumidores perciben el lujo como algo frívolo, caro y reservado a unos pocos, mientras que los consumidores de lujo son más sensibles a la calidad, singularidad y forma de expresión artística del lujo.



Entre las coocurrencias de estos factores, en el grupo de los no consumidores, destacó la combinación de precio alto y exclusividad o escasez, seguida de las combinaciones de calidad alta y exclusividad o escasez, y calidad alta y precio alto. Se genera así, un triángulo, como podemos ver en el mapa de coocurrencias, entre estos tres factores clave a la hora de definir un artículo de lujo (de acuerdo con este grupo).

Coocurrencias más frecuentes:

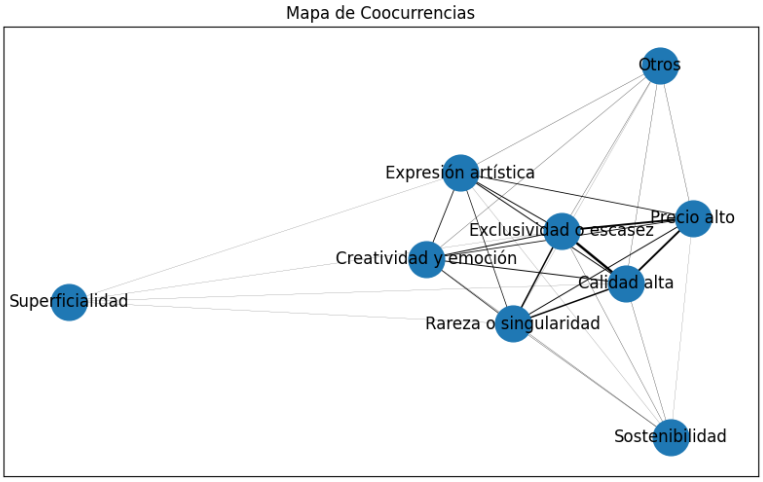
	Par	Frecuencia
3	(Exclusividad o escasez, Precio alto)	8
0	(Calidad alta, Exclusividad o escasez)	6
1	(Calidad alta, Precio alto)	6
2	(Calidad alta, Superficialidad)	3
4	(Exclusividad o escasez, Superficialidad)	3



Por otro lado, los consumidores de lujo mencionaron de forma conjunta, sobre todo, calidad alta y exclusividad o escasez. Las siguientes combinaciones más mencionadas fueron calidad alta y precio alto, y exclusividad o escasez y precio alto, estableciendo el mismo triángulo que grupo de no consumidores. Podemos comprobar que superficialidad es la característica más alejada en el mapa.

Coocurrencias más frecuentes:

	Par	Frecuencia
1	(Calidad alta, Exclusividad o escasez)	18
4	(Calidad alta, Precio alto)	14
11	(Exclusividad o escasez, Precio alto)	14
15	(Calidad alta, Rareza o singularidad)	11
19	(Exclusividad o escasez, Rareza o singularidad)	11

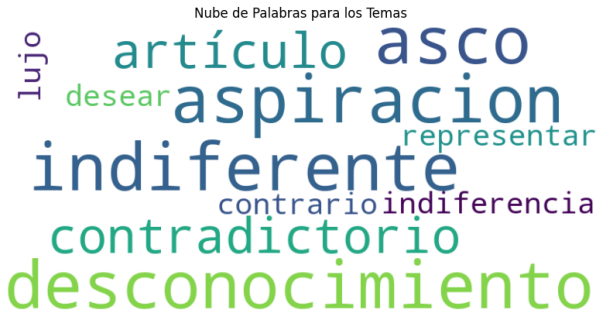


Por tanto, ambos grupos de encuestados asocian a lujo, sobre todo, calidad alta, precio alto y exclusividad, aunque en diferente orden.

Pregunta 7: ¿Qué sentimientos te genera el concepto de lujo?

Aquí se observa claramente una polarización de sentimientos en el grupo de no consumidores. Por un lado, observamos sentimientos como deseo y aspiración, contrapuestos al asco y la indiferencia. Son los mismos encuestados los que mencionan el carácter contradictorio de sus sentimientos hacia el concepto de lujo, e incluso desconocimiento hacia lo que sienten.

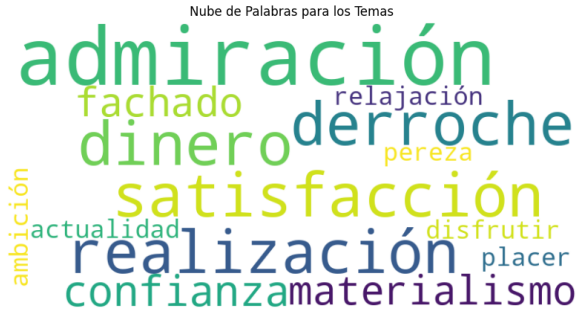
Temas generados:
Tema 1: 0.089*"contradictorio" + 0.089*"artículo" + 0.089*"representar" + 0.089*"contrario" + 0.089*"lujo"
Tema 2: 0.166*"desear" + 0.166*"desconocimiento" + 0.166*"asco" + 0.042*"indiferente" + 0.042*"aspiracion"
Tema 3: 0.189*"indiferencia" + 0.189*"aspiracion" + 0.049*"indiferente" + 0.048*"asco" + 0.048*"desconocimiento"



En cambio, en los consumidores de lujo no encontramos esta contradicción tan clara. En uno de los temas, encontramos principalmente sentimientos como ambición, placer, admiración y disfrute, siendo todos sentimientos positivos con carácter aspiracional. También se pueden relacionar con esto las menciones de satisfacción, confianza,

realización y relajación. Aunque en menor medida que en otro grupo, también encontramos sentimientos negativos asociados a conceptos como materialismo, derroche, e incluso pereza.

```
Temas generados:
Tema 1: 0.100*"dinero" + 0.100*"satisfacción" + 0.100*"realización" + 0.100*"derroche" + 0.100*"admiración"
Tema 2: 0.087*"confianza" + 0.087*"materialismo" + 0.087*"fachado" + 0.087*"relajación" + 0.087*"pereza"
Tema 3: 0.123*"ambición" + 0.120*"placer" + 0.069*"admiración" + 0.069*"disfrutir" + 0.069*"actualidad"
```



En general, podemos ver que el concepto de lujo, genera en ambos grupos sentimientos positivos aspiracionales, a la vez que sentimientos de desagrado (aunque en diferente proporción).

Pregunta 8: ¿Qué sentimientos te genera consumir lujo?

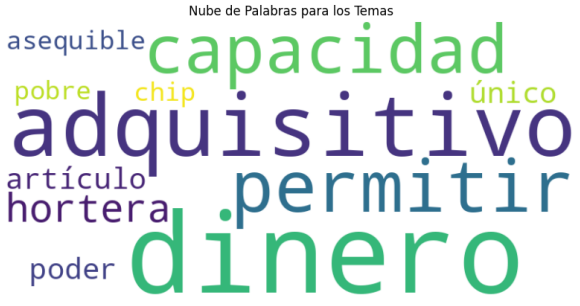
Esta pregunta de la encuesta solo se le hizo a los encuestados que pertenecían al grupo de consumidores de lujo. Entre los principales sentimientos extraídos se encuentran alegría o felicidad, satisfacción personal, disfrute o placer y culpa. Es decir, los consumidores de lujo en general tienen sentimientos positivos al consumir lujo, que se ven interrumpidos por la culpa. Cabe destacar menciones a la autoestima (elevada al consumir lujo), el sentirse especial, y el confort o comodidad.

```
Temas generados:
Tema 1: 0.161*"placer" + 0.113*"disfrutir" + 0.113*"culpa" + 0.113*"alegrio" + 0.113*"felicidad"
Tema 2: 0.178*"alegría" + 0.072*"satisfacción" + 0.071*"adquirirlo" + 0.071*"personal" + 0.071*"precio"
Tema 3: 0.085*"autoestima" + 0.085*"falso" + 0.085*"coqueto" + 0.085*"confort" + 0.085*"especial"
```

Pregunta 9: ¿Cuáles son las razones para que consumas/no consumas lujo?

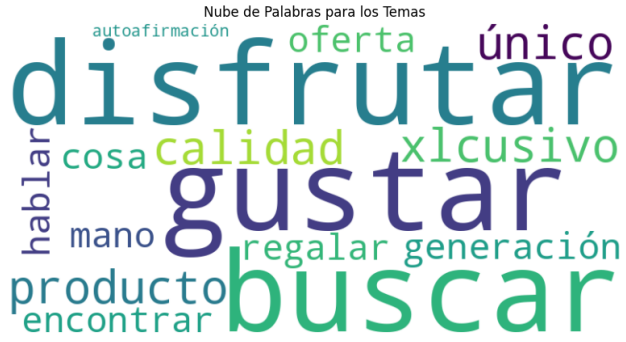
En el grupo de no consumidores, la principal razón es la falta de poder adquisitivo, que viene dada por términos como dinero, pobre, capacidad, asequible, etc. Por otro lado, vemos que se menciona hortera, aludiendo al estilo o carácter llamativo de algunos artículos de lujo. Hay incluso quien menciona que se pueden encontrar artículos únicos por un precio menor. Por tanto, vemos que hay argumentos para no consumir lujo como la falta de poder adquisitivo, que indican que, si se tuviera la posibilidad, el encuestado sería consumidor. Sin embargo, otros sugieren que, aunque se dieran las condiciones necesarias, no están interesados en consumir lujo.

```
Temas generados:
Tema 1: 0.337*"dinero" + 0.117*"hortera" + 0.034*"adquisitivo" + 0.034*"permitir" + 0.034*"capacidad"
Tema 2: 0.122*"artículo" + 0.070*"dinero" + 0.069*"poder" + 0.069*"único" + 0.069*"asequible"
Tema 3: 0.256*"adquisitivo" + 0.102*"chip" + 0.102*"pobre" + 0.102*"capacidad" + 0.102*"permitir"
```



Por otro lado, las razones de los consumidores de lujo para continuar siéndolo se basan sobre todo en temas como el disfrute, la calidad, la autoafirmación e incluso el poder regalarlos o pasarlos entre generaciones, dándole a los artículos de lujo un carácter hereditario y emocional.

```
Temas generados:
Tema 1: 0.103*"disfrutar" + 0.055*"gustar" + 0.055*"buscar" + 0.032*"producto" + 0.032*"xlcusivo"
Tema 2: 0.104*"calidad" + 0.044*"único" + 0.044*"generación" + 0.025*"cosa" + 0.025*"hablar"
Tema 3: 0.045*"encontrar" + 0.045*"oferta" + 0.045*"regalar" + 0.045*"mano" + 0.045*"autoafirmación"
```



4. Investigación confirmatoria

A. Encuestas cerradas

Para validar el modelo teórico y evaluar las hipótesis, se diseñó una encuesta confirmatoria con preguntas sobre las variables independientes, la variable dependiente y sus interacciones.

La primera pregunta fue un filtro para distinguir entre encuestados que han consumido productos de lujo de segunda mano y aquellos que no. A diferencia del estudio exploratorio, esta encuesta especificó el origen de los productos como de segunda mano, ya que su objetivo era comprobar la aplicabilidad del modelo teórico en este contexto.

Tras la pregunta filtro, se incluyeron 14 preguntas en escala Likert (1 a 7) para medir el nivel de acuerdo con afirmaciones relacionadas con las variables del modelo, como se muestra en la siguiente tabla.

VARIABLE	COMPONENTE	AFIRMACIÓN
Intención de compra	Intencion_Compra	Me interesa comprar artículos de lujo de segunda mano
Bandwagon consumption	BC1	Suelo estar atento a las tendencias de moda
	BC2	Me atrae la idea de adquirir productos de lujo populares entre mis amigos o conocidos
	BC3	Suelo dejarme llevar por las tendencias y comprar lo que está de moda
	BC4	Creo que comprar artículos de lujo que están en tendencia me hace sentir más integrado/a socialmente
	BC5	Considero que poseer prendas de lujo mejora mi estatus
	BC6	Me interesa consumir productos de lujo con logos o en los que la marca sea fácilmente reconocible
Snob Consumption	SC1	Prefiero comprar prendas que pocas personas tienen
	SC2	Me atrae la idea de destacar/diferenciarme de mi entorno por mi forma de vestir y mi estilo personal
	SC3	Suelo dejarme llevar por las tendencias y comprar lo que está de moda
	SC4	La exclusividad de un producto de lujo es un factor clave en mi decisión de compra.
	SC5	En caso de consumir artículos de lujo, me interesa consumir ediciones especiales por encima de productos
	SC6	En caso de comprar artículos de lujo, me interesa más adquirir piezas que representen rareza o escasez.
	SC7	Me interesa consumir productos de lujo sin logos o en los que la marca no es fácilmente reconocible

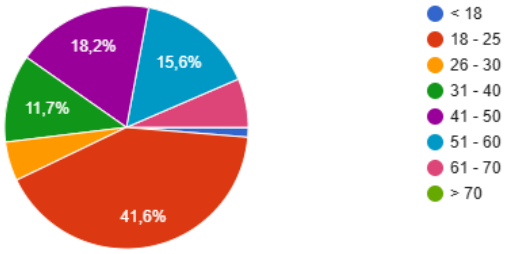
Actitudes	A1	Tengo una percepción positiva hacia las prendas de lujo de segunda mano
	A2	Creo que comprar prendas de lujo de segunda mano es una elección inteligente en términos de calidad-precio.
	A3	Creo que comprar prendas de lujo de segunda mano es una elección sostenible.
	A4	Suelo mirar plataformas de segunda mano cuando quiero adquirir un producto de lujo
	A5	Considero que en las plataformas de segunda mano hay prendas de lujo más originales o especiales que en las alternativas de primera mano
Normas subjetivas	SN1	Mis decisiones de compra se suelen ver afectadas por las opiniones de mis familiares y amigos
	SN2	Considero que comprar lujo de segunda mano es algo aceptado positivamente en mi entorno
	SN3	Siento presión social para comprar productos de lujo de segunda mano en lugar de productos nuevos
	SN4	La recomendación de influencers o figuras públicas me anima a considerar el lujo de segunda mano.
PBC	PBC1	Considero que es fácil acceder a prendas de lujo de segunda mano en el mercado actual
	PBC2	Me siento capaz de identificar prendas de lujo de segunda mano auténticas y de calidad
	PC3	Creo que tengo suficiente información para tomar decisiones de compra en el mercado de lujo de segunda
	PC4	Considero que la falta de información sobre el estado o la autenticidad de productos de lujo de segunda mano me supone un riesgo para la compra
	PC5	Considero que la información y la facilidad de acceso a comprar lujo de segunda mano incrementa mi intención de compra
Sostenibilidad	So1	Considero que comprar prendas de lujo es más sostenible que consumir prendas fast-fashion
	So2	Creo que optar por segunda mano es una forma de apoyar la sostenibilidad en la moda
	So3	Me preocupa el impacto medioambiental de mis decisiones de compra de moda
	So4	Valoro más un producto de lujo si está vinculado a prácticas sostenibles
	So5	Considero que la durabilidad de un producto de lujo lo convierte en una elección sostenible
	So6	Siento cierta presión social para tomar decisiones de compra de moda sostenibles

Conocimiento de marca	CM1	Reconozco fácilmente las marcas de lujo en el mercado de segunda mano
	CM2	Tengo un conocimiento profundo sobre las marcas de lujo y sus productos
	CM3	La reputación de una marca de lujo influye en mi decisión de comprar sus productos de segunda mano
	CM4	Me siento más seguro/a comprando prendas de lujo de segunda mano cuando provienen de una marca reconocida o que conozco bien
	CM5	Me atrae más comprar productos de lujo de segunda mano de marcas reconocidas o que conozco bien
Singularidad	Si1	Comprar prendas de segunda mano me permite expresar mi individualidad
	Si2	Valoro las prendas de lujo de segunda mano porque me permiten diferenciarme de los demás
	Si3	Me atrae la idea de poseer productos de lujo únicos que no están disponibles en el mercado tradicional
	Si4	Las ediciones limitadas de segunda mano productos de lujo me atraen más que el producto original
Calidad-Precio	CP1	Las prendas de lujo de segunda mano ofrecen una buena relación calidad-precio
	CP2	El precio es un factor muy relevante para mí a la hora de consumir productos de lujo
	CP3	Me atrae la idea comprar productos de lujo a un precio inferior que en el mercado tradicional
	CP4	La calidad de los productos de lujo de segunda mano justifica su precio
	CP5	Considero que el lujo de segunda mano es una opción más rentable que el lujo nuevo
Nostalgia	No1	Me interesa adquirir prendas características una época
	No2	Suelo sentir nostalgia por objetos que ya no se
	No3	Me atraen los productos de lujo de segunda mano por su historia y tradición
	No4	Valoro las prendas de lujo de segunda mano porque evocan emociones y recuerdos especiales
Riesgo	Ri1	Considero que comprar lujo de segunda mano conlleva un nivel de riesgo elevado
	Ri2	Me disuade de comprar productos de lujo de segunda mano el hecho de que puedan ser falsos
	Ri3	Percibo un riesgo alto asociado al estado de las prendas de lujo de segunda mano
	Ri4	Creo que existe poca transparencia en las transacciones de lujo de segunda mano
	Ri5	Suelo tomar riesgos con mi dinero
	Ri6	Evitaría comprar lujo de segunda mano debido a la falta de garantías de autenticidad y/o buen estado

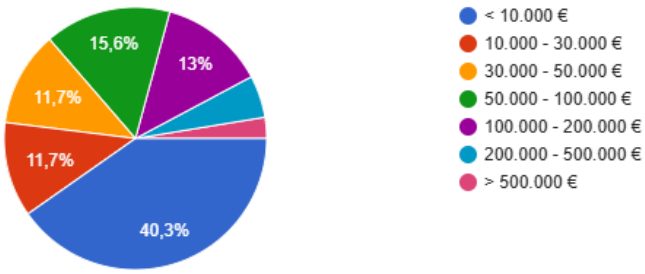
Valor de reventa	VR1	Me intereso por el valor de reventa de productos de lujo
	VR2	Me interesa comprar productos de lujo de segunda mano por la posibilidad de revenderlos en el futuro por un precio superior al que lo compré
	VR3	Considero que el valor de reventa de los productos de lujo se mantiene o incluso crece
	VR4	Considero que comprar productos de lujo de segunda mano es una buena inversión por su valor de reventa
Creatividad y emoción	CE1	Comprar prendas de lujo de segunda mano es una experiencia emocionante para mí
	CE2	Me siento creativo/a al buscar y elegir productos de segunda mano.
	CE3	Experimento emociones positivas al encontrar un producto que me gusta después de una búsqueda en plataformas de
	CE4	Considero estimulante el proceso de encontrar piezas únicas de lujo de segunda

La lista completa de preguntas se encuentra en el anexo del documento.

La muestra tomada para esta encuesta fue de 101 personas, de las cuales el 68% aproximadamente fueron mujeres, y el resto hombres. Evaluando los rangos de edad, la mayor parte tenían entre 18 y 25 años seguidos de los rangos de edad de 41 a 50 y 51 a 60 años, como se muestra en el gráfico.



En cuanto al rango de ingresos anuales, la mayor parte de los encuestados pertenecían al rango de menos de diez mil euros anuales. El gráfico muestra una distribución similar a la de los rangos de edad. Casi el 80 % de los encuestados tiene un salario anual inferior a 100.000 euros, lo que puede limitar su consumo de lujo, pero favorecer la compra de artículos de lujo de segunda mano.



Por último, algo más del 36% de los encuestados son estudiantes, y alrededor del 31% son empleados por cuenta ajena del sector privado. El 14% son autónomos o profesionales independientes y casi el 12% son empresarios o dueños de un negocio. El resto de las ocupaciones apenas se ven representadas. Esto podría suponer una limitación para el estudio.

B. Análisis de los factores del modelo teórico con regresión múltiple

Este estudio emplea modelos de regresión múltiple para analizar los factores que influyen en la intención de compra de productos de lujo de segunda mano. Siguiendo la *Theory of Planned Behavior* (TPB), la metodología se desarrolla en dos fases. La primera examina constructos intermedios que afectan la intención de compra; la segunda sintetiza los hallazgos en un modelo final que integra todos los factores identificados.

En la primera fase, se construyen tres modelos independientes, cada uno con una variable dependiente clave de la TPB: actitudes hacia la compra, normas subjetivas y PBC. Estos modelos descomponen el proceso de decisión del consumidor, permitiendo analizar cómo cada variable influye en la intención de compra.

El primer modelo estudia las actitudes del consumidor, es decir, su percepción positiva o negativa hacia el lujo de segunda mano. Se incluyen como variables independientes factores como conocimiento de marca, sostenibilidad, valor de reventa, singularidad, creatividad y emoción, calidad-precio, nostalgia y riesgo percibido. Este análisis permite comprender qué elementos contribuyen a una percepción favorable del mercado de segunda mano.

El segundo modelo analiza el impacto de las normas subjetivas, es decir, la influencia de factores sociales en la decisión de compra. La variable dependiente es la presión social percibida para adquirir productos de lujo de segunda mano. Como variables independientes se consideran el consumo de arrastre y consumo snob, y se incluyen interacciones con otros factores, como la relación entre sostenibilidad y consumo de arrastre o entre singularidad y consumo snob, lo que permite capturar efectos indirectos que enriquecen la comprensión del impacto social en la compra de lujo de segunda mano.

El tercer modelo examina el PBC, que refleja la percepción del consumidor sobre la facilidad o dificultad de adquirir productos de lujo de segunda mano. La única variable independiente en este caso es el riesgo percibido, ya que afecta la confianza del consumidor en la autenticidad y calidad del producto, lo que a su vez puede influir en su decisión de compra.

Finalmente, sobre la base de estos tres modelos, se desarrolla un modelo final, donde la variable dependiente es la intención de compra y las variables independientes son actitudes, normas subjetivas y control percibido del comportamiento. Este modelo proporciona una visión integral de cómo se combinan percepciones individuales, influencias sociales y percepción de control en la toma de decisiones del consumidor.

Este enfoque permite descomponer el fenómeno de la compra de lujo de segunda mano y garantizar un análisis estructurado y riguroso del comportamiento del consumidor mediante modelos estadísticos robustos.

Ahora pasaremos a analizar los resultados obtenidos.

En el proceso de la creación de los modelos, calculamos el Alpha de Cronbach de cada una de las afirmaciones establecidas para las diferentes variables del modelo. Cuanto más cercano a 1 su valor, mayor será la fiabilidad o consistencia interna del conjunto de ítems, evaluando si los elementos de una misma variable miden realmente el mismo concepto. Obtuvimos la siguiente matriz, en la que se ve una buena consistencia interna. Traté de mejorar el Alpha de las Normas Subjetivas eliminando afirmaciones poco significativas, pero dado que la mejora no era considerable, decidí mantener las afirmaciones originales.

Bandwagon Consumption	Snob Consumption	Normas Subjetivas	Actitudes	PBC	Conocimiento de Marca	Sostenibilidad	Valor de Reventa	Singularidad	Creatividad y Emoción	Nostalgia	Calidad-Precio	Riesgo
0.837567	0.887752	0.65479	0.876893	0.787425	0.890783	0.858459	0.874534	0.883316	0.952845	0.910008	0.904301	0.855256

Se calcularon las cargas factoriales (*CFA Loadings*) para evaluar la relación entre cada ítem (pregunta) y su factor latente (constructo). Al detectar valores superiores a 1, indicando colinealidad elevada, se creó una matriz de correlaciones y se eliminaron afirmaciones con correlaciones mayores a 0,7, reduciendo la multicolinealidad y mejorando la precisión del modelo.

Después, se pasó a construir los tres modelos con cada uno de los componentes de la TPB como variables dependientes. Para analizar sus resultados, estableceremos como nivel de

significación del P-valor un 8% (es decir, la influencia de una variable del modelo se considerará significativa con un P-valor inferior 0,08).

El primer modelo, tal y como lo habíamos diseñado, inicialmente quedaba así.

```
Evaluación del modelo para Actitudes:
Mean Squared Error (MSE): 1.1865
                            OLS Regression Results
==============================================================================
Dep. Variable:              Actitudes   R-squared:                       0.687
Model:                            OLS   Adj. R-squared:                  0.660
Method:                 Least Squares   F-statistic:                     25.27
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           3.71e-20
Time:                        09:48:20   Log-Likelihood:                -135.73
No. Observations:                 101   AIC:                             289.5
Df Residuals:                      92   BIC:                             313.0
Df Model:                           8
Covariance Type:            nonrobust
===========================================================================================
                              coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------
Intercept                   0.6245      0.419      1.491      0.139      -0.207       1.456
Conocimiento_Marca          0.0613      0.089      0.692      0.491      -0.115       0.237
Sostenibilidad              0.1003      0.092      1.087      0.280      -0.083       0.284
Valor_Reventa               0.0655      0.076      0.861      0.392      -0.086       0.217
Singularidad                0.1828      0.103      1.779      0.079      -0.021       0.387
Creatividad_Emocion         0.3624      0.078      4.676      0.000       0.209       0.516
Calidad_Precio              0.2830      0.105      2.706      0.008       0.075       0.491
Nostalgia                  -0.0929      0.090     -1.029      0.306      -0.272       0.086
Riesgo                     -0.0711      0.081     -0.877      0.383      -0.232       0.090
==============================================================================
Omnibus:                        5.151   Durbin-Watson:                   1.938
Prob(Omnibus):                  0.076   Jarque-Bera (JB):                6.183
Skew:                           0.221   Prob(JB):                       0.0454
Kurtosis:                       4.129   Cond. No.                         52.7
==============================================================================
```

El MSE obtenido indica un error aceptable pero no óptimo, con un margen de 1,19 puntos en una escala de Likert del 1 al 7. El estadístico R^2 muestra que el modelo explica el 68,7% de la variabilidad en actitudes, reflejando una buena capacidad predictiva. Además, el estadístico F de 25,27 confirma la significancia global del modelo.

En cuanto a las variables, los P-valores cercanos a 0 indican que creatividad_emoción y calidad_precio tienen una influencia positiva y significativa en actitudes, por lo que no se rechazan las hipótesis H8 y H10. Asimismo, la singularidad también resulta significativa, manteniendo la hipótesis H7. Sin embargo, las variables conocimiento_marca y riesgo no muestran efectos significativos, lo que lleva a rechazar H4 y H11. También se rechazan H5, H6 y H9, ya que sostenibilidad, valor de reventa y nostalgia no influyen significativamente en actitudes.

Se evaluó la posibilidad de mejorar el modelo eliminando variables no significativas y elevando al cuadrado las más relevantes, pero los cambios no fueron sustanciales, por lo que se mantiene el modelo original.

En el segundo modelo, inicialmente, se observaron los siguientes resultados.

```
Evaluación del modelo para PBC:
Mean Squared Error (MSE): 1.7315
                            OLS Regression Results
==============================================================================
Dep. Variable:                    PBC   R-squared:                       0.142
Model:                            OLS   Adj. R-squared:                  0.133
Method:                 Least Squares   F-statistic:                     16.38
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           0.000103
Time:                        09:48:20   Log-Likelihood:                -167.93
No. Observations:                 101   AIC:                             339.9
Df Residuals:                      99   BIC:                             345.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.5115      0.394      6.382      0.000       1.731       3.292
Riesgo         0.3963      0.098      4.048      0.000       0.202       0.591
==============================================================================
Omnibus:                        1.518   Durbin-Watson:                   2.171
Prob(Omnibus):                  0.468   Jarque-Bera (JB):                1.282
Skew:                          -0.093   Prob(JB):                        0.527
Kurtosis:                       2.481   Cond. No.                         13.0
==============================================================================
```

El error en este caso es algo más elevado, y el R^2 es relativamente bajo (14,2%), por lo que la variabilidad del PBC apenas viene explicada por el modelo. Por otro lado, el estadístico F de 16,38 parece indicar que el modelo es globalmente significativo.

En cuanto a la variable independiente del modelo, parece ser que el Riesgo influye significativamente y de forma positiva en el PBC. Por tanto, rechazaremos la hipótesis H15, que sugería influencia significativa negativa.

Como la capacidad predictiva del modelo era baja, se intentó mejorarlo elevando al cuadrado la variable Riesgo, pero sin resultados significativos, por lo que se mantuvo el modelo original.

El tercer modelo, inicialmente presentó estos resultados.

```
Evaluación del modelo para Normas_subjetivas:
Mean Squared Error (MSE): 0.4877
                            OLS Regression Results
==============================================================================
Dep. Variable:      Normas_subjetivas   R-squared:                       0.399
Model:                            OLS   Adj. R-squared:                  0.361
Method:                 Least Squares   F-statistic:                     10.40
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           8.06e-09
Time:                        09:58:25   Log-Likelihood:                -131.04
No. Observations:                 101   AIC:                             276.1
Df Residuals:                      94   BIC:                             294.4
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.7168      0.335      5.124      0.000       1.051       2.382
Bandwagon_Consumption     0.2656      0.323      0.824      0.412      -0.375       0.906
Snob_Consumption         -0.0159      0.250     -0.064      0.949      -0.513       0.481
BC_CM                    -0.0238      0.068     -0.352      0.726      -0.158       0.110
BC_So                     0.0491      0.027      1.833      0.070      -0.004       0.102
SC_CM                    -0.0158      0.052     -0.302      0.763      -0.120       0.088
SC_Si                     0.0330      0.019      1.730      0.087      -0.005       0.071
==============================================================================
Omnibus:                       11.296   Durbin-Watson:                   2.301
Prob(Omnibus):                  0.004   Jarque-Bera (JB):               11.738
Skew:                           0.723   Prob(JB):                      0.00283
Kurtosis:                       3.834   Cond. No.                         168.
==============================================================================
```

El modelo presenta un error bajo, inferior a 0,5 puntos, lo que indica una buena capacidad predictiva. El R^2 muestra que explica aproximadamente el 40 % de la variabilidad en normas subjetivas, sugiriendo que otros factores no incluidos pueden influir. El estadístico F confirma que al menos una variable independiente tiene un efecto significativo.

Las interacciones entre singularidad y consumo esnob, y entre sostenibilidad y consumo de arrastre, son las únicas que parecen influir significativamente. Sin embargo, la influencia del conocimiento de marca a través del consumo de arrastre es negativa, por lo que se rechaza la hipótesis H12.1. Aunque el conocimiento de marca también afecta negativamente a través del consumo esnob, su P-valor no es lo suficientemente bajo, lo que lleva a rechazar H12.2. También se rechaza H14, ya que la singularidad no tiene un efecto significativo mediante el consumo esnob. Las hipótesis H1.1 y H1.2 también se rechazan, ya que el consumo de arrastre y el consumo esnob parecen influir significativamente en las normas subjetivas. Solo se mantiene H13, donde la sostenibilidad tiene un impacto significativo y positivo en normas subjetivas a través del consumo de arrastre.

Se intentó mejorar el modelo eliminando variables poco significativas, reduciendo el MSE a 0,4258 puntos y ajustando el R^2 al 38 %.

Finalmente, añadir nuevas interacciones basadas en correlaciones no mejoró los resultados en ninguno de los modelos, por lo que se procedió a construir el modelo final.

Para construir el modelo final, emplearemos únicamente las variables surgidas de la TPB, variables dependientes de los modelos anteriores: Actitudes, PBC y Normas Subjetivas, con el objetivo de verificar las hipótesis que nos faltan. Posteriormente, incluiremos las variables relativas a las hipótesis no rechazadas en los modelos anteriores, para obtener un modelo final que englobe todo.

```
Evaluación del modelo para Intencion_Compra:
Mean Squared Error (MSE): 2.1999
                            OLS Regression Results
==============================================================================
Dep. Variable:       Intencion_Compra   R-squared:                       0.441
Model:                           OLS   Adj. R-squared:                  0.424
Method:                Least Squares   F-statistic:                     25.51
Date:               Sun, 16 Feb 2025   Prob (F-statistic):           2.99e-12
Time:                       10:37:30   Log-Likelihood:                -188.01
No. Observations:                101   AIC:                             384.0
Df Residuals:                     97   BIC:                             394.5
Df Model:                          3
Covariance Type:            nonrobust
===========================================================================================
                              coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------
Intercept                   0.3400      0.570      0.597      0.552      -0.791       1.471
Actitudes                   0.5188      0.136      3.814      0.000       0.249       0.789
Normas_subjetivas          -0.2749      0.156     -1.762      0.081      -0.585       0.035
PBC                         0.5593      0.159      3.517      0.001       0.244       0.875
==============================================================================
Omnibus:                        2.472   Durbin-Watson:                   2.044
Prob(Omnibus):                  0.290   Jarque-Bera (JB):                1.723
Skew:                           0.100   Prob(JB):                        0.423
Kurtosis:                       2.392   Cond. No.                         26.0
==============================================================================
```

Como podemos ver, el modelo tiene un error algo elevado, una explicabilidad de R^2 moderada y una significación global muy elevada.

Además, la única variable que no es altamente significativa es Normas Subjetivas (cuya influencia además parece ser negativa) por lo que únicamente rechazaremos la hipótesis H0.

Para mejorar el modelo, incluimos las variables significativas de los modelos anteriores a modo de interacción con las variables que eran las dependientes del modelo del que proceden y obtuvimos un modelo con menor error y mejor capacidad predictiva.

```
Evaluación del modelo para Intencion_Compra:
Mean Squared Error (MSE): 1.9580
                            OLS Regression Results
==============================================================================
Dep. Variable:       Intencion_Compra   R-squared:                       0.476
Model:                           OLS   Adj. R-squared:                  0.425
Method:                Least Squares   F-statistic:                     9.197
Date:               Sun, 16 Feb 2025   Prob (F-statistic):           8.65e-10
Time:                       10:37:46   Log-Likelihood:                -184.72
No. Observations:                101   AIC:                             389.4
Df Residuals:                     91   BIC:                             415.6
Df Model:                          9
Covariance Type:            nonrobust
===========================================================================================
                              coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------
Intercept                  -0.0525      0.761     -0.069      0.945      -1.563       1.458
Actitudes                   0.8233      0.306      2.693      0.008       0.216       1.431
Normas_subjetivas          -0.3029      0.188     -1.612      0.111      -0.676       0.070
PBC                         0.4111      0.235      1.746      0.084      -0.057       0.879
PBC_Ri                      0.0136      0.035      0.393      0.695      -0.055       0.082
Si_A                       -0.0912      0.041     -2.199      0.030      -0.174      -0.009
CE_A                        0.0096      0.027      0.353      0.725      -0.044       0.064
CP_A                        0.0098      0.040      0.246      0.806      -0.069       0.089
BC_So                      -0.0158      0.028     -0.553      0.581      -0.072       0.041
SC_Si                       0.0639      0.031      2.040      0.044       0.002       0.126
==============================================================================
Omnibus:                        2.867   Durbin-Watson:                   1.982
Prob(Omnibus):                  0.238   Jarque-Bera (JB):                1.905
Skew:                           0.113   Prob(JB):                        0.386
Kurtosis:                       2.367   Cond. No.                         261.
==============================================================================
```

Finalmente, podemos establecer que influyen de forma significativa en la intención de compra: las actitudes hacia el producto (muy positivamente), la Singularidad a través de

las Actitudes (levemente negativa) y la Singularidad a través del consumo esnob (positivamente). También influye, aunque de forma más, leve el PBC (positivamente).

5. Conclusiones

A. Estudio exploratorio

El estudio exploratorio proporciona una primera aproximación a las percepciones, asociaciones emocionales y motivaciones de los consumidores en relación con el lujo y, en particular, con la moda de lujo de segunda mano. Mediante el análisis de respuestas abiertas y técnicas de *text-mining* como *topic modeling* y coocurrencias, se identificaron patrones clave en la percepción del lujo entre distintos segmentos de consumidores.

Uno de los hallazgos más relevantes es la diferencia en la percepción del lujo entre quienes han consumido artículos de lujo y quienes no. Los no consumidores asocian el lujo con atributos tangibles como precio alto, exclusividad y estatus social, muchas veces con una connotación negativa. También lo relacionan con opulencia, superficialidad y gasto innecesario. En cambio, los consumidores de lujo lo perciben de manera más subjetiva y experiencial, vinculándolo con calidad, disfrute, autoexpresión y legado. Este grupo también valora la historia y manufactura de los productos, integrándolos en su estilo de vida.

Otro hallazgo importante es la relación del lujo con la calidad, la exclusividad y el precio. Ambos grupos coinciden en que el lujo se sustenta en estos tres pilares, aunque con diferencias. Mientras los consumidores priorizan la calidad y exclusividad sobre el precio, los no consumidores lo asocian principalmente con su alto costo y acceso restringido a ciertos grupos socioeconómicos.

El análisis de sentimientos reveló una polarización emocional. Entre los no consumidores, las emociones más mencionadas fueron aspiración y deseo, pero también rechazo e indiferencia, vinculando el lujo con superficialidad y materialismo. En los consumidores predominan sentimientos positivos como satisfacción, placer, confianza y autoafirmación, aunque también aparece la culpa, lo que sugiere dilemas éticos en el consumo de lujo.

Además, se identificaron factores del modelo teórico que están naturalmente asociados al lujo y otros que no parecen ser esenciales. Calidad alta, exclusividad, escasez, precio

elevado y singularidad fueron ampliamente mencionados, indicando que forman parte del imaginario colectivo del lujo. Sin embargo, la sostenibilidad, la creatividad y la emoción tienen menor asociación. Aunque algunos consumidores valoran estos aspectos, no se perciben como atributos inherentes al lujo. En particular, la sostenibilidad es poco mencionada, lo que sugiere que, aunque el lujo de segunda mano puede beneficiarse de la tendencia hacia un consumo más responsable, este no es aún un factor determinante.

Las razones principales para consumir lujo incluyen calidad y durabilidad, disfrute personal, autoafirmación, prestigio y la posibilidad de transmitir las prendas como legado. En contraste, los no consumidores mencionaron la falta de poder adquisitivo, la percepción del lujo como algo superficial o innecesario y el rechazo hacia marcas ostentosas.

B. Estudio confirmatorio

Los resultados de la regresión múltiple evidencian que las actitudes hacia los productos de lujo de segunda mano tienen una influencia positiva y significativa en la intención de compra. Es decir, cuanto más favorable sea la percepción de estos productos, mayor será la predisposición a adquirirlos.

El PBC también influye positivamente en la intención de compra, sugiriendo que una mayor facilidad de acceso y confianza en la compra favorecen su adopción. En contraste, las normas subjetivas no resultaron significativas, lo que indica que la presión social no es un factor clave en este consumo. Sin embargo, la sostenibilidad, a través del consumo de arrastre, influye en la percepción social del lujo de segunda mano, lo que podría evolucionar con las tendencias culturales.

Se confirma que singularidad y creatividad o emoción juegan un papel relevante en la formación de actitudes favorables, ya que los consumidores ven este mercado como una oportunidad para adquirir piezas únicas y con valor simbólico. Asimismo, el consumo esnob potencia la intención de compra cuando está vinculado con la singularidad, ya que quienes buscan exclusividad encuentran en la segunda mano acceso a piezas difíciles de conseguir en *retail*.

Por otro lado, conocimiento de marca y riesgo percibido no tienen un impacto significativo en las actitudes hacia la compra de lujo de segunda mano. Esto podría explicarse porque los consumidores más informados sobre marcas de lujo son más exigentes con la autenticidad y estado de los productos.

El riesgo percibido tampoco afecta directamente la intención de compra, pero sí influye en el PBC, reduciendo la percepción de control y afectando indirectamente la decisión de compra. En cuanto al valor de reventa, no se ha encontrado una relación significativa con las actitudes hacia el producto, lo que sugiere que los consumidores no ven el lujo de segunda mano como una inversión, sino como una alternativa más accesible al lujo tradicional.

A continuación, se presenta un resumen del modelo final con las variables cuya relevancia ha sido demostrada en el estudio.

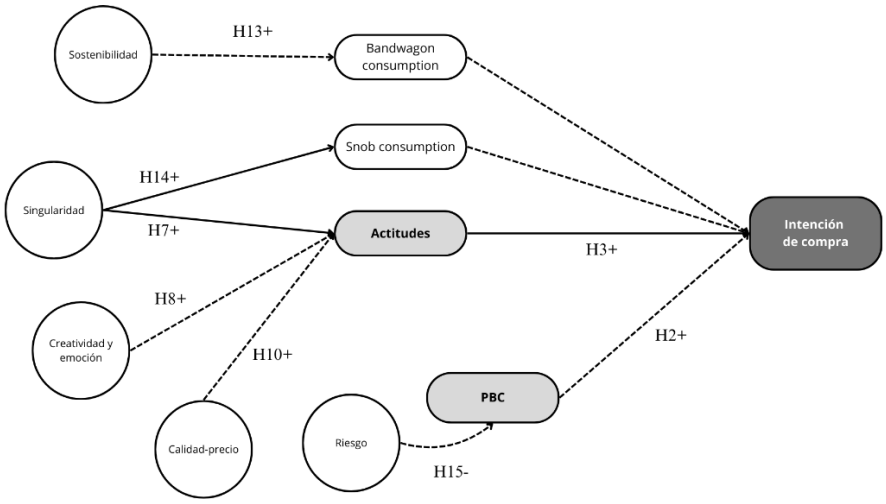


Ilustración 3: Esquema del modelo final, elaboración propia (líneas continuas: influencias significativas; líneas discontinuas: influencias casi significativas e interacción entre variables; todo ello a un nivel de significación del 7%).

C. Limitaciones

Una de las principales limitaciones de este estudio es que la encuesta fue realizada en castellano, restringiendo la muestra a participantes mayoritariamente de España o Latinoamérica. Esto puede haber influido en los resultados, ya que las percepciones sobre la compra de lujo de segunda mano varían entre países. En particular, en España, el mercado de segunda mano no está tan normalizado como en otras regiones europeas, lo

que podría haber generado una actitud más negativa hacia este tipo de consumo en comparación con mercados donde es más aceptado.

Otra limitación importante es el tamaño de la muestra en el estudio confirmatorio. Aunque es válida para análisis exploratorios y modelos de regresión, una muestra más amplia habría permitido resultados más extrapolables y una mayor robustez estadística, reduciendo posibles sesgos. Además, la encuesta se difundió principalmente en redes sociales, lo que pudo atraer a un perfil de encuestados más jóvenes y familiarizados con la tecnología, dejando fuera a otros segmentos con patrones de consumo distintos. La extensión de la encuesta también pudo disuadir a algunos participantes, afectando la tasa de respuesta y la calidad de los datos.

Las respuestas pueden haber estado influenciadas por percepciones subjetivas del lujo según el nivel de exposición y experiencia del encuestado. Quienes nunca han consumido lujo podrían haber respondido en función de ideas preconcebidas, en lugar de experiencias reales. Asimismo, algunos encuestados pueden haber respondido de manera socialmente aceptable en temas como sostenibilidad o estatus, cayendo en el sesgo de deseabilidad social.

Por último, dado que este estudio es correlacional y no experimental, no se pueden establecer relaciones de causalidad directa entre variables. Si bien estas limitaciones no invalidan los hallazgos del estudio, sí sugieren la necesidad de interpretarlos con prudencia y en su debido contexto. Futuras investigaciones deberían ampliar la muestra, incluir participantes de diversas culturas, reducir la extensión de la encuesta y utilizar metodologías experimentales o longitudinales para analizar la evolución del comportamiento del consumidor en el mercado del lujo de segunda mano.

D. Recomendaciones

En base a los hallazgos del estudio, se proponen las siguientes recomendaciones para marcas de lujo interesadas en el mercado de segunda mano, plataformas de reventa y estrategias de marketing enfocadas en consumidores de lujo de segunda mano.

En primer lugar, es clave potenciar la percepción de singularidad y exclusividad. Los consumidores valoran la unicidad en estos productos, por lo que se recomienda reforzar el *storytelling* sobre la historia, procedencia y exclusividad de cada artículo. Estrategias

como la certificación de productos vintage, la selección de ediciones limitadas y la personalización pueden aumentar la percepción de valor.

Otra recomendación es reducir la percepción de riesgo y mejorar el PBC. Para ello, las plataformas deben continuar implementando estrategias que garanticen autenticidad y calidad, como sistemas de verificación, certificación y políticas de devolución claras, brindando mayor confianza a los consumidores.

Además, se recomienda fomentar la sostenibilidad como atributo diferenciador. La sostenibilidad tiene un impacto positivo en la percepción social del lujo de segunda mano. Por ello, las marcas y plataformas deben comunicar más claramente los beneficios ambientales de este mercado, destacando su contribución a la economía circular y la reducción de residuos en la moda.

Otra estrategia clave es generar campañas de marketing basadas en emociones y creatividad. La creatividad y la emoción influyen en las actitudes positivas hacia el lujo de segunda mano, por lo que las marcas deben explorar nuevas formas de interacción con los consumidores. Estrategias como experiencias inmersivas sobre la historia de los productos y campañas que resalten el valor emocional de las prendas pueden reforzar la conexión con el consumidor, apelando a la nostalgia y la identidad personal.

Finalmente, es crucial reforzar la segmentación de consumidores y diferenciar estrategias. Los resultados indican que los consumidores de lujo de segunda mano tienen motivaciones distintas. Se recomienda adaptar la comunicación y el marketing según el perfil del consumidor, diferenciando entre consumidores esnob, de arrastre o interesados en sostenibilidad, y destacando los atributos más relevantes para cada segmento.

Estas estrategias permitirán a las marcas y plataformas de reventa mejorar la percepción del lujo de segunda mano, aumentar la confianza del consumidor y fortalecer su posicionamiento en el mercado.

6. Declaración uso herramientas de inteligencia artificial generativa

ADVERTENCIA: Desde la Universidad consideramos que ChatGPT u otras herramientas similares son herramientas muy útiles en la vida académica, aunque su uso queda siempre bajo la responsabilidad del alumno, puesto que las respuestas que proporciona pueden no ser veraces. En este sentido, NO está permitido su uso en la elaboración del Trabajo fin de Grado para generar código porque estas herramientas no son fiables en esa tarea. Aunque el código funcione, no hay garantías de que metodológicamente sea correcto, y es altamente probable que no lo sea.

Por la presente, yo, Rocío Medina Llamas, estudiante de E2+Business Analytics de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado "De Prada a Pre-Amada: democratización del lujo a través de la segunda mano", declaro que he utilizado la herramienta de Inteligencia Artificial Generativa ChatGPT u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación:

1. **Metodólogo:** Para descubrir métodos aplicables a problemas específicos de investigación.
2. **Interpretador de código:** Para realizar análisis de datos preliminares.
3. **Generador de problemas de ejemplo:** Para ilustrar conceptos y técnicas.
4. **Revisor:** Para recibir sugerencias sobre cómo mejorar y perfeccionar el trabajo con diferentes niveles de exigencia.

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG y he explicitado para que se ha usado ChatGPT u otras herramientas similares). Soy consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

Fecha: 18/02/2025

Firma: 

7. Bibliografía

1) Abtan, O., Ducasse, P., Finet, L., Gardet, C., Gasc, M., & Salaire, S. (2019). Why luxury brands should celebrate the preowned boom. *BCG https://www. bcg. com/publications/2019/luxury-brands-should-celebrate-preowned-boom. aspx*.

2) Ajzen, I. (1991). The Theory of planned behavior. *Organizational Behavior and Human Decision Processes*.

3) Berthon, P., Pitt, L., Parent, M., & Berthon, J. P. (2009). Aesthetics and ephemerality: Observing and preserving the luxury brand. California management review, 52(1), p.48.

4) Canals, C. (2019, 16 septiembre). La emergencia de la clase media: cosa de emergentes. CaixaBank Research. https://www.caixabankresearch.com/es/economia-y-mercados/mercado-laboral-y-demografia/emergencia-clase-media-cosa-emergentes?

5) Carrigan, M., Moraes, C., & McEachern, M. (2013). From conspicuous to considered fashion: A harm-chain approach to the responsibilities of luxury-fashion businesses. *Journal Of Marketing Management*, *29*(11-12), 1277-1307. https://doi.org/10.1080/0267257x.2013.798675

6) Cervellon, M.-C., Carey, L., & Harms, T. (2012). Something old, something used: Determinants of women's purchase of vintage fashion vs second-hand fashion. International Journal of Retail & Distribution Management, 40(12), 956-974. https://doi.org/10.1108/09590551211274946

7) Christodoulides, G., Athwal, N., Boukis, A., & Semaan, R. W. (2021). New forms of luxury consumption in the sharing economy. Journal of Business Research, 137, 89–99.

8) Cinco Días. (2022, 21 de septiembre). La generación Z representará el 70% de las compras de las marcas de lujo para 2025. El País. Recuperado de https://cincodias.elpais.com/cincodias/2022/09/21/fortunas/1663763569_841273.html

9) D'Arpizio, C., Levato, F., Gault, C., De Montgolfier, J., & Jaroudi, L. (2021). From surging recovery to elegant advance: the evolving future of luxury. *Bain & Company*.

10) Eastman, J. K., Goldsmith, R. E., & Flynn, L. R. (1999). Status consumption in consumer behavior: Scale development and validation. Journal of marketing theory and practice, 7(3), p.42.

11) Gerval, O. (2008). Fashion: Concept to catwalk. *(No Title)*.

12) Jiang, Y., Xiao, L., Jalees, T., Naqvi, M. H., & Zaman, S. I. (2018). Moral and ethical antecedents of attitude toward counterfeit luxury products: Evidence from Pakistan. Emerging Markets Finance and Trade, 54(15), 3519-3538.

13) Jones, M. A., Reynolds, K. E., & Arnold, M. J. (2006). Hedonic and utilitarian shopping value: Investigating differential effects on retail outcomes. Journal of business research, 59(9), p.974.

14) Joung, H. M., & Park-Poaps, H. (2013). Factors motivating and influencing clothing disposal behaviours. *International Journal of consumer studies*, *37*(1).

15) Kessous, A., & Valette-Florence, P. (2019). "From Prada to Nada": Consumers and their luxury products: A contrast between second-hand and first-hand luxury products. *Journal of Business Research, 102*.

16) Leibenstein, H. (1950). Bandwagon, snob, and Veblen effects in the theory of consumers' demand. The quarterly journal of economics, 64(2), p.183-207.

17) Liu, C., Xia, S., & Lang, C. (2023). Online Luxury Resale Platforms and Customer Experiences: A Text Mining Analysis of Online Reviews. *Sustainability*, *15*(10), 8137. https://doi.org/10.3390/su15108137

18) Loeb, W. (2022, 21 de febrero). *Luxury brand prices rise sharply — will it cut demand?* Forbes. https://www.forbes.com/sites/walterloeb/2022/02/21/luxury-brand-prices-rise-sharply--will-it-cut-demand/

19) Sarial-Abi, G., Vohs, K. D., Hamilton, R., & Ulqinaku, A. (2016). Stitching time: Vintage consumption connects the past, present, and future. Journal of Consumer Psychology, 26(4), 451–460. https://doi.org/10.1016/j.jcps.2016.06.005

20) Shim, S. (1996). Adolescent consumer decision-making styles: The consumer socialization perspective. Psychology & Marketing, 13(6), 547-569.

21) Shukla, P., Rosendo-Rios, V., Trott, S., Lyu, J., & Khalifa, D. (2022). Managing the challenge of luxury democratization: A multicountry analysis. *Journal of International Marketing, 30(4).*

22) Sihvonen, J., & Turunen, L. L. M. (2016). As good as new–valuing fashion brands in the online second-hand markets. *Journal of Product & Brand Management*, *25*(3).

23) Statista. (2023). *Valor del mercado mundial de artículos de lujo personales usados de 2022 a 2027*. Recuperado el 23 de diciembre de 2024, de https://es.statista.com/estadisticas/1274894/valor-del-mercado-mundial-de-articulos-de-lujo-personales-usados/

24) Stolz, K. (2022). Why do (n't) we buy second-hand luxury products? *Sustainability*, *14*(14).

25) Turunen, L. L. M., & Leipämaa-Leskinen, H. (2015). Pre-loved luxury: Identifying the meanings of second-hand luxury possessions. *Journal of Product & Brand Management*, *24*(1).

26) Turunen, L. L. M., & Pöyry, E. (2019). Shopping with the resale value in mind: A study on second-hand luxury consumers. *International Journal of Consumer Studies*, *43*(6).

8. Anexos

A. Cuestiones de la encuesta del estudio exploratorio

1. Indique su género

	Hombre
	Mujer
	Prefiero no decirlo

2. Indique su rango de edad

	< 18
	18 – 25
	26 – 30
	31 – 40
	41 – 50
	51 – 60
	61 – 70
	>70

3. Indique su rango de ingresos anuales

	< 10.000€
	10.000-30.000 €

	30.000-50.000€
	50.000-100.000€
	100.000-200.000€
	200.000-500.000€
	>500.000€

4. Indique su ocupación

	Estudiante
	Desempleado
	Jubilado
	Empleado/a por cuenta ajena (sector privado)
	Empleado/a por cuenta ajena (sector público)
	Autónomo/a o profesional independiente
	Empresario/a o propietario de un negocio
	Director/a o Socio/a de un negocio
	Amo/a de casa
	Otro

5. ¿Ha comprado o consumido alguna vez un artículo de moda de lujo? Ya sean prendas, accesorios, joyería, etc.

	Sí, alguna vez
	No, nunca

6. ¿Qué marcas de lujo conoces? Menciona al menos 3.
7. ¿Qué significa para ti la palabra lujo?
8. Ahora trata de definir lujo en una única palabra
9. ¿Con qué asocias tú el lujo?
10. ¿Qué características debe tener, en tu opinión, un artículo de lujo?
11. ¿Cuáles de estos factores consideras que están relacionados con el lujo?

a. Precio alto

b. Calidad alta

c. Exclusividad o escasez

d. Sostenibilidad

e. Rareza o singularidad

f. Creatividad y emoción

g. Expresión artística

h. Superficialidad

i. Otros

12. ¿Qué sentimientos te genera el concepto de lujo?

13. ¿Qué sentimientos te genera consumir artículos de lujo?

14. ¿Cuál/es son las razones para que consumas artículos de lujo?

15. Cuestiones de la encuesta del estudio exploratorio

1. ¿Has comprado o consumido alguna vez un artículo de moda de lujo de segunda mano? Ya sean prendas, accesorios, joyería, etc.

2. Indique su grado de acuerdo o desacuerdo con la siguiente afirmación, siendo 1 totalmente en desacuerdo y 7 totalmente de acuerdo.

1. Intención de compra

Me interesa comprar artículos de lujo de segunda mano	1	2	3	4	5	6	7

2. Bandwagon consumption

Suelo estar atento a las tendencias de moda	1	2	3	4	5	6	7
Me atrae la idea de adquirir productos de lujo populares entre mis amigos o conocidos	1	2	3	4	5	6	7
Suelo dejarme llevar por las tendencias y comprar lo que está de moda	1	2	3	4	5	6	7

Creo que comprar artículos de lujo que están en tendencia me hace sentir más integrado/a socialmente	1	2	3	4	5	6	7
Considero que poseer prendas de lujo mejora mi estatus social	1	2	3	4	5	6	7
Me interesa consumir productos de lujo con logos o en los que la marca sea fácilmente reconocible	1	2	3	4	5	6	7

3. Snob consumption

Prefiero comprar prendas que pocas personas tienen	1	2	3	4	5	6	7
Me atrae la idea de destacar/diferenciarme de mi entorno por mi forma de vestir y mi estilo personal	1	2	3	4	5	6	7
Suelo dejarme llevar por las tendencias y comprar lo que está de moda	1	2	3	4	5	6	7
La exclusividad de un producto de lujo es un factor clave en mi decisión de compra.	1	2	3	4	5	6	7
En caso de consumir artículos de lujo, me interesa consumir ediciones especiales por encima de productos permanentes.	1	2	3	4	5	6	7
En caso de comprar artículos de lujo, me interesa más adquirir piezas que representen rareza o escasez.	1	2	3	4	5	6	7
Me interesa consumir productos de lujo sin logos o en los que la marca no es fácilmente reconocible	1	2	3	4	5	6	7

4. Attitudes

Tengo una percepción positiva hacia las prendas de lujo de segunda mano	1	2	3	4	5	6	7
Creo que comprar prendas de lujo de segunda mano es una elección inteligente en términos de calidad-precio.	1	2	3	4	5	6	7
Creo que comprar prendas de lujo de segunda mano es una elección sostenible.	1	2	3	4	5	6	7
Suelo mirar plataformas de segunda mano cuando quiero adquirir un producto de lujo	1	2	3	4	5	6	7
Considero que en las plataformas de segunda mano hay prendas de lujo más originales o especiales que en las alternativas de primera mano	1	2	3	4	5	6	7

5. Subjective norms

Mis decisiones de compra se suelen ver afectadas por las opiniones de mis familiares y amigos	1	2	3	4	5	6	7

Considero que comprar lujo de segunda mano es algo aceptado positivamente en mi entorno	1	2	3	4	5	6	7
Siento presión social para comprar productos de lujo de segunda mano en lugar de productos nuevos	1	2	3	4	5	6	7
La recomendación de influencers o figuras públicas me anima a considerar el lujo de segunda mano.	1	2	3	4	5	6	7

6. *PBC*

Considero que es fácil acceder a prendas de lujo de segunda mano en el mercado actual	1	2	3	4	5	6	7
Me siento capaz de identificar prendas de lujo de segunda mano auténticas y de calidad	1	2	3	4	5	6	7
Creo que tengo suficiente información para tomar decisiones de compra en el mercado de lujo de segunda mano	1	2	3	4	5	6	7
Considero que la falta de información sobre el estado o la autenticidad de productos de lujo de segunda mano me supone un riesgo para la compra	1	2	3	4	5	6	7
Considero que la información y la facilidad de acceso a comprar lujo de segunda mano incrementa mi intención de compra	1	2	3	4	5	6	7

7. Sostenibilidad

Considero que comprar prendas de lujo es más sostenible que consumir prendas fast-fashion	1	2	3	4	5	6	7
Creo que optar por segunda mano es una forma de apoyar la sostenibilidad en la moda	1	2	3	4	5	6	7
Me preocupa el impacto medioambiental de mis decisiones de compra de moda	1	2	3	4	5	6	7
Valoro más un producto de lujo si está vinculado a prácticas sostenibles	1	2	3	4	5	6	7
Considero que la durabilidad de un producto de lujo lo convierte en una elección sostenible	1	2	3	4	5	6	7
Siento cierta presión social para tomar decisiones de compra de moda sostenibles	1	2	3	4	5	6	7

8. Conocimiento de marca

Reconozco fácilmente las marcas de lujo en el mercado de segunda mano	1	2	3	4	5	6	7

Tengo un conocimiento profundo sobre las marcas de lujo y sus productos	1	2	3	4	5	6	7
La reputación de una marca de lujo influye en mi decisión de comprar sus productos de segunda mano	1	2	3	4	5	6	7
Me siento más seguro/a comprando prendas de lujo de segunda mano cuando provienen de una marca reconocida o que conozco bien	1	2	3	4	5	6	7
Me atrae más comprar productos de lujo de segunda mano de marcas reconocidas o que conozco bien	1	2	3	4	5	6	7

9. Singularidad

Comprar prendas de segunda mano me permite expresar mi individualidad	1	2	3	4	5	6	7
Valoro las prendas de lujo de segunda mano porque me permiten diferenciarme de los demás	1	2	3	4	5	6	7
Me atrae la idea de poseer productos de lujo únicos que no están disponibles en el mercado tradicional	1	2	3	4	5	6	7
Las ediciones limitadas de segunda mano productos de lujo me atraen más que el producto original	1	2	3	4	5	6	7

10. Calidad-precio

Las prendas de lujo de segunda mano ofrecen una buena relación calidad-precio	1	2	3	4	5	6	7
El precio es un factor muy relevante para mí a la hora de consumir productos de lujo	1	2	3	4	5	6	7
Me atrae la idea comprar productos de lujo a un precio inferior que en el mercado tradicional	1	2	3	4	5	6	7
La calidad de los productos de lujo de segunda mano justifica su precio	1	2	3	4	5	6	7
Considero que el lujo de segunda mano es una opción más rentable que el lujo nuevo	1	2	3	4	5	6	7

11. Nostalgia

Me interesa adquirir prendas características una época pasada	1	2	3	4	5	6	7
Suelo sentir nostalgia por objetos que ya no se comercializan	1	2	3	4	5	6	7
Me atraen los productos de lujo de segunda mano por su historia y tradición	1	2	3	4	5	6	7

Valoro las prendas de lujo de segunda mano porque evocan emociones y recuerdos especiales	1	2	3	4	5	6	7

12. *Riesgo*

Considero que comprar lujo de segunda mano conlleva un nivel de riesgo elevado	1	2	3	4	5	6	7
Me disuade de comprar productos de lujo de segunda mano el hecho de que puedan ser falsos	1	2	3	4	5	6	7
Percibo un riesgo alto asociado al estado de las prendas de lujo de segunda mano	1	2	3	4	5	6	7
Creo que existe poca transparencia en las transacciones de lujo de segunda mano	1	2	3	4	5	6	7
Suelo tomar riesgos con mi dinero	1	2	3	4	5	6	7
Evitaría comprar lujo de segunda mano debido a la falta de garantías de autenticidad y/o buen estado	1	2	3	4	5	6	7

13. Valor de reventa

Me intereso por el valor de reventa de productos de lujo	1	2	3	4	5	6	7
Me interesa comprar productos de lujo de segunda mano por la posibilidad de revenderlos en el futuro por un precio superior al que lo compré	1	2	3	4	5	6	7
Considero que el valor de reventa de los productos de lujo se mantiene o incluso crece	1	2	3	4	5	6	7
Considero que comprar productos de lujo de segunda mano es una buena inversión por su valor de reventa	1	2	3	4	5	6	7

14. Creatividad y emoción

Comprar prendas de lujo de segunda mano es una experiencia emocionante para mí	1	2	3	4	5	6	7
Me siento creativo/a al buscar y elegir productos de segunda mano.	1	2	3	4	5	6	7
Experimento emociones positivas al encontrar un producto que me gusta después de una búsqueda en plataformas de segunda mano	1	2	3	4	5	6	7
Considero estimulante el proceso de encontrar piezas únicas de lujo de segunda	1	2	3	4	5	6	7

3. Indique su género

☐ Hombre

	Mujer
	Prefiero no decirlo

4. Indique su rango de edad

	< 18
	18 – 25
	26 – 30
	31 – 40
	41 – 50
	51 – 60
	61 – 70
	>70

5. Indique su rango de ingresos anuales

	< 10.000€
	10.000-30.000 €
	30.000-50.000€
	50.000-100.000€
	100.000-200.000€
	200.000-500.000€
	>500.000€

6. Indique su ocupación

	Estudiante
	Desempleado
	Jubilado
	Empleado/a por cuenta ajena (sector privado)
	Empleado/a por cuenta ajena (sector público)
	Autónomo/a o profesional independiente
	Empresario/a o propietario de un negocio

	Director/a o Socio/a de un negocio
	Amo/a de casa
	Otro

16. Código del estudio exploratorio (LDA)

ANÁLISIS DEL ESTUDIO EXPLORATORIO

Para este análisis, emplearemos topic modelling mediante un modelo LDA de aprendizaje automático para extraer los temas principales de cada una de las preguntas del estudio exploratorio y las palabras clave de cada uno de los temas.

Primero, dividiremos el dataset en dos grupos, en base a la respuesta a la oregunta filtro sobre si habían consumido lujo con anterioridad.Una vez tengamos ambos grupos separados, trataremos cada pregunta del modelo por separado y luego le daremos como entrada al modelo los documentos uno a uno para que extraiga los temas de cada una de las preguntas (en cada grupo de consumidores).

```
[ ] import pandas as pd

    # Cargar el archivo CSV con separador de comas y manejo de comillas dobles
    df = pd.read_csv("Estudio Exploratorio TFG BA.csv", sep=",", quotechar='"')

    # Verificar las primeras filas del DataFrame
    print(df.head())
```

```
                  Marca temporal Indique su género Indique su rango de edad  \
0   2025/01/29 9:46:06 p. m. CET            Mujer                 18 - 25
1   2025/01/29 9:53:04 p. m. CET            Mujer                 18 - 25
2   2025/01/30 6:58:54 p. m. CET            Mujer                 51 - 60
3  2025/02/02 10:01:49 a. m. CET           Hombre                 51 - 60
4   2025/02/02 7:23:47 p. m. CET           Hombre                 18 - 25

  Indique su rango de ingresos anuales  \
0                        < 10.000 €
1                        < 10.000 €
2                100.000 - 200.000 €
3                       > 500.000 €
4                        < 10.000 €

                    Indique su ocupación  \
0                             Estudiante
1                             Estudiante
2  Autónomo/a o profesional independiente
3     Director/a o Socio/a de una empresa
4                             Estudiante

  ¿Ha comprado o consumido alguna vez un artículo de moda de lujo? Ya sean prendas, accesorios, joyería, etc.  \
0                                  Sí, alguna vez
1                                  Sí, alguna vez
2                                  Sí, alguna vez
3                                  Sí, alguna vez
4                                  Sí, alguna vez

  ¿Qué marcas de lujo conoces? Menciona al menos 3.  \
0                                               NaN
```

```
3                          Con trabajo bien hecho
4                       dinero, elegancia, disfrute.

  ¿Qué características debe tener, en tu opinión, un articulo de lujo?.1  \
0          Poco accesible, precio alto, calidad alta
1  Gran calidad, precio elevado, historia, artesa...
2            Alta calidad , exclusividad, elegancia.
3  Materiales nobles, buen diseño y una utilidad/...
4             alta calidad, alto valor y exclusividad

  ¿Cuáles de estos factores consideras que están relacionados con el lujo?.1  \
0  Precio alto;Calidad alta;Exclusividad o escase...
1  Precio alto;Calidad alta;Exclusividad o escase...
2    Precio alto;Calidad alta;Exclusividad o escasez
3  Precio alto;Calidad alta;Exclusividad o escase...
4  Calidad alta;Exclusividad o escasez;Rareza o s...

  ¿Qué sentimientos te genera el concepto de lujo?.1  \
0                              Admiración, ambición
1                              Ambición, relajación
2                                           Placer
3                                     Admiración
4                                        asombro

  ¿Qué sentimientos te genera consumir artículos de lujo?  \
0  Alegría, satisfacción personal por conseguir a...
1                Satisfacción, relajación, felicidad
2                                           Placer
3                                         Alegría
4                                         emocion

  ¿Cuál/es son las razones para que consumas artículos de lujo?
0  más calidad y durabilidad, poderlo pasar de ge...
1  Para tener artículos de calidad que cuenten un...
2       Me gusta disfrutar de la calidad excepcional
3        Autoafirmación y recompensarse a sí mismo
4  buscar disfrutar de una experiencia única o de...

[5 rows x 23 columns]
```

```
[ ] # Nombre de la columna de la pregunta filtro
    columna_filtro = "¿Ha comprado o consumido alguna vez un artículo de moda de lujo? Ya sean prendas, accesorios, joyería, etc."

    # Filtrar los datos en dos grupos
    grupo_si = df[df[columna_filtro] == "Sí, alguna vez"]
    grupo_no = df[df[columna_filtro] == "No, nunca"]

    # Verificar tamaños de los grupos
    print(f"Nº de respuestas de consumidores de lujo: {len(grupo_si)}")
    print(f"Nº de respuestas de no consumidores de lujo: {len(grupo_no)}")
```

```
Nº de respuestas de consumidores de lujo: 18
Nº de respuestas de no consumidores de lujo: 8
```

```
[ ] # Visualización preliminar del grupo de consumidores de lujo
    grupo_si.head()
```

	Marca temporal	Indique su género	Indique su rango de edad	Indique su rango de ingresos anuales	Indique su ocupación	¿Ha comprado o consumido alguna vez un artículo de moda de lujo? Ya sean prendas, accesorios, joyería, etc.	¿Qué marcas de lujo conoces? Menciona al menos 3.	¿Qué significa para ti la palabra lujo?	Ahora trata de definir lujo en una única palabra	¿Con qué asocias tú el lujo?	...	¿Cuál/es son las razones para que no consumas artículos de lujo?	¿Qué marcas de lujo conoces? Menciona al menos 3..1	¿Qué significa para ti la palabra lujo?.1	Ahora trata de definir lujo en una única palabra.1	¿Con qué asocias tú el lujo?.1	¿Qué características debe tener, en tu opinión, un articulo de lujo?.1	¿Cuáles de estos factores consideras que están relacionados con el lujo?.1	¿Qué sentimientos te genera el concepto de lujo?.1	¿Qué sentimientos te genera consumir artículos de lujo?	¿Cuál/es son las razones para que consumas artículos de lujo?
0	2025/01/29 9:46:06 p. m. CET	Mujer	18 - 25	< 10.000 €	Estudiante	Sí, alguna vez	NaN	NaN	NaN	NaN	...	NaN	Chanel, Rolex, YvesSaintLaurent, Louis Vuitton	Elegancia accesible para pocas personas	Distinción	Joyería	Poco accesible, precio alto, calidad alta	Precio alto;Calidad alta;Exclusividad o escase...	Admiración, ambición	Alegría, satisfacción personal por conseguir a...	más calidad y durabilidad, poderlo pasar de ge...
1	2025/01/29 9:53:04 p. m. CET	Mujer	18 - 25	< 10.000 €	Estudiante	Sí, alguna vez	NaN	NaN	NaN	NaN	...	NaN	Gucci, Dior, Balenciaga, Chanel, StellaMcartne...	Es poder tener algo innecesario que facilita l...	Placer	Gran calidad, precios elevados, poca accesibil...	Gran calidad, precio elevado, historia, artesa...	Precio alto;Calidad alta;Exclusividad o escase...	Ambición, relajación	Satisfacción, relajación, felicidad	Para tener artículos de calidad que cuenten un...
2	2025/01/30 8:50:54 p. m. CET	Mujer	51 - 60	100.000 - 200.000 €	Autónomo/a o profesional independiente	Sí, alguna vez	NaN	NaN	NaN	NaN	...	NaN	Rolex, Bottega veneta, loewe	Excepcional calidad en la materia prima y la e...	Calidad	Elegancia	Alta calidad , exclusividad, elegancia.	Precio alto;Calidad alta;Exclusividad o escasez	Placer	Placer	Me gusta disfrutar de la calidad excepcional
3	2025/02/02 10:01:49 a. m. CET	Hombre	51 - 60	> 500.000 €	Directora/o o Socia/o de una empresa	Sí, alguna vez	NaN	NaN	NaN	NaN	...	NaN	Gucci, Prada, Loewe	Darle un homenaje	Placer	Con trabajo bien hecho	Materiales nobles, buen diseño y una utilidad/...	Precio alto;Calidad alta;Exclusividad o escase...	Admiración	Alegría	Autoafirmación y recompensarse a sí mismo
4	2025/02/02 7:23:47 p. m. CET	Hombre	18 - 25	< 10.000 €	Estudiante	Sí, alguna vez	NaN	NaN	NaN	NaN	...	NaN	Porsche, louis vuitton, cartier.	producto o servicio de gran valor, exclusivo o...	valor	dinero, elegancia, disfrute.	alta calidad, alto valor y exclusividad	Calidad alta;Exclusividad o escasez;Rareza o s...	asombro	emocion	buscar disfrutar de una experiencia única o de...

5 rows × 23 columns

```
[ ] # Visualización preliminar del grupo de NO consumidores de lujo
    grupo_no.head()
```

	Marca temporal	Indique su género	Indique su rango de edad	Indique su rango de ingresos anuales	Indique su ocupación	¿Ha comprado o consumido alguna vez un artículo de moda de lujo? Ya sean prendas, accesorios, joyería, etc.	¿Qué marcas de lujo conoces? Menciona al menos 3.	¿Qué significa para ti la palabra lujo?	Ahora trata de definir lujo en una única palabra	¿Con qué asocias tú el lujo?	...	¿Cuál/es son las razones para que no consumas artículos de lujo?	¿Qué marcas de lujo conoces? Menciona al menos 3..1	¿Qué significa para ti la palabra lujo?.1	Ahora trata de definir lujo en una única palabra.1	¿Con qué asocias tú el lujo?.1	¿Qué características debe tener, en tu opinión, un articulo de lujo?.1	¿Cuáles de estos factores consideras que están relacionados con el lujo?.1	¿Qué sentimientos te genera el concepto de lujo?.1	¿Qué sentimientos te genera consumir artículos de lujo?	¿Cuál/es son las razones para que consumas artículos de lujo?
12	2025/02/07 2:23:25 a. m. CET	Mujer	18 - 25	< 10.000 €	Estudiante	No, nunca	Dior, Loewe, Channel, Miu Miu, Versace, Louis ...	Representa status, poder.	Exclusivo	Dinero	...	Puedes encontrar muchos artículos únicos y on...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
13	2025/02/07 8:27:50 a. m. CET	Mujer	18 - 25	< 10.000 €	Estudiante	No, nunca	Luis Vuitton, Gucci, Balenciaga	Algo que te da un cierto estatus alto	Dinero	Dinero	...	No tengo dinero y no creo que sea apropiado pa...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14	2025/02/07 8:52:49 a. m. CET	Mujer	18 - 25	< 10.000 €	Otro	No, nunca	Balenciaga, Prada, Dior, Balmain, Chiloe, etc	Para mí el lujo siempre lo he identificado com...	Privilegio	Poder	...	No tener un poder adquisitivo que me lo permita	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16	2025/02/07 10:10:53 a. m. CET	Hombre	18 - 25	< 10.000 €	Estudiante	No, nunca	Rolex, Dior, Prada	Caro, buena calidad y limitado	Exclusivo	Dinero	...	Dinero y lo horterass que son	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
17	2025/02/07 10:19:02 a. m. CET	Hombre	18 - 25	< 10.000 €	Estudiante	No, nunca	Dior, Hermes, Bvulgary	Poder	Poder	Riqueza	...	No tengo dinero	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 23 columns

Preguntas a quienes no han consumido artículos de moda de lujo con anterioridad

```
# Veamos las preguntas/columnas de la encuesta a los NO consumidores
preguntas= grupo_no.columns
preguntas
```

```
Index(['Marca temporal', 'Indique su género', 'Indique su rango de edad',
       'Indique su rango de ingresos anuales', 'Indique su ocupación',
       '¿Ha comprado o consumido alguna vez un artículo de moda de lujo? Ya sean prendas, accesorios, joyería, etc.',
       '¿Qué marcas de lujo conoces? Menciona al menos 3.',
       '¿Qué significa para ti la palabra lujo?',
       'Ahora trata de definir lujo en una única palabra',
       '¿Con qué asocias tú el lujo?',
       '¿Qué características debe tener, en tu opinión, un articulo de lujo?',
       '¿Cuáles de estos factores consideras que están relacionados con el lujo?',
       '¿Qué sentimientos te genera el concepto de lujo?',
       '¿Cuál/es son las razones para que no consumas artículos de lujo?',
       '¿Qué marcas de lujo conoces? Menciona al menos 3..1',
       '¿Qué significa para ti la palabra lujo?.1',
       'Ahora trata de definir lujo en una única palabra.1',
       '¿Con qué asocias tú el lujo?.1',
       '¿Qué características debe tener, en tu opinión, un articulo de lujo?.1',
       '¿Cuáles de estos factores consideras que están relacionados con el lujo?.1',
       '¿Qué sentimientos te genera el concepto de lujo?.1',
       '¿Qué sentimientos te genera consumir artículos de lujo?',
       '¿Cuál/es son las razones para que consumas artículos de lujo?'],
      dtype='object')
```

```
# Instalamos el modelo que vamos a usar para hacer LDA
!python -m spacy download es_core_news_sm
```

```
Collecting es-core-news-sm==3.7.0
  Downloading https://github.com/explosion/spacy-models/releases/download/es_core_news_sm-3.7.0/es_core_news_sm-3.7.0-py3-none-any.whl (12.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.9/12.9 MB 88.6 MB/s eta 0:00:00
Requirement already satisfied: spacy<3.8.0,>=3.7.0 in /usr/local/lib/python3.11/dist-packages (from es-core-news-sm==3.7.0) (3.7.5)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.0.12)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.0.11)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.0.9)
Requirement already satisfied: thinc<8.3.0,>=8.2.2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (8.2.5)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.5.1)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (0.4.1)
Requirement already satisfied: typer<1.0.0,>=0.3.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (0.15.1)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (4.67.1)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.32.3)
```

```
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (13.9.4)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from weasel<0.5.0,>=0.1.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (0.20.0)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.11/dist-packages (from weasel<0.5.0,>=0.1.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (7.1.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.0.2)
Requirement already satisfied: marisa-trie>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from language-data>=1.2->langcodes<4.0.0,>=3.2.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.2.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.18.0)
Requirement already satisfied: wrapt in /usr/local/lib/python3.11/dist-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.1.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.17.2)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (0.1.2)
Installing collected packages: es-core-news-sm
Successfully installed es-core-news-sm-3.7.0
✓ Download and installation successful
You can now load the package via spacy.load('es_core_news_sm')
⚠ Restart to reload dependencies
If you are in a Jupyter or Colab notebook, you may need to restart Python in
order to load all the package's dependencies. You can do this by selecting the
'Restart kernel' or 'Restart runtime' option.
```

1. ¿Qué marcas de lujo conoces? Menciona al menos 3

```
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Qué marcas de lujo conoces? Menciona al menos 3."  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_no.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta1 = "\n".join(grupo_no[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta1:")
    print(pregunta1)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta1:
Dior, Loewe, Channel, Miu Miu, Versace, Louis Vuitton, Fendi, Balenciaga
Luis Vuitton, Gucci, Balenciaga
Balenciaga, Prada, Dior, Balmain, Chlöe, etc
Rolex, Dior, Prada
Dior, Hermes, Bvulgary
Rolex, rolls Royce, Riofrio
Ferrari, Lamborghini, LVMH
Chanel, Dior, Louis Vuitton, Cartier, Rolex
<ipython-input-7-5dd608f8d772>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)
```

```
[ ] # Comprobamos el contenido del documento
    print(pregunta1)
```

```
Dior, Loewe, Channel, Miu Miu, Versace, Louis Vuitton, Fendi, Balenciaga
Luis Vuitton, Gucci, Balenciaga
Balenciaga, Prada, Dior, Balmain, Chlöe, etc
Rolex, Dior, Prada
Dior, Hermes, Bvulgary
Rolex, rolls Royce, Riofrio
Ferrari, Lamborghini, LVMH
Chanel, Dior, Louis Vuitton, Cartier, Rolex
```

Vemos que algunas marcas no están bien escritas, por lo que antes de analizar nada, crearemos un diccionario de mappeo para que aunque no estén escritas igual, se puedan analizar como una misma marca.

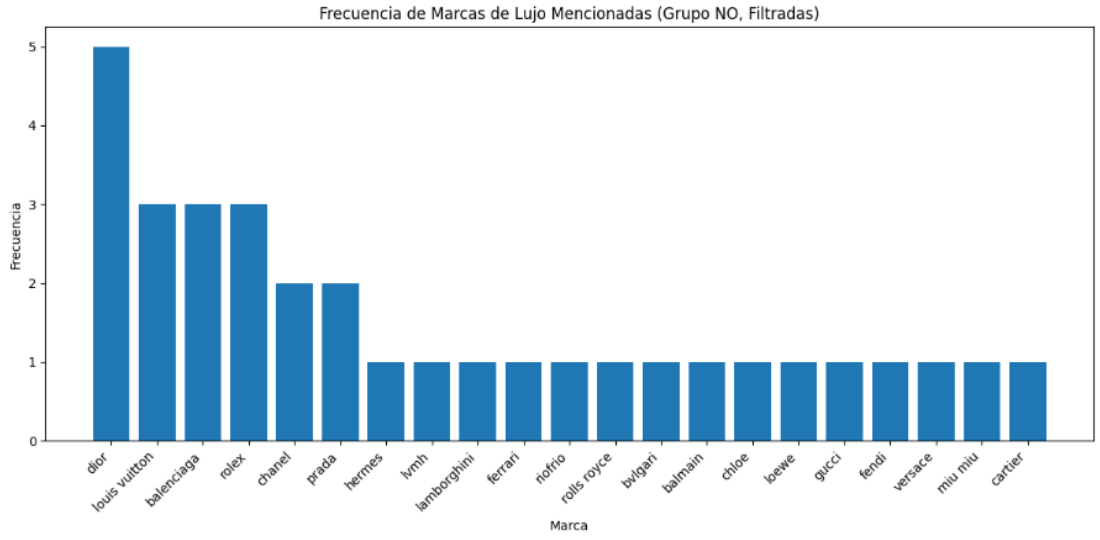
```
[ ] # Instalamos un paquete necesario
    !pip install unidecode
```

```
Collecting unidecode
  Downloading Unidecode-1.3.8-py3-none-any.whl.metadata (13 kB)
Downloading Unidecode-1.3.8-py3-none-any.whl (235 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 235.5/235.5 kB 6.8 MB/s eta 0:00:00
Installing collected packages: unidecode
Successfully installed unidecode-1.3.8
```

```
from collections import Counter
import unidecode
import pandas as pd
import matplotlib.pyplot as plt

# Diccionario de mapeo ampliado para corregir nombres de marcas
mapeo_marcas_no = {
    "channel": "chanel",
    "luis vuitton": "louis vuitton",
    "bvulgary": "bvlgari",
    "miu miu": "miu miu",
    "balanciaga": "balenciaga",
    "lvmh": "lvmh",
    "versace": "versace",
    "rolls royce": "rolls royce",
    "riofrio": "riofrio",
}

# Lista de palabras no deseadas
palabras_excluidas_no = {"etc"}

# Función para normalizar texto
def normalizar_texto_no(texto):
    if isinstance(texto, str):
        return unidecode.unidecode(texto.strip().lower())
    return texto

# Función para mappear las marcas
def procesar_marca_no(marca):
    marca_normalizada = normalizar_texto_no(marca)
    if marca_normalizada in palabras_excluidas_no:
        return None
    return mapeo_marcas_no.get(marca_normalizada, marca_normalizada)

# Procesar directamente el contenido de 'pregunta1_no'
respuestas_no = pregunta1.split("\n")  # Dividir las respuestas por líneas
print("Respuestas originales (Grupo NO):")
print(respuestas_no[:10])

# Dividir cada respuesta en marcas separadas por comas, normalizar y mapear
opciones_no = [marca.strip() for respuesta in respuestas_no for marca in respuesta.split(",")]
opciones_mapeadas_no = [procesar_marca_no(marca) for marca in opciones_no]

# Filtrar las opciones para eliminar palabras no deseadas
opciones_filtradas_no = [marca for marca in opciones_mapeadas_no if marca not in palabras_excluidas_no and marca is not None]

# Contar la frecuencia de cada marca
frecuencias_no = Counter(opciones_filtradas_no)

# Convertir a DataFrame para visualización
```

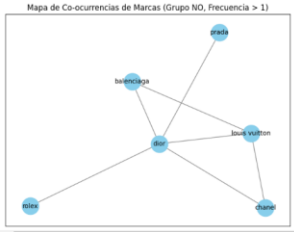
```python
# Convertir a DataFrame para visualización
frecuencias_df_no = pd.DataFrame(frecuencias_no.items(), columns=["Marca", "Frecuencia"]).sort_values(by="Frecuencia", ascending=False)

# Mostrar las frecuencias
print("Frecuencias de las marcas (Grupo NO, corregidas y filtradas):")
print(frecuencias_df_no)

# Gráfico de barras
if not frecuencias_df_no.empty:
    plt.figure(figsize=(12, 6))
    plt.bar(frecuencias_df_no["Marca"], frecuencias_df_no["Frecuencia"])
    plt.xticks(rotation=45, ha="right")
    plt.title("Frecuencia de Marcas de Lujo Mencionadas (Grupo NO, Filtradas)")
    plt.xlabel("Marca")
    plt.ylabel("Frecuencia")
    plt.tight_layout()
    plt.show()
else:
    print("No se encontraron marcas válidas para generar el gráfico.")
```

```
Respuestas originales (Grupo NO):
['Dior, Loewe, Channel, Miu Miu, Versace, Louis Vuitton, Fendi, Balenciaga', 'Luis Vuitton, Gucci, Balenciaga', 'Balenciaga, Prada, Dior, Balmain, Chlöe
Frecuencias de las marcas (Grupo NO, corregidas y filtradas):
           Marca  Frecuencia
0           dior           5
5   louis vuitton          3
7      balenciaga          3
12         rolex           3
2         chanel           2
9          prada           2
13        hermes           1
19           lvmh          1
18    lamborghini           1
17        ferrari           1
16       riofrio           1
15    rolls royce           1
14        bvlgari           1
10        balmain           1
11          chloe           1
1           loewe           1
8           gucci           1
6           fendi           1
4         versace           1
3         miu miu           1
20        cartier           1
```



```python
from itertools import combinations
from collections import Counter
import networkx as nx

# Crear lista de respuestas divididas y mapear marcas
opciones_coocurrencias_no = [
    [procesar_marca_no(marca.strip()) for marca in respuesta.split(",") if procesar_marca_no(marca.strip())]
    for respuesta in respuestas_no
]

# Filtrar respuestas vacías o con una sola marca
opciones_coocurrencias_no = [resp for resp in opciones_coocurrencias_no if len(resp) > 1]

# Verificar si hay datos para generar combinaciones
print("Respuestas procesadas para co-ocurrencias (Grupo NO):", opciones_coocurrencias_no[:10])

# Generar combinaciones de co-ocurrencias
coocurrencias_no = Counter(
    combo for respuesta in opciones_coocurrencias_no for combo in combinations(sorted(set(respuesta)), 2)
)

# Verificar si se han generado co-ocurrencias
print("Co-ocurrencias sin filtrar (Grupo NO):", coocurrencias_no.most_common(10))

# Convertir a DataFrame
coocurrencias_df_no = pd.DataFrame(coocurrencias_no.items(), columns=["Par", "Frecuencia"]).sort_values(by="Frecuencia", ascending=False)

# Print de las Coocurrencias ordenadas por frecuencias
print("Coocurrencias ordenadas por frecuencias (Grupo NO):")
print(coocurrencias_df_no.head())

# Verificar si hay datos después del filtrado
if coocurrencias_df_no.empty:
    print("⚠ No hay co-ocurrencias con frecuencia > 1 en el Grupo NO.")

# Visualizar co-ocurrencias como red
G_no = nx.Graph()
for (op1, op2), freq in coocurrencias_no.items():
    if freq > 1:  # Filtrar para frecuencias mayores a 1
        G_no.add_edge(op1, op2, weight=freq)

# Configuración del gráfico de red
plt.figure(figsize=(8, 6))
pos_no = nx.spring_layout(G_no, k=0.5)  # Diseño de la red
nx.draw_networkx_nodes(G_no, pos_no, node_size=700, node_color="skyblue")
nx.draw_networkx_edges(G_no, pos_no, width=[G_no[u][v]['weight'] / 2 for u, v in G_no.edges()], edge_color="gray")
nx.draw_networkx_labels(G_no, pos_no, font_size=10, font_color="black")
plt.title("Mapa de Co-ocurrencias de Marcas (Grupo NO, Frecuencia > 1)")
plt.show()
```

```
Respuestas procesadas para co-ocurrencias (Grupo NO): [['dior', 'loewe', 'chanel', 'miu miu', 'versace', 'louis vuitton', 'fendi', 'balenciaga'], ['louis vuitton', 'gucci', 'balenciaga'], ['balenciaga', 'prada', 'dior', 'balmain', 'chloe'], ['rolex', 'dior', 'prada'], ['dior', 'hermes', 'bvlgari'], ['rolex', 'rol
Co-ocurrencias sin filtrar (Grupo NO): [(('balenciaga', 'dior'), 2), ((balenciaga', 'louis vuitton'), 2), ((chanel', 'dior'), 2), ((chanel', 'louis vuitton'), 2), ((dior', 'louis vuitton'), 2), ((dior', 'prada'), 2), ((dior', 'rolex'), 2), ((balenciaga', 'chanel'), 1), ((balenciaga', 'fendi'), 1), ((bale
Coocurrencias ordenadas por frecuencias (Grupo NO):
                          Par  Frecuencia
15       (dior, louis vuitton)           2
1        (balenciaga, dior)              2
4   (balenciaga, louis vuitton)          2
7          (chanel, dior)                2
10      (chanel, louis vuitton)          2
```



2. ¿Qué significa para ti la palabra lujo?

```python
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Qué significa para ti la palabra lujo?"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_no.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta2 = "\n".join(grupo_no[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta2:")
    print(pregunta2)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta2:
Representa status, poder..
Algo que te da un cierto estatus alto
Para mí el lujo siempre lo he identificado como algo limitado y excepcional, pero cada vez ves a más personas con artículos de lujo, por lo que ya no me genera tanto interés. Además también dudo de la calidad de los productos, en el sentido de que creo que lo que pagas es la marca.
Caro, buena calidad y limitado
Poder
Capricho
Cosa de re pijos
Lujo es exclusividad a medida
<ipython-input-13-1a8829446c823>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)
```

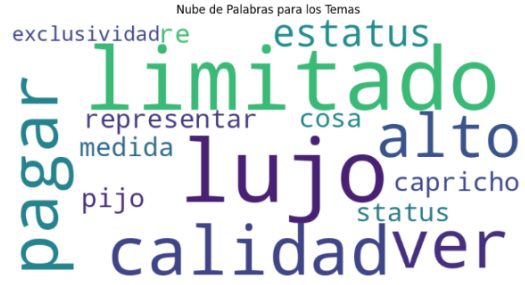
```python
# Comprobamos el contenido del documento
print(pregunta2)
```

```
Representa status, poder..
Algo que te da un cierto estatus alto
Para mí el lujo siempre lo he identificado como algo limitado y excepcional, pero cada vez ves a más personas con artículos de lujo, por lo que ya no me genera tanto interés. Además también dudo de la calidad de los productos, en el sentido de que creo que lo que pagas es la marca.
Caro, buena calidad y limitado
Poder
Capricho
Cosa de re pijos
Lujo es exclusividad a medida
```

```python
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens


# 🔹 Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
```

```python
# 🔹 Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# 🔹 Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n🔹 Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# 🔹 Procesar y analizar el contenido de pregunta2
if 'pregunta2' in locals():  # Verificar que pregunta2 existe
    # Limpiar las respuestas de pregunta2
    cleaned_responses = clean_responses(pregunta2)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta2' no está definido. Asegúrate de haberlo creado previamente.")
```

```
🔹 Temas generados:
Tema 1: 0.085*"lujo" + 0.084*"limitado" + 0.084*"calidad" + 0.048*"ver" + 0.048*"pagar"
Tema 2: 0.097*"alto" + 0.097*"estatus" + 0.097*"representar" + 0.097*"status" + 0.097*"capricho"
Tema 3: 0.091*"pijo" + 0.091*"medida" + 0.091*"cosa" + 0.091*"re" + 0.091*"exclusividad"
```



3. Ahora trata de definir lujo en una única palabra

```
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "Ahora trata de definir lujo en una única palabra"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_no.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta3 = "\n".join(grupo_no[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta3:")
    print(pregunta3)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta3:
Exclusivo
Dinero
Privilegio
Exclusivo
Poder
Poder adquisitivo
Exclusivo
Experiencia
<ipython-input-16-92c481ccdc1b>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)
```

```
[ ] # Comprobamos el contenido del documento
    print(pregunta3)
```

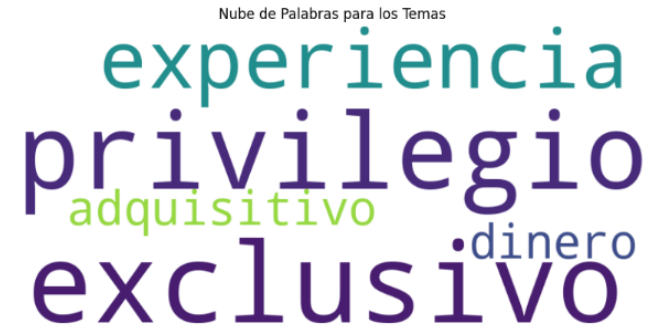
```
Exclusivo
Dinero
Privilegio
Exclusivo
Poder
Poder adquisitivo
Exclusivo
Experiencia
```

```
[ ] import spacy
    from gensim import corpora
    from gensim.models import LdaModel
    from spacy.matcher import PhraseMatcher
    import matplotlib.pyplot as plt
```

```python
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens
```

```python
# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta3
if 'pregunta3' in locals():  # Verificar que pregunta3 existe
    # Limpiar las respuestas de pregunta3
    cleaned_responses = clean_responses(pregunta3)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta3' no está definido. Asegúrate de haberlo creado previamente.")
```

```
 Temas generados:
Tema 1: 0.588*"exclusivo" + 0.234*"privilegio" + 0.060*"experiencia" + 0.059*"adquisitivo" + 0.059*"dinero"
Tema 2: 0.494*"dinero" + 0.127*"exclusivo" + 0.127*"privilegio" + 0.126*"adquisitivo" + 0.126*"experiencia"
Tema 3: 0.362*"experiencia" + 0.361*"adquisitivo" + 0.093*"exclusivo" + 0.092*"privilegio" + 0.092*"dinero"
```



4. ¿Con qué asocias tú el lujo?

```
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Con qué asocias tú el lujo?"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_no.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta4 = "\n".join(grupo_no[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta4:")
    print(pregunta4)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta4:
Dinero
Dinero
Poder
Dinero
Riqueza
Esfuerzo
Mujeres/coches
Comodidad y poder
<ipython-input-19-1fd16c3b28b9>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)
```

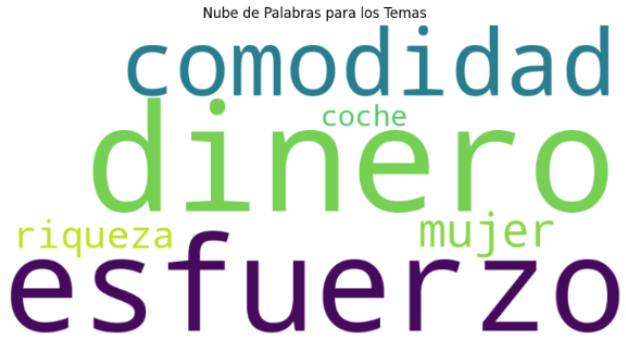
```
[ ] # Comprobamos el contenido del documento
    print(pregunta4)
```

```
Dinero
Dinero
Poder
Dinero
Riqueza
Esfuerzo
Mujeres/coches
Comodidad y poder
```

```python
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens
```

```python
# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta4
if 'pregunta4' in locals():  # Verificar que pregunta4 existe
    # Limpiar las respuestas de pregunta4
    cleaned_responses = clean_responses(pregunta4)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta4' no está definido. Asegúrate de haberlo creado previamente.")
```

```
 Temas generados:
Tema 1: 0.169*"dinero" + 0.168*"esfuerzo" + 0.168*"comodidad" + 0.167*"riqueza" + 0.164*"mujer"
Tema 2: 0.223*"coche" + 0.223*"mujer" + 0.221*"comodidad" + 0.221*"esfuerzo" + 0.057*"dinero"
Tema 3: 0.554*"dinero" + 0.221*"riqueza" + 0.057*"esfuerzo" + 0.056*"comodidad" + 0.056*"mujer"
```



5. ¿Qué características debe tener, en tu opinión un artículo de lujo?

```
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Qué características debe tener, en tu opinión, un articulo de lujo?"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_no.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta5 = "\n".join(grupo_no[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta5:")
    print(pregunta5)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta5:
Buena calidad y materiales. Un proceso de fabricación sostenible, artesanal, etc.
Calidad y autenticidad
Primordialmente, calidad y que sean sostenibles
tiene q ser limitado y por lo general llamativo
Exclusivo
Revalorizable
Que sea caro, de mucha calidad y sea exclusivo
Gran calidad, poco común, exclusivo
<ipython-input-21-4d38a9a85939>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)
```

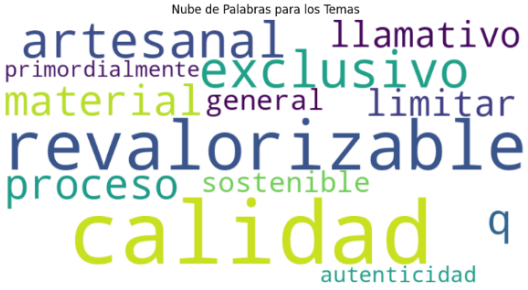
```
[ ] # Comprobamos el contenido del documento
    print(pregunta5)
```

```
Buena calidad y materiales. Un proceso de fabricación sostenible, artesanal, etc.
Calidad y autenticidad
Primordialmente, calidad y que sean sostenibles
tiene q ser limitado y por lo general llamativo
Exclusivo
Revalorizable
Que sea caro, de mucha calidad y sea exclusivo
Gran calidad, poco común, exclusivo
```

```
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens
```

```python
# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta5
if 'pregunta5' in locals():  # Verificar que pregunta5 existe
    # Limpiar las respuestas de pregunta5
    cleaned_responses = clean_responses(pregunta5)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta5' no está definido. Asegúrate de haberlo creado previamente.")
```

```
 Temas generados:
Tema 1: 0.172*"exclusivo" + 0.172*"calidad" + 0.069*"artesanal" + 0.069*"material" + 0.069*"proceso"
Tema 2: 0.142*"q" + 0.142*"limitar" + 0.142*"llamativo" + 0.142*"general" + 0.036*"revalorizable"
Tema 3: 0.226*"calidad" + 0.129*"sostenible" + 0.128*"primordialmente" + 0.128*"autenticidad" + 0.033*"revalorizable"
```



6. ¿Cuáles de estos factores consideras que están relacionados con el lujo?

```
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Cuáles de estos factores consideras que están relacionados con el lujo?"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_no.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta6 = "\n".join(grupo_no[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta6:")
    print(pregunta6)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta6:
Precio alto;Calidad alta;Exclusividad o escasez;Superficialidad
Precio alto;Calidad alta;Exclusividad o escasez;Superficialidad
Precio alto;Exclusividad o escasez;Expresión artística
Precio alto;Calidad alta;Exclusividad o escasez;Superficialidad
Precio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad
Precio alto;Exclusividad o escasez;Rareza o singularidad
Precio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad
Precio alto;Calidad alta;Exclusividad o escasez;Expresión artística
<ipython-input-24-2ab64097a9b9>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)
```

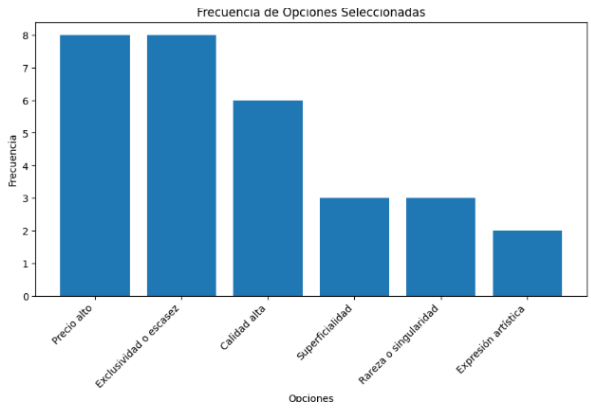
```
[ ] # Comprobamos el contenido del documento
    print(pregunta6)
```

```
Precio alto;Calidad alta;Exclusividad o escasez;Superficialidad
Precio alto;Calidad alta;Exclusividad o escasez;Superficialidad
Precio alto;Exclusividad o escasez;Expresión artística
Precio alto;Calidad alta;Exclusividad o escasez;Superficialidad
Precio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad
Precio alto;Exclusividad o escasez;Rareza o singularidad
Precio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad
Precio alto;Calidad alta;Exclusividad o escasez;Expresión artística
```

Para esta pregunta, al ser multirrespuesta, en vez de usar un modelo LDA, calcularemos las frecuencias de cada uno de los factores que se ofrecen como respuestas y haremos una matriz de co-ocurrencias para ver qué factores se suelen dar de forma conjunta como respuesta

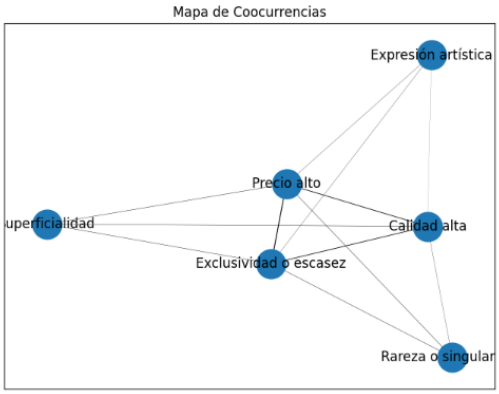
```
from collections import Counter

# Dividir las respuestas en opciones individuales
opciones = [respuesta.split(";") for respuesta in df[columna_respuestas].dropna()]

# Contar la frecuencia de cada opción
frecuencias = Counter(opcion.strip() for respuesta in opciones for opcion in respuesta)

# Mostrar las frecuencias
print("Frecuencias de las opciones:")
for opcion, frecuencia in frecuencias.items():
    print(f"{opcion}: {frecuencia}")

# Convertir a DataFrame para visualización
import pandas as pd
import matplotlib.pyplot as plt

frecuencias_df = pd.DataFrame(frecuencias.items(), columns=["Opción", "Frecuencia"]).sort_values(by="Frecuencia", ascending=False)

# Gráfico de barras
plt.figure(figsize=(10, 5))
plt.bar(frecuencias_df["Opción"], frecuencias_df["Frecuencia"])
plt.xticks(rotation=45, ha="right")
plt.title("Frecuencia de Opciones Seleccionadas")
plt.xlabel("Opciones")
plt.ylabel("Frecuencia")
plt.show()
```

```
Frecuencias de las opciones:
Precio alto: 8
Calidad alta: 6
Exclusividad o escasez: 8
Superficialidad: 3
Expresión artística: 2
Rareza o singularidad: 3
```




```
[ ] from itertools import combinations
    from collections import Counter

    # Crear lista de respuestas divididas
    opciones = [respuesta.split(";") for respuesta in df[columna_respuestas].dropna()]

    # Generar combinaciones de coocurrencia
    coocurrencias = Counter(
        combo for respuesta in opciones for combo in combinations(sorted(opcion.strip() for opcion in respuesta), 2)
    )

    # Convertir a DataFrame
    coocurrencias_df = pd.DataFrame(coocurrencias.items(), columns=["Par", "Frecuencia"]).sort_values(by="Frecuencia", ascending=False)

    # Mostrar las coocurrencias más frecuentes
    print("Coocurrencias más frecuentes:")
    print(coocurrencias_df.head())

    # Visualizar coocurrencias (opcional)
    import networkx as nx
    import matplotlib.pyplot as plt

    G = nx.Graph()
    for (op1, op2), freq in coocurrencias.items():
        G.add_edge(op1, op2, weight=freq)

    plt.figure(figsize=(8, 6))
    pos = nx.spring_layout(G, k=0.5)
    nx.draw_networkx_nodes(G, pos, node_size=700)
    nx.draw_networkx_edges(G, pos, width=[G[u][v]['weight'] / 10 for u, v in G.edges()])
    nx.draw_networkx_labels(G, pos)
    plt.title("Mapa de Coocurrencias")
    plt.show()
```

```
Coocurrencias más frecuentes:
                                        Par  Frecuencia
3      (Exclusividad o escasez, Precio alto)          8
0     (Calidad alta, Exclusividad o escasez)          6
1               (Calidad alta, Precio alto)          6
2          (Calidad alta, Superficialidad)          3
4  (Exclusividad o escasez, Superficialidad)          3
```



7. ¿Qué sentimientos te genera el concepto de lujo?

```
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Qué sentimientos te genera el concepto de lujo?"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_no.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta7 = "\n".join(grupo_no[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta7:")
    print(pregunta7)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta7:
Contradictorios. Ciertas artículos de lujo pueden representar elegancia, pero otros todo lo contrario.
Aspiracion
Ahora mismo, malos, pero también depende de la marca
Indiferente
Asco
Desconocimiento
Indiferencia
Deseo
<ipython-input-32-3bd86fe63bd8>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)
```

```
[ ] # Comprobamos el contenido del documento
    print(pregunta7)
```

```
Contradictorios. Ciertas artículos de lujo pueden representar elegancia, pero otros todo lo contrario.
Aspiracion
Ahora mismo, malos, pero también depende de la marca
Indiferente
Asco
Desconocimiento
Indiferencia
Deseo
```

```python
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens
```

```python
# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta7
if 'pregunta7' in locals():  # Verificar que pregunta7 existe
    # Limpiar las respuestas de pregunta7
    cleaned_responses = clean_responses(pregunta7)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta7' no está definido. Asegúrate de haberlo creado previamente.")
```

```
 Temas generados:
Tema 1: 0.089*"contradictorio" + 0.089*"artículo" + 0.089*"representar" + 0.089*"contrario" + 0.089*"lujo"
Tema 2: 0.166*"desear" + 0.166*"desconocimiento" + 0.166*"asco" + 0.042*"indiferente" + 0.042*"aspiracion"
Tema 3: 0.189*"indiferencia" + 0.189*"aspiracion" + 0.049*"indiferente" + 0.048*"asco" + 0.048*"desconocimiento"
```

8. ¿Cuáles son las razones para que no consumas productos de lujo?

```
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Cuál/es son las razones para que no consumas artículos de lujo?"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_no.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta8 = "\n".join(grupo_no[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta8:")
    print(pregunta8)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta8:
Puedes encontrar muchos artículos únicos y originales por un precio mucho más asequible
No tengo dinero y no creo que sea apropiado para mi edad, soy muy joven para tener artículos tan caros
No tener un poder adquisitivo que me lo permita
Dinero y lo horteras que son
No tengo dinero
Mi poder adquisitivo, chip soy pobre
Capacidad adquisitiva
Dinero
<ipython-input-35-6bc2d9ad65a4>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  grupo_no[columna_respuestas] = grupo_no[columna_respuestas].replace(r"\n", " ", regex=True)
```

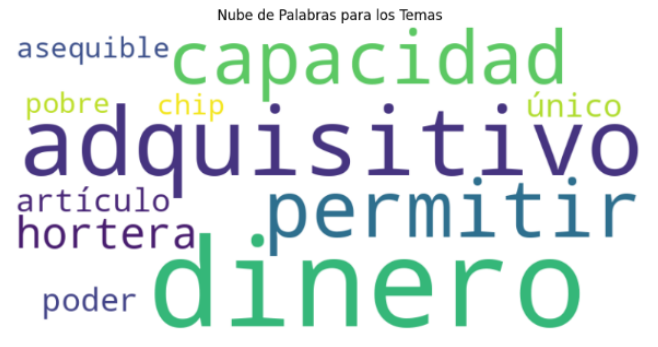
```
[ ] # Comprobamos el contenido del documento
    print(pregunta8)
```

```
Puedes encontrar muchos artículos únicos y originales por un precio mucho más asequible
No tengo dinero y no creo que sea apropiado para mi edad, soy muy joven para tener artículos tan caros
No tener un poder adquisitivo que me lo permita
Dinero y lo horteras que son
No tengo dinero
Mi poder adquisitivo, chip soy pobre
Capacidad adquisitiva
Dinero
```

```
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens
```

```python
# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta8
if 'pregunta8' in locals():  # Verificar que pregunta8 existe
    # Limpiar las respuestas de pregunta8
    cleaned_responses = clean_responses(pregunta8)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta8' no está definido. Asegúrate de haberlo creado previamente.")
```

```
 Temas generados:
Tema 1: 0.337*"dinero" + 0.117*"hortera" + 0.034*"adquisitivo" + 0.034*"permitir" + 0.034*"capacidad"
Tema 2: 0.122*"artículo" + 0.070*"dinero" + 0.069*"poder" + 0.069*"único" + 0.069*"asequible"
Tema 3: 0.256*"adquisitivo" + 0.102*"chip" + 0.102*"pobre" + 0.102*"capacidad" + 0.102*"permitir"
```



Preguntas a quienes SÍ han consumido moda de lujo con anterioridad

```
[ ] # Filtrar los datos y crear copias explícitas
    grupo_si = df[df[columna_filtro] == "Sí, alguna vez"].copy()
    grupo_no = df[df[columna_filtro] == "No, nunca"].copy()
```

```
# Veamos las preguntas/columnas de la encuesta a los SÍ consumidores
preguntas= grupo_si.columns
preguntas
```

```
Index(['Marca temporal', 'Indique su género', 'Indique su rango de edad',
       'Indique su rango de ingresos anuales', 'Indique su ocupación',
       '¿Ha comprado o consumido alguna vez un artículo de moda de lujo? Ya sean prendas, accesorios, joyería, etc.',
       '¿Qué marcas de lujo conoces? Menciona al menos 3.',
       '¿Qué significa para ti la palabra lujo?',
       'Ahora trata de definir lujo en una única palabra',
       '¿Con qué asocias tú el lujo?',
       '¿Qué características debe tener, en tu opinión, un articulo de lujo?',
       '¿Cuáles de estos factores consideras que están relacionados con el lujo?',
       '¿Qué sentimientos te genera el concepto de lujo?',
       '¿Cuál/es son las razones para que no consumas artículos de lujo?',
       '¿Qué marcas de lujo conoces? Menciona al menos 3..1',
       '¿Qué significa para ti la palabra lujo?.1',
       'Ahora trata de definir lujo en una única palabra.1',
       '¿Con qué asocias tú el lujo?.1',
       '¿Qué características debe tener, en tu opinión, un articulo de lujo?.1',
       '¿Cuáles de estos factores consideras que están relacionados con el lujo?.1',
       '¿Qué sentimientos te genera el concepto de lujo?.1',
       '¿Qué sentimientos te genera consumir artículos de lujo?',
       '¿Cuál/es son las razones para que consumas artículos de lujo?'],
      dtype='object')
```

```
[ ] grupo_si.head()
```

	Marca temporal	Indique su género	Indique su rango de edad	Indique su rango de ingresos anuales	Indique su ocupación	¿Ha comprado o consumido alguna vez un artículo de moda de lujo? Ya sean prendas, accesorios, joyería, etc.	¿Qué marcas de lujo conoces? Menciona al menos 3.	¿Qué significa para ti la palabra lujo?	Ahora trata de definir lujo en una única palabra	¿Con qué asocias tú el lujo?	...	¿Cuál/es son las razones para que no consumas artículos de lujo?	¿Qué marcas de lujo conoces? Menciona al menos 3..1	¿Qué significa para ti la palabra lujo?.1	Ahora trata de definir lujo en una única palabra.1	¿Con qué asocias tú el lujo?.1	¿Qué características debe tener, en tu opinión, un articulo de lujo?.1	¿Cuáles de estos factores consideras que están relacionados con el lujo?.1	¿Qué sentimientos te genera el concepto de lujo?.1	¿Qué sentimientos te genera consumir artículos de lujo?	¿Cuál/es son las razones para que consumas artículos de lujo?
0	2025/01/29 9:46:06 p. m. CET	Mujer	18 - 25	< 10.000 €	Estudiante	Sí, alguna vez	NaN	NaN	NaN	NaN	...	NaN	Chanel, Rolex, YvesSaintLaurent, Louis Vuitton	Elegancia accesible para pocas personas	Distinción	Joyería	Poco accesible, precio alto, calidad alta	Precio alto;Calidad alta;Exclusividad o escase...	Admiración, ambición	Alegría, satisfacción personal por conseguir a...	más calidad y durabilidad, poderlo pasar de ge...
1	2025/01/29 9:53:04 p. m. CET	Mujer	18 - 25	< 10.000 €	Estudiante	Sí, alguna vez	NaN	NaN	NaN	NaN	...	NaN	Gucci, Dior, Balenciaga, Chanel, Stellallcartne...	Es poder tener algo innecesario que facilita l...	Placer	Gran calidad, precios elevados, poca accesibil...	Gran calidad, precio elevado, historia, artesa...	Precio alto;Calidad alta;Exclusividad o escase...	Ambición, relajación	Satisfacción, relajación, felicidad	Para tener artículos de calidad que cuenten un...
2	2025/01/30 6:58:54 p. m. CET	Mujer	51 - 60	100.000 - 200.000 €	Autónomo/a o profesional independiente	Sí, alguna vez	NaN	NaN	NaN	NaN	...	NaN	Rolex, Bottega veneta, loewe	Excepcional calidad en la materia prima y la e...	Calidad	Elegancia	Alta calidad , exclusividad, elegancia.	Precio alto;Calidad alta;Exclusividad o escasez	Placer	Placer	Me gusta disfrutar de la calidad excepcional
3	2025/02/02 10:01:49 a. m. CET	Hombre	51 - 60	> 500.000 €	Director/a o Socio/a de una empresa	Sí, alguna vez	NaN	NaN	NaN	NaN	...	NaN	Gucci, Prada, Loewe	Darle un homenaje	Placer	Con trabajo bien hecho	Materiales nobles, buen diseño y una utilidad/...	Precio alto;Calidad alta;Exclusividad o escase...	Admiración	Alegría	Autoafirmación y recompensarse a sí mismo
4	2025/02/02 7:23:47 p. m. CET	Hombre	18 - 25	< 10.000 €	Estudiante	Sí, alguna vez	NaN	NaN	NaN	NaN	...	NaN	Porsche, louis vuitton, cartier.	producto o servicio de gran valor, exclusivo o...	valor	dinero, elegancia, disfrute.	Calidad alta,exclusividad o escasez;Rareza o s...	Precio alto;Calidad alta;Exclusividad o escase...	asombro	emocion	buscar disfrutar de una experiencia única o de...

5 rows × 23 columns

Podemos ver que hay una pregunta más que en el otro grupo de consumidores, referente a los sentimientos que les genera consumir artículos de lujo

```
# Instalamos el modelo que vamos a usar para hacer LDA
!python -m spacy download es_core_news_sm
```

```
Collecting es-core-news-sm==3.7.0
  Downloading https://github.com/explosion/spacy-models/releases/download/es_core_news_sm-3.7.0/es_core_news_sm-3.7.0-py3-none-any.whl (12.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.9/12.9 MB 51.7 MB/s eta 0:00:00
Requirement already satisfied: spacy<3.8.0,>=3.7.0 in /usr/local/lib/python3.11/dist-packages (from es-core-news-sm==3.7.0) (3.7.5)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.0.12)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.0.11)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.0.9)
Requirement already satisfied: thinc<8.3.0,>=8.2.2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (8.2.5)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.5.1)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (0.4.1)
Requirement already satisfied: typer<1.0.0,>=0.3.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (0.15.1)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (4.67.1)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.32.3)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.10.6)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.1.5)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (75.1.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (24.2)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.5.0)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.26.4)
Requirement already satisfied: language-data>=1.2 in /usr/local/lib/python3.11/dist-packages (from langcodes<4.0.0,>=3.2.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.3.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (0.7.0)
Requirement already satisfied: pydantic-core==2.27.2 in /usr/local/lib/python3.11/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.27.2)
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/python3.11/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2025.1.31)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.11/dist-packages (from thinc<8.3.0,>=8.2.2->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (0.7.11)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.11/dist-packages (from thinc<8.3.0,>=8.2.2->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (0.1.5)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (8.1.8)
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (13.9.4)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from weasel<0.5.0,>=0.1.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (0.20.0)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.11/dist-packages (from weasel<0.5.0,>=0.1.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (7.1.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.0.2)
Requirement already satisfied: marisa-trie>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from language-data>=1.2->langcodes<4.0.0,>=3.2.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.2.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (2.18.0)
Requirement already satisfied: wrapt in /usr/local/lib/python3.11/dist-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.1.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (1.17.2)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.0->es-core-news-sm==3.7.0) (0.1.2)
✓ Download and installation successful
You can now load the package via spacy.load('es_core_news_sm')
⚠ Restart to reload dependencies
If you are in a Jupyter or Colab notebook, you may need to restart Python in
order to load all the package's dependencies. You can do this by selecting the
'Restart kernel' or 'Restart runtime' option.
```

1. ¿Qué marcas de lujo conoces? Menciona al menos 3.

```
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Qué marcas de lujo conoces? Menciona al menos 3..1"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_si.columns:
    # Reemplazar los saltos de línea dentro de las respuestas usando .apply()
    grupo_si[columna_respuestas] = grupo_si[columna_respuestas].apply(lambda x: x.replace("\n", " ") if isinstance(x, str) else x)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    preguntas_si = "\n".join(grupo_si[columna_respuestas].dropna().tolist())

# Verificar el contenido
print("Contenido procesado de pregunta1:")
print(preguntas_si)
```

```
Contenido procesado de pregunta1:
Chanel, Rolex, YvesSaintLaurent, Louis Vuitton
Gucci, Dior, Balenciaga, Chanel, StellaMcCartney, Loro Piana, Brunello Cucinelli, Zegna, YSL, Armani, Etro, MaxMara, Vivienne Westwood, Isabel Marant...
Rolex, Bottega veneta, loewe
Gucci, Prada, Loewe
Porsche, louis vuitton, cartier.
Gucci, gabana, rolex
Ysl, Gucci, Rolex
Louis Vuitton, Prada, Gucci
Loewe Louis Vuitton Chanel
Louis Vouitton, Maison Margela, Gucci
Celine, Ralph Lauren, Rolex, Cartier
Loewe, Prada, LV
Hermes, Louis Voitton, Loro Piana
Prada, gucci, dior
Four seasons, Loewe, cartier
Ferrari, Rolls Royce, rolex
Dior chanel givenchy
Loewe, LV, Chanel
```

```
preguntas_si
```

```
'Chanel, Rolex, YvesSaintLaurent, Louis Vuitton\nGucci, Dior, Balenciaga, Chanel, StellaMcCartney, Loro Piana, Brunello Cucinelli, Zegna, YSL, Armani, Etro, MaxMara, Vivienne Westwood, Isabel Marant...\nRolex, Bottega veneta, loewe\nGucci, Prada, Loewe\nPorsche, louis vuitton, cartier.\nGucci, gabana, rolex\nYsl, Gucci, Rolex\nLouis Vuitton, Prada, Gucci\nLoewe Louis Vuitton Chanel\nLouis Vouitton, Maison Margela, Gucci\nCeline, Ralph Lauren, Rolex, Cartier\nLoewe, Prada, LV\nHermes, Louis Voitton, Loro Piana\nPrada, gucci, dior \nFour seasons, Loewe, cartier\nFerrari, Rolls Royce, rolex\nDior chanel givenchy \nLoewe, LV, Chanel'
```

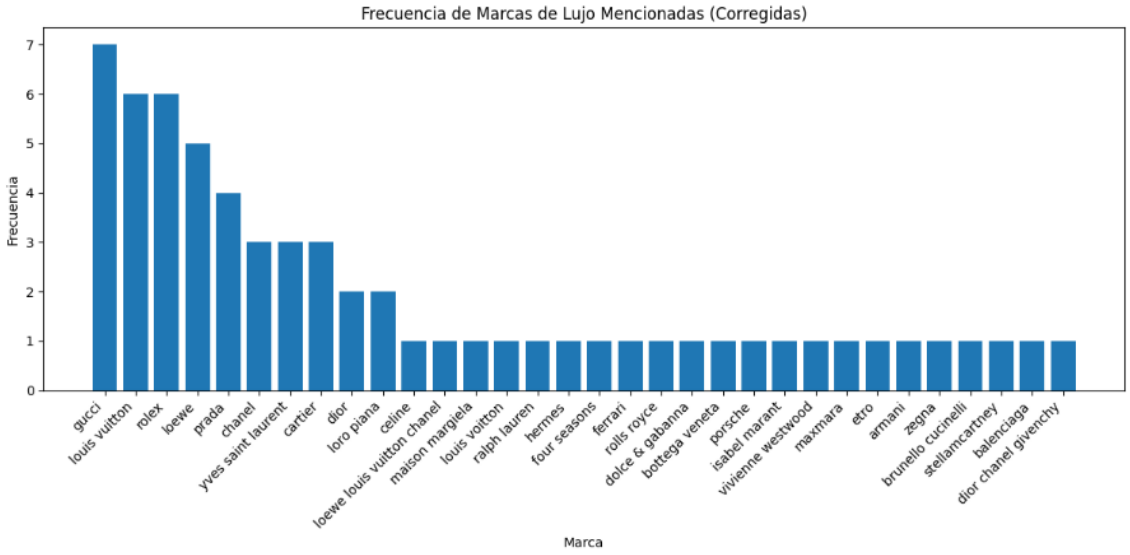
```
# Instalamos un paquete necesario
!pip install unidecode
```

```
Requirement already satisfied: unidecode in /usr/local/lib/python3.11/dist-packages (1.3.8)
```

```python
import re
from collections import Counter
import pandas as pd
import matplotlib.pyplot as plt
import unidecode

# Diccionario de mapeo corregido
mapeo_marcas = {
    "channel": "chanel",
    "luis vuitton": "louis vuitton",
    "lv": "louis vuitton",
    "louis vouitton": "louis vuitton",
    "bvulgary": "bvlgari",
    "cartier.": "cartier",
    "ysl": "yves saint laurent",
    "yvessaintlaurent": "yves saint laurent",
    "miu miu": "miu miu",
    "balanciaga": "balenciaga",
    "gabana": "dolce & gabanna",
    "isabel marant...": "isabel marant",
    "maison margela": "maison margiela",
    "lvmh": "lvmh",
    "rolls royce": "rolls royce",
    "dolce & gabanna": "dolce & gabanna",
    "bottega veneta": "bottega veneta",
    "vivienne westwood": "vivienne westwood",
    "brunello cucinelli": "brunello cucinelli",
    "stella mccartney": "stella mccartney",
}

# Lista de marcas compuestas para evitar separaciones incorrectas
marcas_compuestas = [
    "louis vuitton", "miu miu", "yves saint laurent", "dolce & gabanna",
    "bottega veneta", "vivienne westwood", "brunello cucinelli", "stella mccartney", "maison margiela", "rolls royce"
]

# Lista de palabras no deseadas
palabras_excluidas = {"etc"}

# Función para normalizar texto
def normalizar_texto(texto):
    if isinstance(texto, str):
        return unidecode.unidecode(texto.strip().lower())
    return texto

# Función para mapear marcas y corregir errores
def mapear_marca(marca):
    marca_normalizada = normalizar_texto(marca)
    return mapeo_marcas.get(marca_normalizada, marca_normalizada)
```

```python
# Paso 1: Sustituir marcas compuestas temporalmente
def proteger_marcas_compuestas(texto):
    for marca in marcas_compuestas:
        texto = texto.replace(marca, marca.replace(" ", "_"))  # Sustituye espacios por _
    return texto

# Paso 2: Restaurar las marcas después de la separación
def restaurar_marcas_compuestas(marca):
    return marca.replace("_", " ")  # Vuelve a poner espacios

# Procesar el contenido de 'pregunta1'
respuestas = pregunta1_si.split("\n")  # Dividir respuestas por líneas
respuestas = [proteger_marcas_compuestas(resp) for resp in respuestas]  # Proteger marcas compuestas

# Dividir respuestas en marcas usando expresiones regulares
opciones = [marca.strip() for respuesta in respuestas for marca in re.split(r",| & |/", respuesta)]

# Aplicar mapeo de marcas
opciones_mapeadas = [mapear_marca(marca) for marca in opciones]

# Restaurar marcas compuestas
opciones_finales = [restaurar_marcas_compuestas(marca) for marca in opciones_mapeadas]

# Filtrar marcas no deseadas
opciones_filtradas = [marca for marca in opciones_finales if marca not in palabras_excluidas and marca != ""]

# Contar la frecuencia de cada marca
frecuencias = Counter(opciones_filtradas)

# Convertir a DataFrame para visualización
frecuencias_df = pd.DataFrame(frecuencias.items(), columns=["Marca", "Frecuencia"]).sort_values(by="Frecuencia", ascending=False)

# Mostrar resultados
print("Frecuencias de las marcas (corregidas y filtradas):")
print(frecuencias_df)

# Graficar los resultados
if not frecuencias_df.empty:
    plt.figure(figsize=(12, 6))
    plt.bar(frecuencias_df["Marca"], frecuencias_df["Frecuencia"])
    plt.xticks(rotation=45, ha="right")
    plt.title("Frecuencia de Marcas de Lujo Mencionadas (Corregidas)")
    plt.xlabel("Marca")
    plt.ylabel("Frecuencia")
    plt.tight_layout()
    plt.show()
else:
    print("No se encontraron marcas válidas para generar el gráfico.")
```

```
Frecuencias de las marcas (corregidas y filtradas):
                        Marca  Frecuencia
4                       gucci           7
3                louis vuitton           6
1                       rolex           6
17                      loewe           5
18                      prada           4
0                       chanel           3
2           yves saint laurent           3
20                    cartier           3
5                        dior           2
8                  loro piana           2
24                     celine           1
22  loewe louis vuitton chanel           1
23             maison margiela           1
27              louis voitton           1
25               ralph lauren           1
26                     hermes           1
28               four seasons           1
29                    ferrari           1
30                 rolls royce           1
21            dolce & gabanna           1
16             bottega veneta           1
19                    porsche           1
15               isabel marant           1
14           vivienne westwood           1
13                    maxmara           1
12                      etro           1
11                    armani           1
10                     zegna           1
9          brunello cucinelli           1
7              stellamcartney           1
6                   balenciaga           1
31       dior chanel givenchy           1
```




```
[ ] # Revisemos las respuestas que no se están procesando correctamente
    print(frecuencias_df.loc[22])
    print(frecuencias_df.loc[31])
```

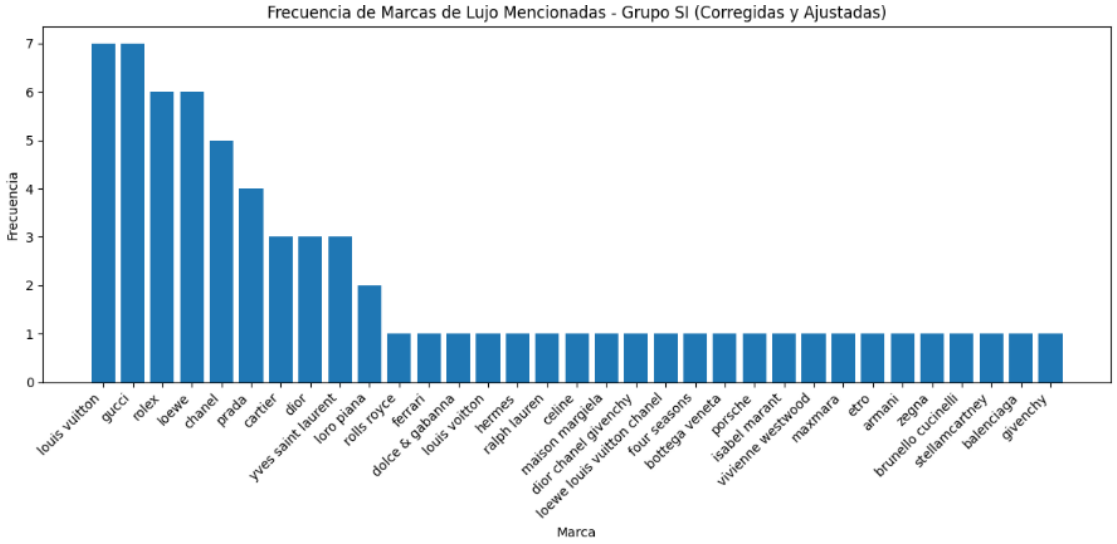
```
Marca         loewe louis vuitton chanel
Frecuencia                             1
Name: 22, dtype: object
Marca         dior chanel givenchy
Frecuencia                       1
Name: 31, dtype: object
```

```
# Tras diversos errores intentando incluir estas respuestas, sumaremos manualmente las frecuencias, eliminando esas dos respuestas
import re
from collections import Counter
import pandas as pd
import matplotlib.pyplot as plt
import unidecode

# Diccionario de mapeo manual de marcas corregidas
mapeo_marcas_si = {
    "channel": "chanel",
    "luis vuitton": "louis vuitton",
    "lv": "louis vuitton",
    "louis vouitton": "louis vuitton",
    "bvulgary": "bvlgari",
    "cartier.": "cartier",
    "ysl": "yves saint laurent",
    "yvessaintlaurent": "yves saint laurent",
    "miu miu": "miu miu",
    "balanciaga": "balenciaga",
    "gabana": "dolce & gabanna",
    "isabel marant...": "isabel marant",
    "maison margela": "maison margiela",
    "lvmh": "lvmh",
    "rolls royce": "rolls royce",
    "dolce & gabanna": "dolce & gabanna",
    "bottega veneta": "bottega veneta",
    "vivienne westwood": "vivienne westwood",
    "brunello cucinelli": "brunello cucinelli",
    "stella mccartney": "stella mccartney",
}

# Lista de palabras no deseadas
palabras_excluidas_si = {"etc"}

# Función para normalizar texto
def normalizar_texto_si(texto):
    if isinstance(texto, str):
        return unidecode.unidecode(texto.strip().lower())
    return texto
```

```python
# Función para mapear marcas manualmente
def mapear_marca_si(marca):
    marca_normalizada = normalizar_texto_si(marca)
    return mapeo_marcas_si.get(marca_normalizada, marca_normalizada)  # No devuelve listas

# Procesar el contenido de 'pregunta1_si' (Grupo SÍ)
respuestas_procesadas_si = pregunta1_si.split("\n")  # Dividir respuestas por líneas

# Excluir respuestas problemáticas (índices 22 y 31)
indices_a_excluir_si = [22, 31]
respuestas_filtradas_si = [respuesta for idx, respuesta in enumerate(respuestas_procesadas_si) if idx not in indices_a_excluir_si]

# Separar marcas usando comas, &, y otras divisiones simples
opciones_si = [marca.strip() for respuesta in respuestas_filtradas_si for marca in re.split(r",| & |/", respuesta)]

# Aplicar mapeo manual de marcas
opciones_mapeadas_si = [mapear_marca_si(marca) for marca in opciones_si]

# Filtrar marcas no deseadas
opciones_finales_si = [marca for marca in opciones_mapeadas_si if marca not in palabras_excluidas_si and marca != ""]

# Contar la frecuencia de cada marca
frecuencias_si = Counter(opciones_finales_si)

# Ajustamos frecuencias manualmente para el grupo SÍ
ajustes_manuales_si = {
    "loewe": 1,
    "dior": 1,
    "givenchy": 1,
    "louis vuitton": 1,
    "chanel": 2
}

for marca, ajuste in ajustes_manuales_si.items():
    if marca in frecuencias_si:
        frecuencias_si[marca] += ajuste
    else:
        frecuencias_si[marca] = ajuste  # Si la marca no estaba, se añade con el ajuste

# Convertir a DataFrame para visualización
frecuencias_df_si = pd.DataFrame(frecuencias_si.items(), columns=["Marca", "Frecuencia"]).sort_values(by="Frecuencia", ascending=False)

# Mostrar resultados
print("Frecuencias de las marcas (Grupo SÍ, corregidas y ajustadas manualmente):")
print(frecuencias_df_si)
```

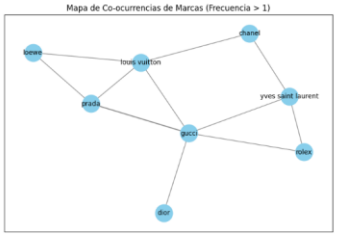
```python
# Graficar los resultados
if not frecuencias_df_si.empty:
    plt.figure(figsize=(12, 6))
    plt.bar(frecuencias_df_si["Marca"], frecuencias_df_si["Frecuencia"])
    plt.xticks(rotation=45, ha="right")
    plt.title("Frecuencia de Marcas de Lujo Mencionadas - Grupo SÍ (Corregidas y Ajustadas)")
    plt.xlabel("Marca")
    plt.ylabel("Frecuencia")
    plt.tight_layout()
    plt.show()
else:
    print("No se encontraron marcas válidas para generar el gráfico.")
```

```
Frecuencias de las marcas (Grupo SÍ, corregidas y ajustadas manualmente):
                         Marca  Frecuencia
3                louis vuitton           7
4                        gucci           7
1                        rolex           6
17                       loewe           6
0                       chanel           5
18                       prada           4
20                      cartier           3
5                         dior           3
2           yves saint laurent           3
8                  loro piana           2
30                rolls royce           1
29                    ferrari           1
21              dolce & gabanna           1
27                louis voitton           1
26                      hermes           1
25                ralph lauren           1
24                      celine           1
23              maison margiela           1
31         dior chanel givenchy           1
22  loewe louis vuitton chanel           1
28                four seasons           1
16              bottega veneta           1
19                      porsche           1
15               isabel marant           1
14           vivienne westwood           1
13                     maxmara           1
12                        etro           1
11                      armani           1
10                       zegna           1
9           brunello cucinelli           1
7               stellamcartney           1
6                   balenciaga           1
32                    givenchy           1
```




```
from itertools import combinations
from collections import Counter
import networkx as nx

# Generar lista de respuestas divididas y mapear marcas
opciones_coocurrencias_si = [
    [mapear_marca_si(marca.strip()) for marca in respuesta.split(",") if mapear_marca_si(marca.strip())]
    for respuesta in respuestas_filtradas_si
]

# Filtrar respuestas vacías o de una sola marca (porque no generan coocurrencias)
opciones_coocurrencias_si = [resp for resp in opciones_coocurrencias_si if len(resp) > 1]

# Verificar si se han filtrado demasiadas respuestas
print("Respuestas procesadas para co-ocurrencias (Grupo SÍ):", opciones_coocurrencias_si[:10])

# Generar combinaciones de co-ocurrencias
coocurrencias_si = Counter(
    combo for respuesta in opciones_coocurrencias_si for combo in combinations(sorted(set(respuesta)), 2)
)

# Verificar si se han generado co-ocurrencias
print("Co-ocurrencias sin filtrar:", coocurrencias_si.most_common(10))

# Convertir a DataFrame
coocurrencias_df_si = pd.DataFrame(coocurrencias_si.items(), columns=["Par", "Frecuencia"]).sort_values(by="Frecuencia", ascending=False)

# Print de las Coocurrencias
print("Coocurrencias por orden de frecuencias:")
print(coocurrencias_df_si)

# Visualizar co-ocurrencias como red
G = nx.Graph()
for (op1, op2), freq in coocurrencias_si.items():
    if freq > 1:  # Filtrar para frecuencias mayores a 1
        G.add_edge(op1, op2, weight=freq)

# Configuración del gráfico de red
plt.figure(figsize=(9, 6))
pos = nx.spring_layout(G, k=0.5)  # Diseño de la red
nx.draw_networkx_nodes(G, pos, node_size=700, node_color="skyblue")
nx.draw_networkx_edges(G, pos, width=[G[u][v]['weight'] / 2 for u, v in G.edges()], edge_color="gray")
nx.draw_networkx_labels(G, pos, font_size=10, font_color="black")
plt.title("Mapa de Co-ocurrencias de Marcas (Frecuencia > 1)")
plt.show()
```

```
Respuestas procesadas para co-ocurrencias (Grupo SÍ): [['chanel', 'rolex', 'yves saint laurent', 'louis vuitton'], ['gucci', 'dior', 'balenciaga', 'chanel', 'stellamcartney', 'loro piana', 'brunello cucinelli', 'zegna', 'yves saint laurent', 'armani', 'etro', 'maxmara', 'vivienne westwood', 'isabel marant'], ['r
Co-ocurrencias sin filtrar: [((('gucci', 'prada'), 3), (('chanel', 'louis vuitton'), 2), (('chanel', 'yves saint laurent'), 2), (('rolex', 'yves saint laurent'), 2), (('dior', 'gucci'), 2), (('gucci', 'yves saint laurent'), 2), (('loewe', 'prada'), 2), (('gucci', 'rolex'), 2), (('gucci', 'louis vuitton'), 2), ((
Coocurrencias por orden de frecuencias:
                                  Par  Frecuencia
100                    (gucci, prada)           3
0                (chanel, louis vuitton)          2
2              (chanel, yves saint laurent)       2
109                (louis vuitton, prada)         2
108                 (gucci, louis vuitton)        2
..                                    ...        ...
39   (brunello cucinelli, vivienne westwood)        1
38      (brunello cucinelli, stellamcartney)        1
37             (brunello cucinelli, maxmara)        1
36          (brunello cucinelli, loro piana)        1
129                    (chanel, loewe)             1

[130 rows x 2 columns]
```



2. ¿Qué significa para ti la palabra lujo?

```
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Qué significa para ti la palabra lujo?.1"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_si.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_si[columna_respuestas] = grupo_si[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta2_si = "\n".join(grupo_si[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta2:")
    print(pregunta2_si)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta2:
Elegancia accesible para pocas personas
Es poder tener algo innecesario que facilita la vida o la hace más disfrutable
Excepcional calidad en la materia prima y la elaboración ( manufactura, fabricación)
Darte un homenaje
producto o servicio de gran valor, exclusivo o de alta calidad
Caro
Exclusividad y experiencia
Exclusividad
Alta calidad
Lujo es exclusividad, diferenciación y estatus.
garantizacion de buena calidad
Exclusividad y calidad
Premium, caro, calidad, exclusividad
Lo mejor de algo
Excelencia
Exclusividad calidad que dura en el tiempo
Estatus
```

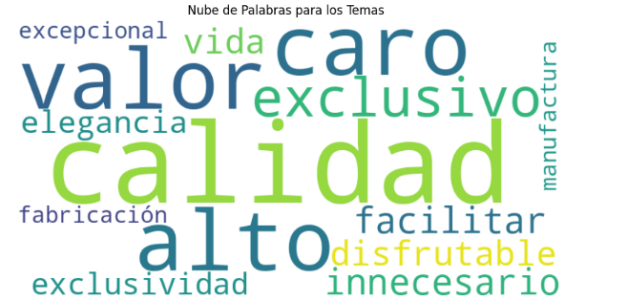
```
pregunta2_si
```

```
'Elegancia accesible para pocas personas\nEs poder tener algo innecesario que facilita la vida o la hace más disfrutable\nExcepcional calidad en la materia prima y la elaboración ( manufactura, fabricación)\nDarte un homenaje \nproducto o servicio de gran valor, exclusivo o de alta calidad \nCaro\nExclusividad y experiencia \nExclusividad\nAlta calidad\nLujo es exclusividad, diferenciación y estatus.\ngarantizacion de buena calidad\nExclusividad y calidad\nPremium, caro, calidad, exclusividad\nLo mejor de algo\nExcelencia\nExclusividad calidad que dura en el tiempo \nEstatus '
```

```
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens


# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
```

```python
# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta2
if 'pregunta2' in locals():  # Verificar que pregunta2 existe
    # Limpiar las respuestas de pregunta2
    cleaned_responses = clean_responses(pregunta2_si)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta2' no está definido. Asegúrate de haberlo creado previamente.")
```

```
 Temas generados:
Tema 1: 0.140*"calidad" + 0.098*"alto" + 0.098*"caro" + 0.056*"valor" + 0.056*"exclusivo"
Tema 2: 0.063*"innecesario" + 0.063*"facilitar" + 0.063*"disfrutable" + 0.063*"vida" + 0.063*"elegancia"
Tema 3: 0.163*"exclusividad" + 0.127*"calidad" + 0.039*"excepcional" + 0.039*"fabricación" + 0.039*"manufactura"
```



3. Ahora trata de definir lujo en una única palabra

```
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "Ahora trata de definir lujo en una única palabra.1"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_si.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_si[columna_respuestas] = grupo_si[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta3_si = "\n".join(grupo_si[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta3:")
    print(pregunta3_si)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta3:
Distinción
Placer
Calidad
Placer
valor
Caro
Experiencia
dinero
Exclusividad
Caro
Estatus
Exclusividad
Único
Caro
Calidad
Excelencia
Exclusividad
Estatus
```

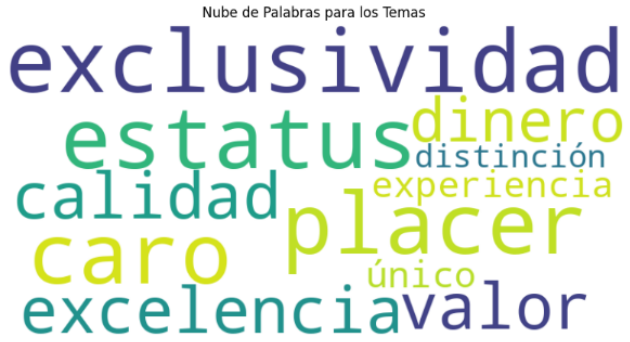
```
[ ] pregunta3_si
```

```
'Distinción\nPlacer\nCalidad\nPlacer\nvalor\nCaro\nExperiencia\ndinero\nExclusividad\nCaro\nEstatus\nExclusividad\nÚnico \nCaro\nCalidad\nExcelencia\nExclusividad \nEstatus'
```

```
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens
```

```python
# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta3
if 'pregunta3' in locals():  # Verificar que pregunta3 existe
    # Limpiar las respuestas de pregunta8
    cleaned_responses = clean_responses(pregunta3_si)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta3' no está definido. Asegúrate de haberlo creado previamente.")
```

```
 Temas generados:
Tema 1: 0.312*"exclusividad" + 0.218*"placer" + 0.124*"excelencia" + 0.124*"valor" + 0.032*"caro"
Tema 2: 0.241*"calidad" + 0.240*"estatus" + 0.137*"dinero" + 0.137*"experiencia" + 0.035*"exclusividad"
Tema 3: 0.383*"caro" + 0.153*"único" + 0.153*"distinción" + 0.039*"estatus" + 0.039*"placer"
```



4. ¿Con qué asocias tú el lujo?

```python
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Con qué asocias tú el lujo?.1"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_si.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_si[columna_respuestas] = grupo_si[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta4_si = "\n".join(grupo_si[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta4:")
    print(pregunta4_si)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta4:
Joyería
Gran calidad, precios elevados, poca accesibilidad, distinción, algo no necesario
Elegancia
Con trabajo bien hecho
dinero, elegancia, disfrute.
Dinero
Con el trato y la calidad
gente rica
Exclusividad
Con un cierto nivel económico
Con la exclusividad.
Hoteles , ropa, joyas
Capricho
Con coches de marcas caras, hoteles caros.. Lo asocio a un lifestyle por encima de la media
Dinero
Riqueza, belleza
A ropa de mi madre que sigue quedando bien a pesar de Los años
Dinero
```

```
[ ] pregunta4_si
```

```
'Joyería\nGran calidad, precios elevados, poca accesibilidad, distinción, algo no necesario\nElegancia\nCon trabajo bien hecho \ndinero, elegancia, disfrute.\nDinero\nCon el trato y la calidad\ngente rica\nExclusividad\nCon un cierto nivel económico \nCon la exclusividad.\nHoteles , ropa, joyas\nCapricho\nCon coches de marcas caras, hoteles caros.. Lo asocio a un lifestyle por encima de la media\nDinero\nRiqueza, belleza\nA ropa de mi madre que sigue quedando bien a pesar de Los años\nDinero'
```

```python
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens
```

```python
# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta4
if 'pregunta4' in locals():  # Verificar que pregunta4 existe
    # Limpiar las respuestas de pregunta4
    cleaned_responses = clean_responses(pregunta4_si)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta4' no está definido. Asegúrate de haberlo creado previamente.")
```

```
Temas generados:
Tema 1: 0.081*"calidad" + 0.080*"exclusividad" + 0.046*"elevado" + 0.046*"accesibilidad" + 0.046*"precio"
Tema 2: 0.216*"dinero" + 0.116*"elegancia" + 0.067*"disfrutir" + 0.066*"trabajo" + 0.066*"capricho"
Tema 3: 0.054*"hotel" + 0.054*"ropa" + 0.053*"coche" + 0.053*"marca" + 0.053*"cara"
```

5. ¿Qué características debe tener, en tu opinión, un artículo de lujo?

```python
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Qué características debe tener, en tu opinión, un artículo de lujo?.1"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_si.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_si[columna_respuestas] = grupo_si[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    preguntas_si = "\n".join(grupo_si[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de preguntas:")
    print(preguntas_si)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de preguntas:
Poco accesible, precio alto, calidad alta
Gran calidad, precio elevado, historia, artesanía, legado
Alta calidad , exclusividad, elegancia.
Materiales nobles, buen diseño y una utilidad/practicidad
Alta calidad, alto valor y exclusividad
Calidad
Atemporal, calidad, especial
calidad, precio alto, exclusividad
Ser unico
Que sea de máxima calidad
Debe ser o acercarse a la unicidad, ser deseado, conocido por pocos
Exclusividad, calidad, detalle
Calidad, simplicidad, exclusividad, personalidad
Exclusividad y calidad
Ser caro, bueno, escaso
Exclusivo, excelente, bello
Calidad durabilidad exclusividad limitado
Calidad
```

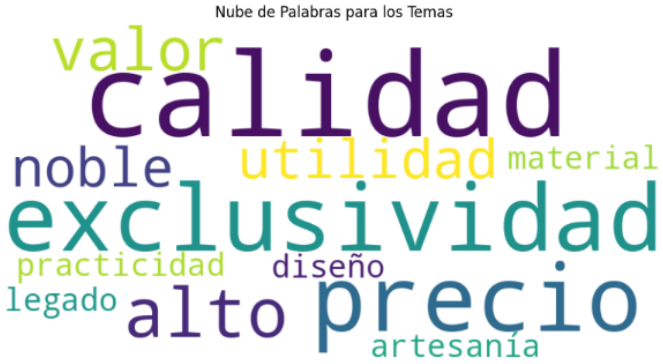
```python
preguntas_si
```

```
'Poco accesible, precio alto, calidad alta\nGran calidad, precio elevado, historia, artesanía, legado\nAlta calidad , exclusividad, elegancia. \nMateriales nobles, buen diseño y una utilidad/practicidad \nalta calidad, alto valor y exclusividad\ncalidad\nAtemporal, calidad, especial\ncalidad, precio alto, exclusividad\nSer unico\nQue sea de máxima calidad\nDebe ser o acercarse a la unicidad, ser deseado, conocido por pocos\nExclusividad, calidad, detalle\nCalidad, simplicidad, exclusividad, personalidad \nExclusividad y calidad\nSer caro, bueno, escaso\nExclusivo, excelente, bello\nCalidad durabilidad exclusividad limitado\nCalidad'
```

```python
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens
```

```python
# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta5
if 'pregunta5' in locals():  # Verificar que pregunta5 existe
    # Limpiar las respuestas de pregunta5
    cleaned_responses = clean_responses(pregunta5_si)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta5' no está definido. Asegúrate de haberlo creado previamente.")
```

```
 Temas generados:
Tema 1: 0.187*"calidad" + 0.177*"alto" + 0.153*"exclusividad" + 0.065*"precio" + 0.038*"valor"
Tema 2: 0.058*"utilidad" + 0.058*"noble" + 0.058*"practicidad" + 0.058*"diseño" + 0.058*"material"
Tema 3: 0.188*"calidad" + 0.064*"exclusividad" + 0.039*"precio" + 0.038*"artesanía" + 0.038*"legado"
```



6. ¿Cuáles de estos factores consideras que están relacionados con el lujo?

```python
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Cuáles de estos factores consideras que están relacionados con el lujo?.1"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_si.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_si[columna_respuestas] = grupo_si[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta6_si = "\n".join(grupo_si[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta6:")
    print(pregunta6_si)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

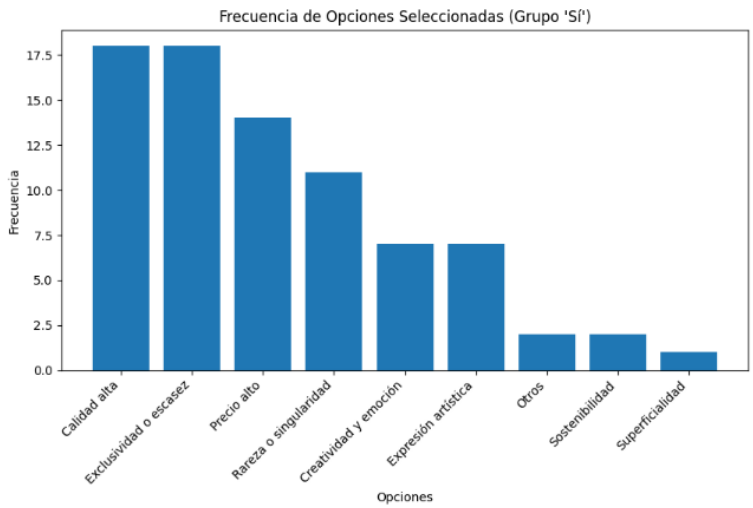
```
Contenido procesado de pregunta6:
Precio alto;Calidad alta;Exclusividad o escasez;Creatividad y emoción;Expresión artística;Otros
Precio alto;Calidad alta;Exclusividad o escasez;Sostenibilidad;Rareza o singularidad;Creatividad y emoción;Expresión artística
Precio alto;Calidad alta;Exclusividad o escasez
Precio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad;Creatividad y emoción;Expresión artística;Otros
Calidad alta;Exclusividad o escasez;Rareza o singularidad
Precio alto;Calidad alta;Exclusividad o escasez
Calidad alta;Exclusividad o escasez;Rareza o singularidad
Precio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad
Precio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad;Creatividad y emoción;Expresión artística
Calidad alta;Exclusividad o escasez;Sostenibilidad;Rareza o singularidad;Creatividad y emoción
Precio alto;Calidad alta;Exclusividad o escasez;Creatividad y emoción;Expresión artística
Calidad alta;Exclusividad o escasez;Rareza o singularidad;Expresión artística;Superficialidad
Precio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad
Precio alto;Calidad alta;Exclusividad o escasez
Precio alto;Calidad alta;Exclusividad o escasez
Precio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad;Creatividad y emoción;Expresión artística
Precio alto;Calidad alta;Exclusividad o escasez
```

```
[ ] pregunta6_si
```

```
'Precio alto;Calidad alta;Exclusividad o escasez;Creatividad y emoción;Expresión artística;Otros\nPrecio alto;Calidad alta;Exclusividad o escasez;Sostenibilidad;Rareza o singularidad;Creatividad y emoción;Expresión artística\nPrecio alto;Calidad alta;Exclusividad o escasez\nPrecio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad;Creatividad y emoción;Expresión artística;Otros\nCalidad alta;Exclusividad o escasez;Rareza o singularidad\nPrecio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad\nPrecio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad;Creatividad y emoción;Expresión artística\nCalidad alta;Exclusividad o escasez;Sostenibilidad;Rareza o singularidad;Creatividad y emoción\nPrecio alto;Calidad alta;Exclusividad o escasez;Creatividad y emoción;Expresión artística\nCalidad alta;Exclusividad o escasez;Rareza o singularidad;Expresión artística;Superficialidad\nPrecio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad\nPrecio alto;Calidad alta;Exclusividad o escasez\nPrecio alto;Calidad alta;Exclusividad o escasez\nPrecio alto;Calidad alta;Exclusividad o escasez;Rareza o singularidad;Creatividad y emoción;Expresión artística\nPrecio alto;Calidad alta;Exclusividad o escasez'
```

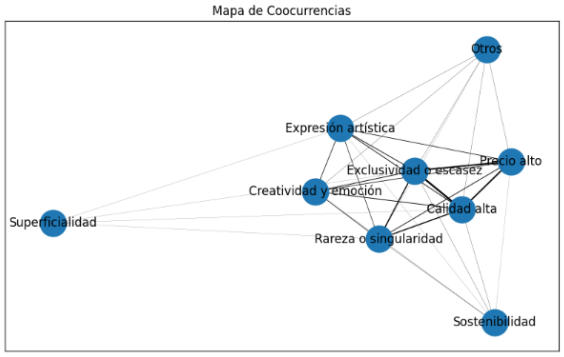
```python
from collections import Counter
import pandas as pd
import matplotlib.pyplot as plt

# Verificar el contenido de la columna
print("Contenido de la columna en el grupo 'Sí':")
print(grupo_si[columna_respuestas].head(10))

# Dividir las respuestas en opciones individuales
opciones = [respuesta.split(";") for respuesta in grupo_si[columna_respuestas].dropna()]
print("Opciones divididas:")
print(opciones[:10])

# Contar la frecuencia de cada opción
frecuencias = Counter(opcion.strip() for respuesta in opciones for opcion in respuesta)

# Mostrar las frecuencias
print("Frecuencias de las opciones (Grupo 'Sí'):")
for opcion, frecuencia in frecuencias.items():
    print(f"{opcion}: {frecuencia}")

# Convertir a DataFrame para visualización
frecuencias_df = pd.DataFrame(frecuencias.items(), columns=["Opción", "Frecuencia"]).sort_values(by="Frecuencia", ascending=False)

# Gráfico de barras
if not frecuencias_df.empty:
    plt.figure(figsize=(10, 5))
    plt.bar(frecuencias_df["Opción"], frecuencias_df["Frecuencia"])
    plt.xticks(rotation=45, ha="right")
    plt.title("Frecuencia de Opciones Seleccionadas (Grupo 'Sí')")
    plt.xlabel("Opciones")
    plt.ylabel("Frecuencia")
    plt.show()
else:
    print("No se encontraron opciones válidas para generar el gráfico.")
```

```
Contenido de la columna en el grupo 'Sí':
0    Precio alto;Calidad alta;Exclusividad o escase...
1    Precio alto;Calidad alta;Exclusividad o escase...
2      Precio alto;Calidad alta;Exclusividad o escasez
3    Precio alto;Calidad alta;Exclusividad o escase...
4    Calidad alta;Exclusividad o escasez;Rareza o s...
5      Precio alto;Calidad alta;Exclusividad o escasez
6    Calidad alta;Exclusividad o escasez;Rareza o s...
7    Precio alto;Calidad alta;Exclusividad o escase...
8    Precio alto;Calidad alta;Exclusividad o escase...
9    Precio alto;Calidad alta;Exclusividad o escase...
Name: ¿Cuáles de estos factores consideras que están relacionados con el lujo?.1, dtype: object
Opciones divididas:
[['Precio alto', 'Calidad alta', 'Exclusividad o escasez', 'Creatividad y emoción', 'Expresión artística', 'Otros'], ['Precio alto', 'Calidad alta', 'Exclusividad o escasez', 'Sostenibilidad', 'Rareza o singularidad', 'Creatividad y emoción', 'Expresión artística'], ['Precio alto', 'Calidad alta', 'Exclusividad
Frecuencias de las opciones (Grupo 'Sí'):
Precio alto: 14
Calidad alta: 18
Exclusividad o escasez: 18
Creatividad y emoción: 7
Expresión artística: 7
Otros: 2
Sostenibilidad: 2
Rareza o singularidad: 11
Superficialidad: 1
```



```python
from itertools import combinations
from collections import Counter

# Crear lista de respuestas divididas
opciones = [respuesta.split(";") for respuesta in df[columna_respuestas].dropna()]

# Generar combinaciones de coocurrencia
coocurrencias = Counter(
    combo for respuesta in opciones for combo in combinations(sorted(opcion.strip() for opcion in respuesta), 2)
)

# Convertir a DataFrame
coocurrencias_df = pd.DataFrame(coocurrencias.items(), columns=["Par", "Frecuencia"]).sort_values(by="Frecuencia", ascending=False)

# Mostrar las coocurrencias más frecuentes
print("Coocurrencias más frecuentes:")
print(coocurrencias_df.head())

# Visualizar coocurrencias (opcional)
import networkx as nx
import matplotlib.pyplot as plt

G = nx.Graph()
for (op1, op2), freq in coocurrencias.items():
    G.add_edge(op1, op2, weight=freq)

plt.figure(figsize=(10, 6))
pos = nx.spring_layout(G, k=0.5)
nx.draw_networkx_nodes(G, pos, node_size=700)
nx.draw_networkx_edges(G, pos, width=[G[u][v]['weight'] / 10 for u, v in G.edges()])
nx.draw_networkx_labels(G, pos)
plt.title("Mapa de Coocurrencias")
plt.show()
```

```
Coocurrencias más frecuentes:
                                                Par  Frecuencia
1            (Calidad alta, Exclusividad o escasez)          18
4                      (Calidad alta, Precio alto)          14
11            (Exclusividad o escasez, Precio alto)          14
15            (Calidad alta, Rareza o singularidad)          11
19  (Exclusividad o escasez, Rareza o singularidad)          11
```



7. ¿Qué sentimientos te genera el concepto de lujo?

```python
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Qué sentimientos te genera el concepto de lujo?.1"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_si.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_si[columna_respuestas] = grupo_si[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta7_si = "\n".join(grupo_si[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta7:")
    print(pregunta7_si)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta7:
Admiración, ambición
Ambición, relajación
Placer
Admiración
asombro
Pereza
Satisfacción y realización
derroche de dinero
Alegria
Apatía
Deseo y ambición
Emoción
En la actualidad, superficialidad
Ambicion
Disfrute, placer
Disfrute
Confianza, poder, materialismo, fachada
```

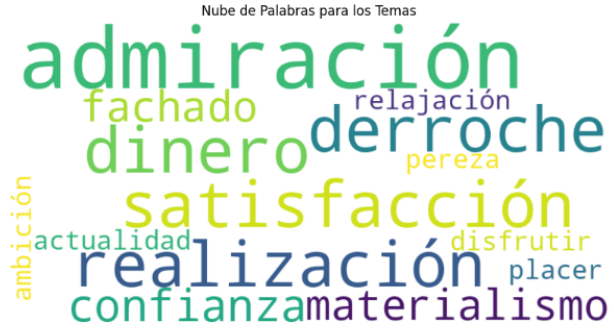
```
[ ] pregunta7_si
```

```
'Admiración, ambición\nAmbición, relajación\nPlacer\nAdmiración  \nasombro \nPereza\nSatisfacción y realización \nderroche de dinero\nAlegria\nApatía\nDeseo y ambición\nEmoción \nEn la actualidad, superficialidad\nAmbicion\nDisfrute, placer\nDisfrute\nConfianza, poder, materialismo, fachada'
```

```python
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens
```

```python
# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta7
if 'pregunta7' in locals():  # Verificar que pregunta7 existe
    # Limpiar las respuestas de pregunta7
    cleaned_responses = clean_responses(pregunta7_si)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta7' no está definido. Asegúrate de haberlo creado previamente.")
```

```
 Temas generados:
Tema 1: 0.100*"dinero" + 0.100*"satisfacción" + 0.100*"realización" + 0.100*"derroche" + 0.100*"admiración"
Tema 2: 0.087*"confianza" + 0.087*"materialismo" + 0.087*"fachado" + 0.087*"relajación" + 0.087*"pereza"
Tema 3: 0.123*"ambición" + 0.120*"placer" + 0.069*"admiración" + 0.069*"disfrutir" + 0.069*"actualidad"
```



8. ¿Qué sentimientos te genera consumir artículos de lujo?

```python
# Seleccionar la columna con las respuestas de la pregunta que queremos
columna_respuestas = "¿Qué sentimientos te genera consumir artículos de lujo?"  # Nombre exacto de la columna, la pregunta que queremos
if columna_respuestas in grupo_si.columns:
    # Eliminar saltos de línea dentro de las respuestas
    grupo_si[columna_respuestas] = grupo_si[columna_respuestas].replace(r"\n", " ", regex=True)

    # Combinar todas las respuestas en un único string con una respuesta por línea
    pregunta8_si = "\n".join(grupo_si[columna_respuestas].dropna().tolist())

    # Verificar el contenido
    print("Contenido procesado de pregunta8:")
    print(pregunta8_si)
else:
    print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta8:
Alegría, satisfacción personal por conseguir adquirirlos
Satisfacción, relajación, felicidad
Placer
Alegría
emocion
Incapacidad
Alegría
superioridad
Alegria
Culpa
En ocasiones engaño por el precio
Ilusion
Felicidad
Disfrute, placer, culpa
Disfrute, placer, alegria
Hace que ese momento sea más especial, coqueto
Confort, Falsa autoestima
```

```
[ ] pregunta8_si
```

```
'Alegría, satisfacción personal por conseguir adquirirlos\nSatisfacción, relajación, felicidad\nPlacer\nAlegría\nemocion\nIncapacidad\nAlegría \nsuperioridad\nAlegria\nCulpa\nEn ocasiones engaño por el precio\nIlusion \nFelicidad \nDisfrute, placer, culpa\nDisfrute, placer, alegria\nHace que ese momento sea más especial, coqueto\nConfort, Falsa autoestima'
```

```python
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens
```

```python
# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta8
if 'pregunta8' in locals():  # Verificar que pregunta8 existe
    # Limpiar las respuestas de pregunta8
    cleaned_responses = clean_responses(pregunta8_si)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta8' no está definido. Asegúrate de haberlo creado previamente.")
```

```
Temas generados:
Tema 1: 0.161*"placer" + 0.113*"disfrutir" + 0.113*"culpa" + 0.113*"alegrio" + 0.113*"felicidad"
Tema 2: 0.178*"alegría" + 0.072*"satisfacción" + 0.071*"adquirirlo" + 0.071*"personal" + 0.071*"precio"
Tema 3: 0.085*"autoestima" + 0.085*"falso" + 0.085*"coqueto" + 0.085*"confort" + 0.085*"especial"
```

9. ¿Cuál/es son las razones para que consumas artículos de lujo?

```
[ ] # Seleccionar la columna con las respuestas de la pregunta que queremos
    columna_respuestas = "¿Cuál/es son las razones para que consumas artículos de lujo?"  # Nombre exacto de la columna, la pregunta que queremos
    if columna_respuestas in grupo_si.columns:
        # Eliminar saltos de línea dentro de las respuestas
        grupo_si[columna_respuestas] = grupo_si[columna_respuestas].replace(r"\n", " ", regex=True)

        # Combinar todas las respuestas en un único string con una respuesta por línea
        pregunta9 = "\n".join(grupo_si[columna_respuestas].dropna().tolist())

        # Verificar el contenido
        print("Contenido procesado de pregunta9:")
        print(pregunta9)
    else:
        print(f"La columna '{columna_respuestas}' no se encontró en el archivo.")
```

```
Contenido procesado de pregunta9:
más calidad y durabilidad, poderlo pasar de generación en generación, disfrute...
Para tener artículos de calidad que cuenten una historia, cosas que sean únicas o casi únicas que hablen del legado de una marca y que me ayuden a construir mi identidad personal y visual
Me gusta disfrutar de la calidad excepcional
Autoafirmación y recompensarse a sí mismo
buscar disfrutar de una experiencia única o de un producto especial.
Dinero
Calidad y atemporabilidad
moda, calidad
Sentirme bien ye xlcusiva
Interés en el producto
Por estatus y diferenciación
Porque me gustan estrictamente o pq busco un servicio determinado
Calidad, comididad, estilo
Para disfrutarlos durante mas tiempo y "diferenciarme"
El placer de disfrutar de lo mejor
Disfrute
Si dura y es de la calidad que quiero merece el precio
Que me lo regalen o que encuentre una buena oferta de segunda mano
```

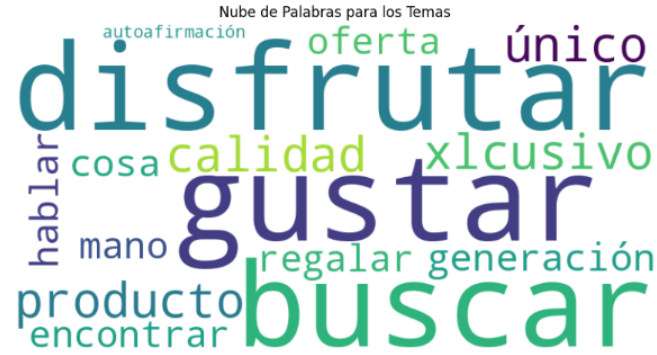
```
pregunta9
```

```
'más calidad y durabilidad, poderlo pasar de generación en generación, disfrute...\nPara tener artículos de calidad que cuenten una historia, cosas que sean únicas o casi únicas que hablen del legado de una marca y que me ayuden a construir mi identidad personal y visual\nMe gusta disfrutar de la calidad excepciona
l\nAutoafirmación y recompensarse a sí mismo  \nbuscar disfrutar de una experiencia única o de un producto especial.\nDinero\ncalidad y atemporabilidad \nmoda, calidad\nsentirme bien ye xlcusiva\ninterés en el producto \nPor estatus y diferenciación \nPorque me gustan estrictamente o pq busco un servicio determinad
o \ncalidad, comididad, estilo\nPara disfrutarlos durante mas tiempo y "diferenciarme" \nel placer de disfrutar de lo mejor\ndisfrute\nsi dura y es de la calidad que quiero merece el precio\nque me lo regalen o que encuentre una buena oferta de segunda mano'
```

```python
import spacy
from gensim import corpora
from gensim.models import LdaModel
from spacy.matcher import PhraseMatcher
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Cargar modelo de lenguaje en español para preprocesamiento
nlp = spacy.load("es_core_news_sm")

# Funcion para que el preprocesamiento del texto mantenga las "frases" de marcas multipalabra como Louis Vuitton o Miu Miu
# Crear un PhraseMatcher para identificar combinaciones de palabras
phrase_matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
marcas_multipalabra = ["miu miu", "louis vuitton", "dolce gabbana", "armani prive"]
patterns = [nlp.make_doc(phrase) for phrase in marcas_multipalabra]
phrase_matcher.add("MARCAS", patterns)

# Función para preprocesar texto y mantener frases multi-palabra y nombres propios
def preprocess_text_with_phrases(text):
    doc = nlp(text.lower())
    matches = phrase_matcher(doc)

    # Convertir combinaciones detectadas en un único token
    spans = [doc[start:end] for _, start, end in matches]
    spans = sorted(spans, key=lambda span: span.start)

    tokens = []
    prev_end = 0
    for span in spans:
        # Agregar tokens previos que no están en combinaciones
        tokens.extend([
            token.text if token.pos_ == "PROPN" else token.lemma_
            for token in doc[prev_end:span.start]
            if token.is_alpha and not token.is_stop
        ])
        # Agregar la combinación como un único token
        tokens.append("_".join(span.text.split()))
        prev_end = span.end

    # Agregar los tokens restantes
    tokens.extend([
        token.text if token.pos_ == "PROPN" else token.lemma_
        for token in doc[prev_end:]
        if token.is_alpha and not token.is_stop
    ])

    return tokens
```

```python
# Limpiar saltos de línea dentro de las respuestas en pregunta1
def clean_responses(responses):
    # Eliminar saltos de línea internos y devolver cada respuesta como un elemento de la lista
    return [line.replace("\n", " ").strip() for line in responses.split("\n") if line.strip()]

# Aplicar LDA a las respuestas procesadas
def apply_lda_to_responses(responses, num_topics=2, num_words=5):
    # Preprocesar cada respuesta
    processed_texts = [preprocess_text_with_phrases(response) for response in responses]

    # Crear diccionario y corpus
    dictionary = corpora.Dictionary(processed_texts)
    corpus = [dictionary.doc2bow(text) for text in processed_texts]

    # Aplicar modelo LDA
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=10)

    # Obtener temas
    topics = lda_model.print_topics(num_topics=num_topics, num_words=num_words)

    # Imprimir los temas generados
    print("\n Temas generados:")
    for i, topic in topics:
        print(f"Tema {i + 1}: {topic}")

    # Generar nube de palabras
    all_words = []
    for _, topic_words in topics:
        words = [word.split("*")[1].replace('"', '').strip() for word in topic_words.split(" + ")]
        all_words.extend(words)

    wordcloud = WordCloud(width=800, height=400, background_color="white", colormap="viridis").generate(" ".join(all_words))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.title("Nube de Palabras para los Temas")
    plt.show()

# Procesar y analizar el contenido de pregunta9
if 'pregunta9' in locals():  # Verificar que pregunta9 existe
    # Limpiar las respuestas de pregunta9
    cleaned_responses = clean_responses(pregunta9)

    # Aplicar LDA
    apply_lda_to_responses(cleaned_responses, num_topics=3, num_words=5)
else:
    print("El objeto 'pregunta9' no está definido. Asegúrate de haberlo creado previamente.")
```

```
 Temas generados:
Tema 1: 0.103*"disfrutar" + 0.055*"gustar" + 0.055*"buscar" + 0.032*"producto" + 0.032*"xlcusivo"
Tema 2: 0.104*"calidad" + 0.044*"único" + 0.044*"generación" + 0.025*"cosa" + 0.025*"hablar"
Tema 3: 0.045*"encontrar" + 0.045*"oferta" + 0.045*"regalar" + 0.045*"mano" + 0.045*"autoafirmación"
```



17. Código del estudio confirmatorio (regresión múltiple)

ESTUDIO CONFIRMATORIO CUANTITATIVO

1. Carga de datos, importación de paquetes necesarios y procesamiento

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import classification_report, roc_auc_score, roc_curve, mean_squared_error, accuracy_score,  r2_score
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import nltk
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
```

```
#importar los datos
df=pd.read_excel('Respuestas Estudio Confirmatorio TFG BA (1).xlsx')
df.head()
```

	Anterioridad	Intencion_Compra	BC1	BC2	BC3	BC4	BC5	BC6	SC1	SC2	...	VR3	VR4	CE1	CE2	CE3	CE4	Genero	Edad	Ingresos	Ocupacion
0	Sí, alguna vez	4	7	6	5	3	4	3	3	3	...	6	6	3	4	5	4	Mujer	18 - 25	< 10.000 €	Estudiante
1	Sí, alguna vez	4	6	4	4	4	5	3	7	7	...	5	5	6	7	7	7	Mujer	18 - 25	< 10.000 €	Estudiante
2	Sí, alguna vez	7	6	7	4	4	5	2	5	6	...	5	5	6	6	7	7	Mujer	18 - 25	< 10.000 €	Estudiante
3	No, nunca	3	5	3	6	3	2	5	2	3	...	4	6	2	5	6	5	NaN	NaN	NaN	NaN
4	Sí, alguna vez	2	4	2	3	1	1	1	7	6	...	4	4	6	5	5	5	Mujer	18 - 25	< 10.000 €	Empleado/a por cuenta ajena (sector privado)

5 rows × 71 columns

```
df_cuant=df.iloc[:,1:67]
df_cuant.head()
```

	Intencion_Compra	BC1	BC2	BC3	BC4	BC5	BC6	SC1	SC2	SC3	...	Ri5	Ri6	VR1	VR2	VR3	VR4	CE1	CE2	CE3	CE4
0	4	7	6	5	3	4	3	3	3	2	...	2	7	5	3	6	6	3	4	5	4
1	4	6	4	4	4	5	3	7	7	6	...	4	6	4	3	5	5	6	7	7	7
2	7	6	7	4	4	5	2	5	6	4	...	4	5	1	1	5	5	6	6	7	7
3	3	5	3	6	3	2	5	2	3	6	...	3	3	5	5	4	6	2	5	6	5
4	2	4	2	3	1	1	1	7	6	5	...	2	2	4	4	4	4	6	5	5	5

5 rows × 66 columns

```
# comprobamos que no hay NAs
df_cuant.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 66 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Intencion_Compra  101 non-null    int64
 1   BC1               101 non-null    int64
 2   BC2               101 non-null    int64
 3   BC3               101 non-null    int64
 4   BC4               101 non-null    int64
 5   BC5               101 non-null    int64
 6   BC6               101 non-null    int64
 7   SC1               101 non-null    int64
 8   SC2               101 non-null    int64
 9   SC3               101 non-null    int64
 10  SC4               101 non-null    int64
 11  SC5               101 non-null    int64
 12  SC6               101 non-null    int64
 13  SC7               101 non-null    int64
 14  A1                101 non-null    int64
 15  A2                101 non-null    int64
 16  A3                101 non-null    int64
 17  A4                101 non-null    int64
 18  A5                101 non-null    int64
 19  SN1               101 non-null    int64
 20  SN2               101 non-null    int64
 21  SN3               101 non-null    int64
 22  SN4               101 non-null    int64
 23  PBC1              101 non-null    int64
 24  PBC2              101 non-null    int64
 25  PBC3              101 non-null    int64
 26  PBC4              101 non-null    int64
 27  PBC5              101 non-null    int64
 28  So1               101 non-null    int64
 29  So2               101 non-null    int64
 30  So3               101 non-null    int64
 31  So4               101 non-null    int64
 32  So5               101 non-null    int64
 33  So6               101 non-null    int64
 34  CM1               101 non-null    int64
 35  CM2               101 non-null    int64
 36  CM3               101 non-null    int64
 37  CM4               101 non-null    int64
 38  CM5               101 non-null    int64
 39  Si1               101 non-null    int64
 43  CP1               101 non-null    int64
 44  CP2               101 non-null    int64
 45  CP3               101 non-null    int64
 46  CP4               101 non-null    int64
 47  CP5               101 non-null    int64
 48  No1               101 non-null    int64
 49  No2               101 non-null    int64
 50  No3               101 non-null    int64
 51  No4               101 non-null    int64
 52  Ri1               101 non-null    int64
 53  Ri2               101 non-null    int64
 54  Ri3               101 non-null    int64
 55  Ri4               101 non-null    int64
 56  Ri5               101 non-null    int64
 57  Ri6               101 non-null    int64
 58  VR1               101 non-null    int64
 59  VR2               101 non-null    int64
 60  VR3               101 non-null    int64
 61  VR4               101 non-null    int64
 62  CE1               101 non-null    int64
 63  CE2               101 non-null    int64
 64  CE3               101 non-null    int64
 65  CE4               101 non-null    int64
dtypes: int64(66)
memory usage: 52.2 KB
```

```
#estadísticos básicos
df_cuant.describe().round(3)
```

	Intencion_Compra	BC1	BC2	BC3	BC4	BC5	BC6	SC1	SC2	SC3	...	Ri5	Ri6	VR1	VR2	VR3	VR4	CE1	CE2	CE3	CE4
count	101.000	101.000	101.000	101.000	101.000	101.000	101.000	101.000	101.000	101.000	...	101.000	101.000	101.000	101.000	101.000	101.000	101.000	101.000	101.000	101.000
mean	3.960	4.297	3.624	3.406	2.614	3.149	2.505	4.921	4.624	3.485	...	2.871	4.267	3.366	3.000	4.238	3.941	3.287	3.624	4.010	4.079
std	2.092	1.978	1.918	1.668	1.555	1.819	1.724	1.963	1.974	1.809	...	1.842	1.949	2.004	1.918	1.744	1.912	2.090	2.253	2.269	2.270
min	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	...	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
25%	2.000	3.000	2.000	2.000	1.000	1.000	1.000	3.000	3.000	2.000	...	1.000	3.000	1.000	1.000	3.000	2.000	1.000	1.000	2.000	2.000
50%	4.000	4.000	3.000	3.000	2.000	3.000	2.000	5.000	5.000	3.000	...	2.000	4.000	3.000	3.000	4.000	4.000	3.000	4.000	4.000	5.000
75%	6.000	6.000	5.000	5.000	4.000	5.000	3.000	7.000	6.000	5.000	...	4.000	6.000	5.000	4.000	6.000	6.000	5.000	6.000	6.000	6.000
max	7.000	7.000	7.000	7.000	6.000	7.000	7.000	7.000	7.000	7.000	...	7.000	7.000	7.000	7.000	7.000	7.000	7.000	7.000	7.000	7.000

8 rows × 66 columns

Medición del alpha de Cronbach

```python
def cronbach_alpha(df):
    items = df.values
    num_items = df.shape[1]  # Número de ítems

    # Calcular la varianza de cada ítem
    item_variances = np.var(items, axis=0, ddof=1)

    # Calcular la varianza total del test
    total_variance = np.var(np.sum(items, axis=1), ddof=1)

    # Aplicar la fórmula del alpha de Cronbach
    alpha = (num_items / (num_items - 1)) * (1 - (np.sum(item_variances) / total_variance))
    return alpha

# seleccionamos las columnas relevantes
subset_df_SC = df[['SC1', 'SC2', 'SC3', 'SC4', 'SC5', 'SC6']] #Snob Consumption
subset_df_BC = df[['BC1', 'BC2', 'BC3', 'BC4', 'BC5', 'BC6']] #Bandwagon Consumption
subset_df_SN = df[['SN1', 'SN2', 'SN3', 'SN4']] #Subjective Norms
subset_df_A = df[['A1', 'A2', 'A3', 'A4', 'A5']] #Attitudes
subset_df_PBC = df[['PBC1', 'PBC2', 'PBC3', 'PBC4', 'PBC5']] #PBC
subset_df_CM = df[['CM1', 'CM2', 'CM3', 'CM4', 'CM5']] #Conocimiento de marca
subset_df_So = df[['So1', 'So2', 'So3', 'So4', 'So5', 'So6']] #Sostenibilidad
subset_df_VR = df[['VR1', 'VR2', 'VR3', 'VR4']] #Valor de reventa
subset_df_Sing = df[['Si1', 'Si2', 'Si3', 'Si4']] #Singularidad
subset_df_CE = df[['CE1', 'CE2', 'CE3', 'CE4']] #Creatividad y emoción
subset_df_Nost = df[['No1', 'No2', 'No3', 'No4']] #Nostalgia
subset_df_CP = df[['CP1', 'CP2', 'CP3', 'CP4', 'CP5']] #Calidad precio
subset_df_Ri = df[['Ri1', 'Ri2', 'Ri3', 'Ri4', 'Ri5', 'Ri6']] #Riesgo

# calculamos la alpha de Cronbach
alpha_BC = cronbach_alpha(subset_df_BC)
alpha_SC = cronbach_alpha(subset_df_SC)
alpha_SN = cronbach_alpha(subset_df_SN)
alpha_A = cronbach_alpha(subset_df_A)
alpha_PBC = cronbach_alpha(subset_df_PBC)
alpha_CM = cronbach_alpha(subset_df_CM)
alpha_So = cronbach_alpha(subset_df_So)
alpha_VR = cronbach_alpha(subset_df_VR)
alpha_Sing = cronbach_alpha(subset_df_Sing)
alpha_CE = cronbach_alpha(subset_df_CE)
alpha_Nost = cronbach_alpha(subset_df_Nost)
alpha_CP = cronbach_alpha(subset_df_CP)
alpha_Ri = cronbach_alpha(subset_df_Ri)
```

```python
# Crear la matriz de Alphas de Cronbach con los valores calculados
alpha_values_computed = {
    "Bandwagon Consumption": alpha_BC,
    "Snob Consumption": alpha_SC,
    "Normas Subjetivas": alpha_SN,
    "Actitudes": alpha_A,
    "PBC": alpha_PBC,
    "Conocimiento de Marca": alpha_CM,
    "Sostenibilidad": alpha_So,
    "Valor de Reventa": alpha_VR,
    "Singularidad": alpha_Sing,
    "Creatividad y Emoción": alpha_CE,
    "Nostalgia": alpha_Nost,
    "Calidad-Precio": alpha_CP,
    "Riesgo": alpha_Ri,
}

# Convertir a una matriz 1x13
alpha_matrix_computed = np.array(list(alpha_values_computed.values())).reshape(1, -1)

# Convertir a un DataFrame para visualización
alpha_df_computed = pd.DataFrame(alpha_matrix_computed, columns=alpha_values_computed.keys())

# Mostrar la matriz de alphas de Cronbach
from IPython.display import display
display(alpha_df_computed)
```

	Bandwagon Consumption	Snob Consumption	Normas Subjetivas	Actitudes	PBC	Conocimiento de Marca	Sostenibilidad	Valor de Reventa	Singularidad	Creatividad y Emoción	Nostalgia	Calidad-Precio	Riesgo
0	0.837567	0.887752	0.65479	0.876893	0.787425	0.890783	0.858459	0.874534	0.883316	0.952845	0.910008	0.904301	0.855256

El alpha de Normas Subjetivas es relativamente bajo, comprobemos si eliminando alguna de las preguntas mejora. El resto de alphas tienen niveles cercanos a 1, aceptables, por lo que las preguntas empleadas son consistentes.

```python
# Calcular el Alpha de Cronbach eliminando cada ítem de Normas Subjetivas

# Lista de columnas del subset de Normas Subjetivas (SN)
sn_columns = subset_df_SN.columns

# Crear un diccionario para almacenar los valores de Alpha al eliminar cada ítem
alpha_results = {}

for col in sn_columns:
    temp_df = subset_df_SN.drop(columns=[col])  # Crear un dataframe sin la columna eliminada
    alpha_results[col] = cronbach_alpha(temp_df)  # Calcular el nuevo Alpha

# Convertir resultados a un DataFrame
alpha_sn_df = pd.DataFrame(alpha_results.items(), columns=["Ítem Eliminado", "Alpha de Cronbach"])

# Mostrar la tabla con los resultados
from IPython.display import display
display(alpha_sn_df)
```

	Ítem Eliminado	Alpha de Cronbach
0	SN1	0.562546
1	SN2	0.673425
2	SN3	0.623371
3	SN4	0.446439

El alpha mejoraría si eliminamos el artibuto SN2, correspondiente a la siguiente pregunta: "Considero que comprar lujo de segunda mano es algo aceptado positivamente en mi entorno". Sin embargo, ya que cambia muy poco, lo dejaremos como está.

CFA Loadings

Aquí montamos el modelo con las variables y lo entreno con los datos de la encuesta para verificar que las preguntas del cuestionario miden los constructos teóricos del modelo

```python
#hemos cogido la base de internet
!pip install factor_analyzer
from factor_analyzer import FactorAnalyzer
from factor_analyzer.confirmatory_factor_analyzer import ModelSpecificationParser, ConfirmatoryFactorAnalyzer

#Cogemos todas las columnas
all_columns = df_cuant.columns.tolist()

# Ajustamos el modelo
model_dict = {
    "Intecion Compra": ['Intencion_Compra'],
    "Bandwagon Consumption": ['BC1', 'BC2', 'BC3', 'BC4', 'BC5', 'BC6'],
    "Snob Consumption": ['SC1', 'SC2', 'SC3', 'SC4', 'SC5', 'SC6', 'SC7'],
    "Normas Subjetivas": ['SN1', 'SN2', 'SN3', 'SN4'],
    "Actitudes": ['A1', 'A2', 'A3', 'A4', 'A5'],
    "PBC": ['PBC1', 'PBC2', 'PBC3', 'PBC4', 'PBC5'],
    "Conocimiento de Marca": ['CM1', 'CM2', 'CM3', 'CM4', 'CM5'],
    "Sostenibilidad": ['So1', 'So2', 'So3', 'So4', 'So5', 'So6'],
    "Valor de Reventa": ['VR1', 'VR2', 'VR3', 'VR4'],
    "Singularidad": ['Si1', 'Si2', 'Si3', 'Si4'],
    "Creatividad y Emoción": ['CE1', 'CE2', 'CE3', 'CE4'],
    "Calidad-Precio": ['CP1', 'CP2', 'CP3', 'CP4', 'CP5'],
    "Nostalgia": ['No1', 'No2', 'No3', 'No4'],
    "Riesgo": ['Ri1', 'Ri2', 'Ri3', 'Ri4', 'Ri5', 'Ri6']
}

# Mostrar el diccionario generado
model_dict
```

```
Collecting factor_analyzer
  Downloading factor_analyzer-0.5.1.tar.gz (42 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.8/42.8 kB 2.0 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from factor_analyzer) (2.2.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from factor_analyzer) (1.13.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from factor_analyzer) (1.26.4)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (from factor_analyzer) (1.6.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas->factor_analyzer) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas->factor_analyzer) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas->factor_analyzer) (2025.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn->factor_analyzer) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn->factor_analyzer) (3.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas->factor_analyzer) (1.17.0)
Building wheels for collected packages: factor_analyzer
  Building wheel for factor_analyzer (pyproject.toml) ... done
  Created wheel for factor_analyzer: filename=factor_analyzer-0.5.1-py2.py3-none-any.whl size=42622 sha256=c475478d8bd8d55baa0326011a6e9ccdfd1f8f2373a05c8e315a93a17cb0c879
  Stored in directory: /root/.cache/pip/wheels/fa/f7/53/a55a8a56668a6fe0199e0e02b6e0ae3007ec35cdf6e4c25df7
Successfully built factor_analyzer
Installing collected packages: factor_analyzer
Successfully installed factor_analyzer-0.5.1
{'Intecion Compra': ['Intencion_Compra'],
 'Bandwagon Consumption': ['BC1', 'BC2', 'BC3', 'BC4', 'BC5', 'BC6'],
 'Snob Consumption': ['SC1', 'SC2', 'SC3', 'SC4', 'SC5', 'SC6', 'SC7'],
 'Normas Subjetivas': ['SN1', 'SN2', 'SN3', 'SN4'],
 'Actitudes': ['A1', 'A2', 'A3', 'A4', 'A5'],
 'PBC': ['PBC1', 'PBC2', 'PBC3', 'PBC4', 'PBC5'],
 'Conocimiento de Marca': ['CM1', 'CM2', 'CM3', 'CM4', 'CM5'],
 'Sostenibilidad': ['So1', 'So2', 'So3', 'So4', 'So5', 'So6'],
```

```
'Valor de Reventa': ['VR1', 'VR2', 'VR3', 'VR4'],
'Singularidad': ['Si1', 'Si2', 'Si3', 'Si4'],
'Creatividad y Emoción': ['CE1', 'CE2', 'CE3', 'CE4'],
'Calidad-Precio': ['CP1', 'CP2', 'CP3', 'CP4', 'CP5'],
'Nostalgia': ['No1', 'No2', 'No3', 'No4'],
'Riesgo': ['Ri1', 'Ri2', 'Ri3', 'Ri4', 'Ri5', 'Ri6']}
```

```
[ ] print(df_cuant.columns)
```

```
Index(['Intencion_Compra', 'BC1', 'BC2', 'BC3', 'BC4', 'BC5', 'BC6', 'SC1',
       'SC2', 'SC3', 'SC4', 'SC5', 'SC6', 'SC7', 'A1', 'A2', 'A3', 'A4', 'A5',
       'SN1', 'SN2', 'SN3', 'SN4', 'PBC1', 'PBC2', 'PBC3', 'PBC4', 'PBC5',
       'So1', 'So2', 'So3', 'So4', 'So5', 'So6', 'CM1', 'CM2', 'CM3', 'CM4',
       'CM5', 'Si1', 'Si2', 'Si3', 'Si4', 'CP1', 'CP2', 'CP3', 'CP4', 'CP5',
       'No1', 'No2', 'No3', 'No4', 'Ri1', 'Ri2', 'Ri3', 'Ri4', 'Ri5', 'Ri6',
       'VR1', 'VR2', 'VR3', 'VR4', 'CE1', 'CE2', 'CE3', 'CE4'],
      dtype='object')
```

```
[ ] # Comprobamos que el diccionario y el df tienen la misma dimension
    print("Dimensiones de df_cuant:", df_cuant.shape)  # (filas, columnas)

    # Contar cuántos ítems hay en total
    total_items = sum(len(v) for v in model_dict.values())
    print("Número total de ítems en model_dict:", total_items)
```

```
Dimensiones de df_cuant: (101, 66)
Número total de ítems en model_dict: 66
```

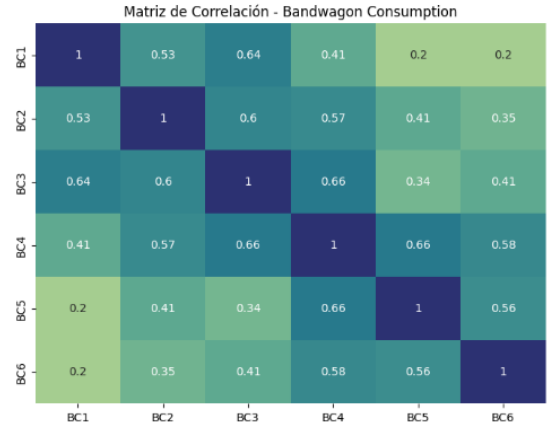
```
model_spec = ModelSpecificationParser.parse_model_specification_from_dict(df_cuant, model_dict)

cfa = ConfirmatoryFactorAnalyzer(model_spec, disp=False)

cfa.fit(df_cuant.values)

# sacamos los loadings
loadings = pd.DataFrame(cfa.loadings_, index=df_cuant.columns)

# lo imprimimos en un formato de tabla
print(loadings.to_string(float_format="{:.3f}".format))
```

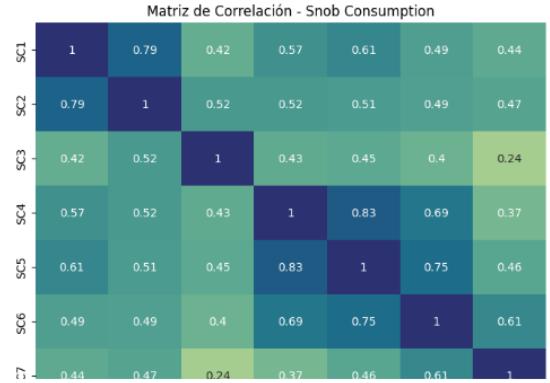
```
                  0     1     2     3     4     5     6     7     8     9    10    11    12    13
Intencion_Compra 1.377 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
BC1              0.000 1.008 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
BC2              0.000 1.137 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
BC3              0.000 1.055 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
BC4              0.000 1.046 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
BC5              0.000 0.971 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
BC6              0.000 0.895 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SC1              0.000 0.000 1.064 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SC2              0.000 0.000 1.046 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SC3              0.000 0.000 0.792 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SC4              0.000 0.000 1.162 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SC5              0.000 0.000 1.226 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SC6              0.000 0.000 1.091 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SC7              0.000 0.000 0.954 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
A1               0.000 0.000 0.000 1.175 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
A2               0.000 0.000 0.000 1.034 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
A3               0.000 0.000 0.000 0.845 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
A4               0.000 0.000 0.000 1.162 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
A5               0.000 0.000 0.000 0.000 1.119 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SN1              0.000 0.000 0.000 0.000 0.909 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SN2              0.000 0.000 0.000 0.000 0.964 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SN3              0.000 0.000 0.000 0.000 0.664 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
SN4              0.000 0.000 0.000 0.000 1.194 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
PBC1             0.000 0.000 0.000 0.000 0.000 0.979 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
PBC2             0.000 0.000 0.000 0.000 0.000 1.003 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
PBC3             0.000 0.000 0.000 0.000 0.000 1.078 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
PBC4             0.000 0.000 0.000 0.000 0.000 0.991 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
PBC5             0.000 0.000 0.000 0.000 0.000 0.975 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
So1              0.000 0.000 0.000 0.000 0.000 0.000 1.050 0.000 0.000 0.000 0.000 0.000 0.000 0.000
So2              0.000 0.000 0.000 0.000 0.000 0.000 1.088 0.000 0.000 0.000 0.000 0.000 0.000 0.000
So3              0.000 0.000 0.000 0.000 0.000 0.000 1.126 0.000 0.000 0.000 0.000 0.000 0.000 0.000
So4              0.000 0.000 0.000 0.000 0.000 0.000 1.055 0.000 0.000 0.000 0.000 0.000 0.000 0.000
So5              0.000 0.000 0.000 0.000 0.000 0.000 1.039 0.000 0.000 0.000 0.000 0.000 0.000 0.000
So6              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.581 0.000 0.000 0.000 0.000 0.000 0.000
CM1              0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.069 0.000 0.000 0.000 0.000 0.000 0.000
CM2              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.825 0.000 0.000 0.000 0.000 0.000 0.000
CM3              0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.121 0.000 0.000 0.000 0.000 0.000 0.000
CM4              0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.187 0.000 0.000 0.000 0.000 0.000 0.000
CM5              0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.240 0.000 0.000 0.000 0.000 0.000 0.000
Si1              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.091 0.000 0.000 0.000 0.000 0.000
Si2              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.212 0.000 0.000 0.000 0.000 0.000
Si3              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.068 0.000 0.000 0.000 0.000 0.000
Si4              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.050 0.000 0.000 0.000 0.000 0.000
CP1              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.073 0.000 0.000 0.000 0.000
CP2              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.022 0.000 0.000 0.000 0.000
CP3              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.101 0.000 0.000 0.000 0.000
CP4              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.951 0.000 0.000 0.000 0.000
CP5              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.820 0.000 0.000 0.000 0.000
No1              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.087 0.000 0.000 0.000 0.000
No2              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.198 0.000 0.000 0.000 0.000
No3              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.120 0.000 0.000 0.000 0.000
No4              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.683 0.000 0.000 0.000 0.000
Ri1              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.037 0.000 0.000 0.000 0.000
Ri2              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.250 0.000 0.000 0.000 0.000
Ri3              0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.202 0.000 0.000 0.000 0.000
```

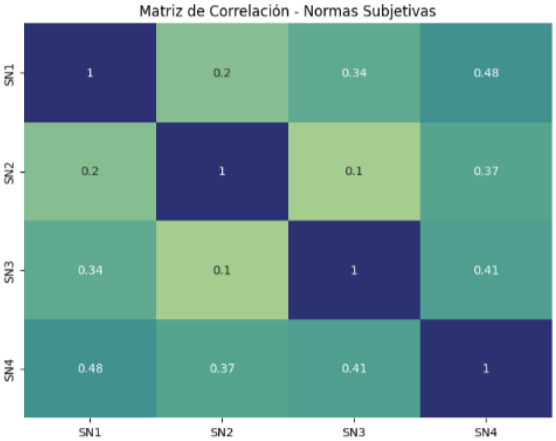
Dado que obtenemos muchos loadings superiores a 1, puede ser que haya multicolinealidad y por tanto afirmaciones redundantes a la hora de explicar un mismo factor. Por ello, haremos una matriz de correlaciones para cada factor, y así veremos las correlaciones entre las afirmaciones.

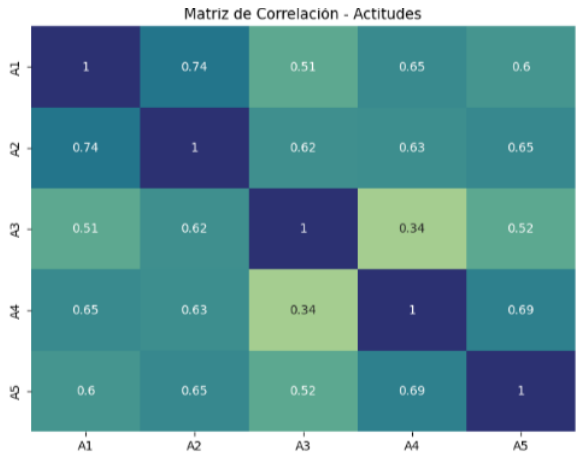
Dado que obtenemos muchos loadings superiores a 1, puede ser que haya multicolinealidad y por tanto afirmaciones redundantes a la hora de explicar un mismo factor. Por ello, haremos una matriz de correlaciones para cada factor, y así veremos las correlaciones entre las afirmaciones.

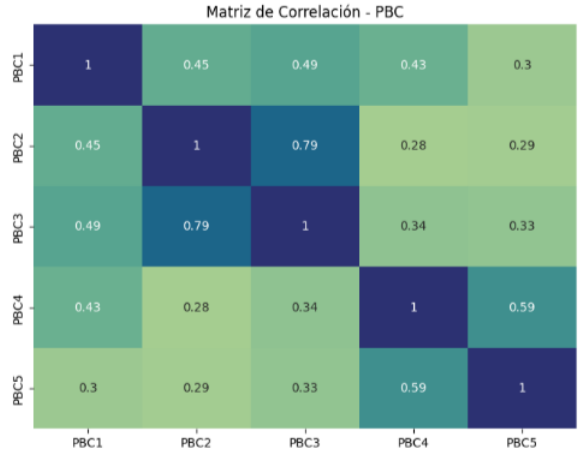
```python
# Diccionario con los constructos y sus respectivas preguntas
constructos = {
    "Bandwagon Consumption": ['BC1', 'BC2', 'BC3', 'BC4', 'BC5', 'BC6'],
    "Snob Consumption": ['SC1', 'SC2', 'SC3', 'SC4', 'SC5', 'SC6', 'SC7'],
    "Normas Subjetivas": ['SN1', 'SN2', 'SN3', 'SN4'],
    "Actitudes": ['A1', 'A2', 'A3', 'A4', 'A5'],
    "PBC": ['PBC1', 'PBC2', 'PBC3', 'PBC4', 'PBC5'],
    "Conocimiento de Marca": ['CM1', 'CM2', 'CM3', 'CM4', 'CM5'],
    "Sostenibilidad": ['So1', 'So2', 'So3', 'So4', 'So5', 'So6'],
    "Valor de Reventa": ['VR1', 'VR2', 'VR3', 'VR4'],
    "Singularidad": ['Si1', 'Si2', 'Si3', 'Si4'],
    "Creatividad y Emoción": ['CE1', 'CE2', 'CE3', 'CE4'],
    "Calidad-Precio": ['CP1', 'CP2', 'CP3', 'CP4', 'CP5'],
    "Nostalgia": ['No1', 'No2', 'No3', 'No4'],
    "Riesgo": ['Ri1', 'Ri2', 'Ri3', 'Ri4', 'Ri5', 'Ri6']
}

# Generar y visualizar la matriz de correlaciones de cada constructo
for constructo, variables in constructos.items():
    plt.figure(figsize=(8, 6))
    sns.heatmap(df_cuant[variables].corr(), annot=True, cbar=False, cmap='crest')
    plt.title(f'Matriz de Correlación - {constructo}')
    plt.show()
```

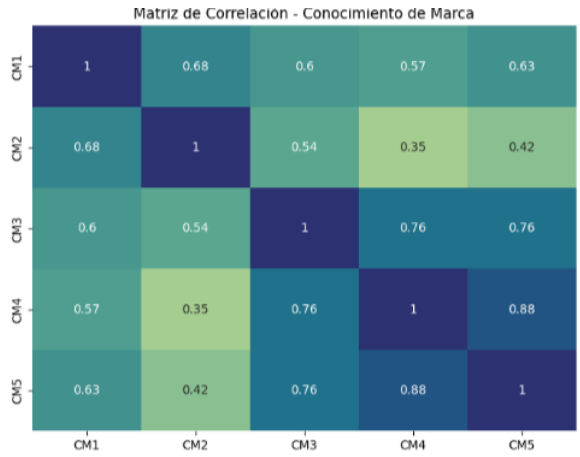


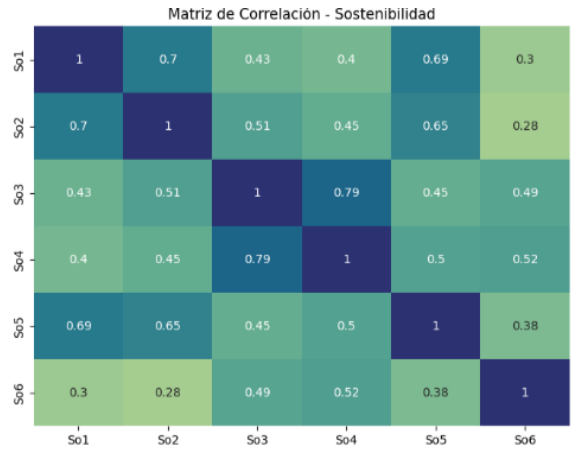


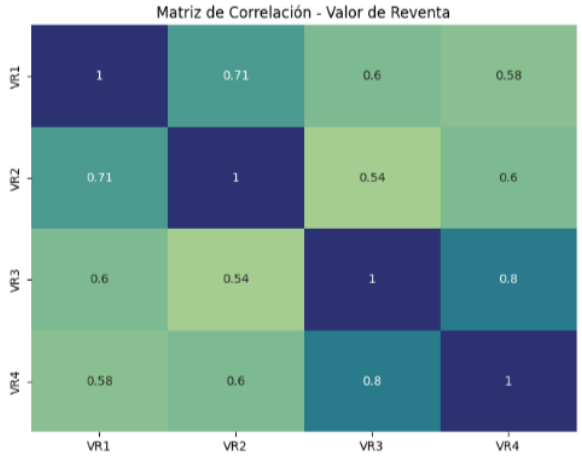


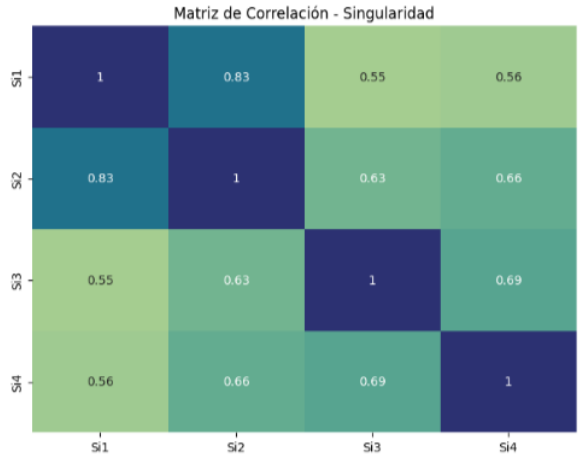

Matriz de Correlación - Actitudes
A1
A2
A3
A4
A5
1 0.74 0.51 0.65 0.6
0.74 1 0.62 0.63 0.65
0.51 0.62 1 0.34 0.52
0.65 0.63 0.34 1 0.69
0.6 0.65 0.52 0.69 1

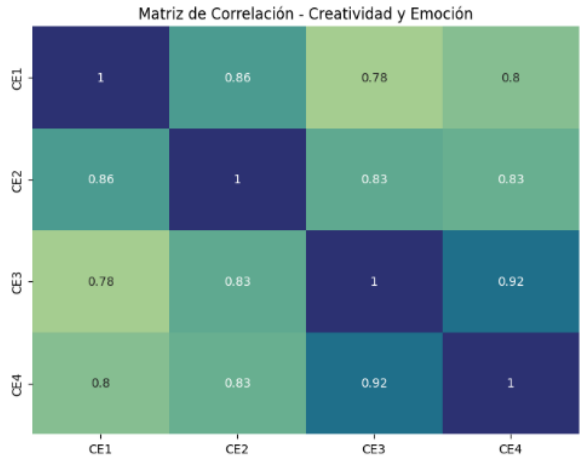

Matriz de Correlación - PBC
PBC1
PBC2
PBC3
PBC4
PBC5
1 0.45 0.49 0.43 0.3
0.45 1 0.79 0.28 0.29
0.49 0.79 1 0.34 0.33
0.43 0.28 0.34 1 0.59
0.3 0.29 0.33 0.59 1

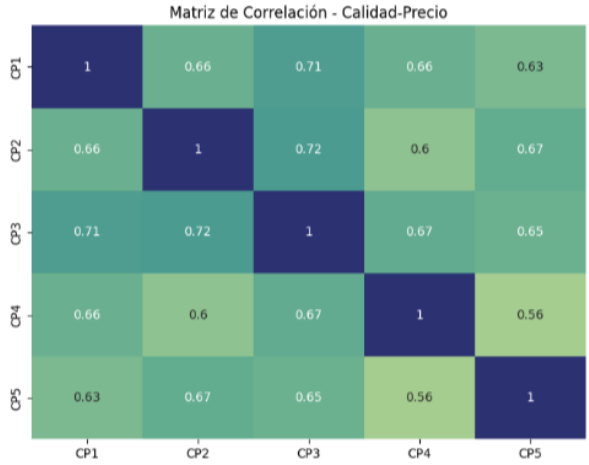

Matriz de Correlación - Conocimiento de Marca
CM1
CM2
CM3
CM4
CM5
1 0.68 0.6 0.57 0.63
0.68 1 0.54 0.35 0.42
0.6 0.54 1 0.76 0.76
0.57 0.35 0.76 1 0.88
0.63 0.42 0.76 0.88 1

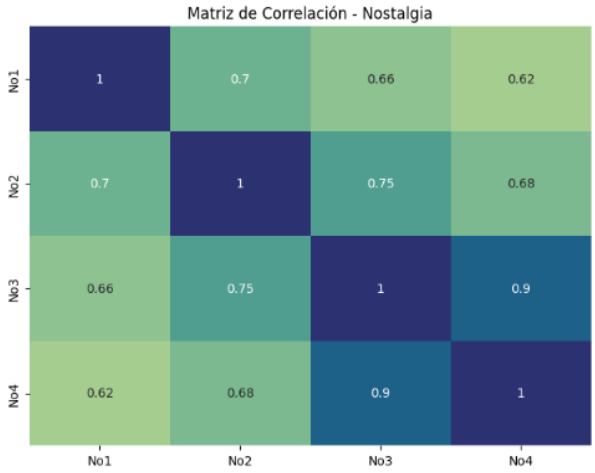

Matriz de Correlación - Sostenibilidad
So1
So2
So3
So4
So5
So6
1 0.7 0.43 0.4 0.69 0.3
0.7 1 0.51 0.45 0.65 0.28
0.43 0.51 1 0.79 0.45 0.49
0.4 0.45 0.79 1 0.5 0.52
0.69 0.65 0.45 0.5 1 0.38
0.3 0.28 0.49 0.52 0.38 1

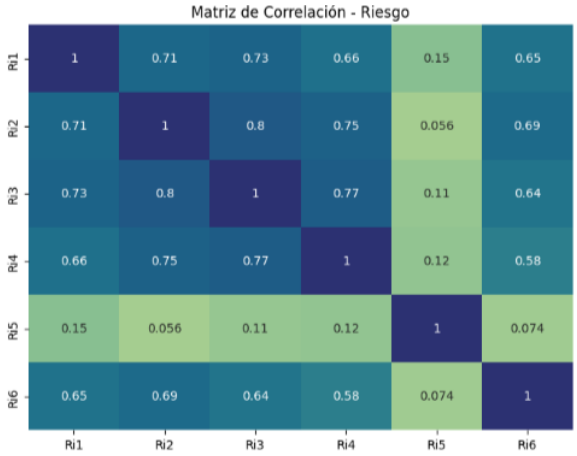

Matriz de Correlación - Valor de Reventa
VR1
VR2
VR3
VR4
1 0.71 0.6 0.58
0.71 1 0.54 0.6
0.6 0.54 1 0.8
0.58 0.6 0.8 1


Matriz de Correlación - Singularidad
Si1
Si2
Si3
Si4
1 0.83 0.55 0.56
0.83 1 0.63 0.66
0.55 0.63 1 0.69
0.56 0.66 0.69 1


Matriz de Correlación - Creatividad y Emoción
CE1
CE2
CE3
CE4
1
0.86
0.78
0.8
0.86
1
0.83
0.83
0.78
0.83
1
0.92
0.8
0.83
0.92
1
CE1
CE2
CE3
CE4


Matriz de Correlación - Calidad-Precio
CP1
CP2
CP3
CP4
CP5
1
0.66
0.71
0.66
0.63
0.66
1
0.72
0.6
0.67
0.71
0.72
1
0.67
0.65
0.66
0.6
0.67
1
0.56
0.63
0.67
0.65
0.56
1
CP1
CP2
CP3
CP4
CP5


Matriz de Correlación - Nostalgia
No1
No2
No3
No4
1
0.7
0.66
0.62
0.7
1
0.75
0.68
0.66
0.75
1
0.9
0.62
0.68
0.9
1
No1
No2
No3
No4



Vemos que tenemos alta correlación entre muchas variables de cada constructo, por lo que eliminaremos las siguientes afirmaciones con el fin de eliminar las correlaciones superiores a 0,7: SC6, SC5, PBC3, CM5, So4, VR4, Si2, CE2, CE4, CP3, No3, Ri2, SN4.

```
# Lista de afirmaciones a eliminar
eliminar_items = ['SC2', 'SC6', 'SC5', 'A2', 'PBC3', 'CM5', 'So4', 'VR4', 'Si2', 'CE2', 'CE4', 'CP3', 'No3', 'Ri2', 'SN4', 'Ri3', 'CM3', 'So1', 'VR1']

# Eliminar del diccionario de constructos
for constructo, variables in constructos.items():
    constructos[constructo] = [item for item in variables if item not in eliminar_items]

# Eliminar del DataFrame df_cuant
df_cuant = df_cuant.drop(columns=eliminar_items, errors='ignore')

# Eliminar del diccionario model_dict
for key, variables in model_dict.items():
    model_dict[key] = [item for item in variables if item not in eliminar_items]

# Mostrar diccionario actualizado y dimensiones del DataFrame
print("Constructos actualizados:", constructos)
print("\nDimensiones del DataFrame después de la eliminación:", df_cuant.shape)
df_cuant.columns
```

```
Constructos actualizados: {'Bandwagon Consumption': ['BC1', 'BC2', 'BC3', 'BC4', 'BC5', 'BC6'], 'Snob Consumption': ['SC1', 'SC3', 'SC4', 'SC7'], 'Normas Subjetivas': ['SN1', 'SN2', 'SN3'], 'Actitudes': ['A1', 'A3', 'A4', 'A5'],

Dimensiones del DataFrame después de la eliminación: (101, 47)
Index(['Intencion_Compra', 'BC1', 'BC2', 'BC3', 'BC4', 'BC5', 'BC6', 'SC1',
       'SC3', 'SC4', 'SC7', 'A1', 'A3', 'A4', 'A5', 'SN1', 'SN2', 'SN3',
       'PBC1', 'PBC2', 'PBC4', 'PBC5', 'So2', 'So3', 'So5', 'So6', 'CM1',
       'CM2', 'CM4', 'Si1', 'Si3', 'Si4', 'CP1', 'CP2', 'CP4', 'CP5', 'No1',
       'No2', 'No4', 'Ri1', 'Ri4', 'Ri5', 'Ri6', 'VR2', 'VR3', 'CE1', 'CE3'],
      dtype='object')
```

2. MODELOS SUPERVISADOS INICIALES

Mediante regresión múltiple, montaremos los modelos que hemos presentado en el marco teórico, y probaremos su efectividad predicitiva.

```
#Comprobamos las columnas del DF con las que nos hemos quedado
df_cuant.columns
```

```
Index(['Intencion_Compra', 'BC1', 'BC2', 'BC3', 'BC4', 'BC5', 'BC6', 'SC1',
       'SC3', 'SC4', 'SC7', 'A1', 'A3', 'A4', 'A5', 'SN1', 'SN2', 'SN3',
       'PBC1', 'PBC2', 'PBC4', 'PBC5', 'So2', 'So3', 'So5', 'So6', 'CM1',
       'CM2', 'CM4', 'Si1', 'Si3', 'Si4', 'CP1', 'CP2', 'CP4', 'CP5', 'No1',
       'No2', 'No4', 'Ri1', 'Ri4', 'Ri5', 'Ri6', 'VR2', 'VR3', 'CE1', 'CE3'],
      dtype='object')
```

```
#agrupamos las variables dependientes e independientes y hacemos la media de cada variable para que sea solo una variable, ya que para hacer el modelo se necesita solo 1
#variables independientes
df_cuant['Bandwagon_Consumption'] =np.mean(df[['BC1', 'BC2', 'BC3', 'BC4', 'BC5', 'BC6']], axis=1)
df_cuant['Snob_Consumption'] =np.mean(df[['SC1', 'SC3', 'SC4', 'SC7']], axis=1)
df_cuant['Conocimiento_Marca'] =np.mean(df[['CM1', 'CM2', 'CM4']], axis=1)
df_cuant['Sostenibilidad'] =np.mean(df[['So2', 'So3', 'So5', 'So6']], axis=1)
df_cuant['Valor_Reventa'] =np.mean(df[['VR2', 'VR3']], axis=1)
df_cuant['Singularidad'] =np.mean(df[['Si1', 'Si3', 'Si4']], axis=1)
df_cuant['Creatividad_Emocion'] =np.mean(df[['CE1', 'CE3']], axis=1)
df_cuant['Calidad_Precio'] =np.mean(df[['CP1', 'CP2', 'CP4', 'CP5']], axis=1)
df_cuant['Nostalgia'] =np.mean(df[['No1', 'No2', 'No4']], axis=1)
df_cuant['Riesgo'] =np.mean(df[['Ri1', 'Ri4', 'Ri5', 'Ri6']], axis=1)

#variables dependientes
df_cuant['Actitudes'] =np.mean(df[['A1', 'A3', 'A4', 'A5']], axis=1)
df_cuant['PBC'] =np.mean(df[['PBC1', 'PBC2', 'PBC4', 'PBC5']], axis=1)
df_cuant['Normas_subjetivas'] =np.mean(df[['SN1', 'SN2', 'SN3']], axis=1)
```

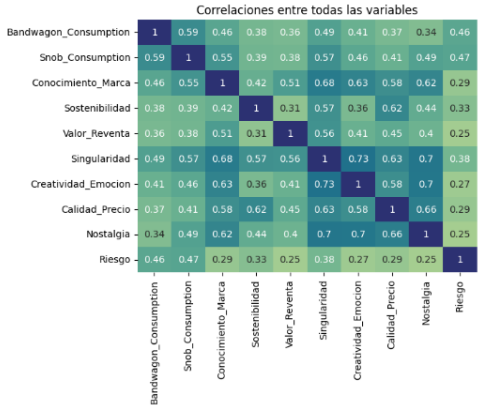
```
# Eliminamos las sub-variables originales del DataFrame
variables_originales = [
    'BC1', 'BC2', 'BC3', 'BC4', 'BC5', 'BC6',
    'SC1', 'SC3', 'SC4', 'SC7',
    'CM1', 'CM2', 'CM4',
    'So2', 'So3', 'So5', 'So6',
    'VR2', 'VR3',
    'Si1', 'Si3', 'Si4',
    'CE1', 'CE3',
    'CP1', 'CP2', 'CP4', 'CP5',
    'No1', 'No2', 'No4',
    'Ri1', 'Ri4', 'Ri5', 'Ri6',
    'A1', 'A3', 'A4', 'A5',
    'PBC1', 'PBC2', 'PBC4', 'PBC5',
    'SN1', 'SN2', 'SN3'
]

df_cuant.drop(columns=variables_originales, inplace=True)
```

```
# Verificacion columnas
df_cuant.columns
```

```
Index(['Intencion_Compra', 'Bandwagon_Consumption', 'Snob_Consumption',
       'Conocimiento_Marca', 'Sostenibilidad', 'Valor_Reventa', 'Singularidad',
       'Creatividad_Emocion', 'Calidad_Precio', 'Nostalgia', 'Riesgo',
       'Actitudes', 'PBC', 'Normas_subjetivas'],
      dtype='object')
```

```
#creamos un df con las variables independientes y otro con las dependientes
variables_independientes=['Bandwagon_Consumption', 'Snob_Consumption', 'Conocimiento_Marca', 'Sostenibilidad', 'Valor_Reventa', 'Singularidad', 'Creatividad_Emocion', 'Calidad_Precio', 'Nostalgia', 'Riesgo']
variables_dependientes=['Actitudes', 'Normas_subjetivas', 'PBC']
variables=variables_independientes+variables_dependientes
```

```
#visualizamos los resultados en una matriz de correlación entre las variables dependientes para obtener información previa al hacer el modelo logit
sns.heatmap(df_cuant[variables_independientes].corr(), annot=True, cbar=False, cmap='crest')
plt.title("Correlaciones entre todas las variables")
plt.show()
```



Regresión múltiple

```
from zlib import DEF_BUF_SIZE
#creamos las variables adicionales, las interacciones, y las unimos a las variables independientes

#creamos las variables independientes al cuadrado para captar no linealidades
df_cuant['Bandwagon_Consumption_2'] = df_cuant['Bandwagon_Consumption'] ** 2
df_cuant['Snob_Consumption_2'] = df_cuant['Snob_Consumption'] ** 2
df_cuant['Conocimiento_Marca_2'] = df_cuant['Conocimiento_Marca'] ** 2
df_cuant['Sostenibilidad_2'] = df_cuant['Sostenibilidad'] ** 2
df_cuant['Valor_Reventa_2'] = df_cuant['Valor_Reventa'] ** 2
df_cuant['Singularidad_2'] = df_cuant['Singularidad'] ** 2
df_cuant['Creatividad_Emocion_2'] = df_cuant['Creatividad_Emocion'] ** 2
df_cuant['Nostalgia_2'] = df_cuant['Nostalgia'] ** 2
df_cuant['Calidad_Precio_2'] = df_cuant['Calidad_Precio'] ** 2
df_cuant['Riesgo_2']=df_cuant['Riesgo'] ** 2

#creamos las iteracciones entre variables basadas en lo que establecimos en el modelo teórico
df_cuant['BC_CM'] = df_cuant['Bandwagon_Consumption'] * df_cuant['Conocimiento_Marca']
df_cuant['BC_So'] = df_cuant['Bandwagon_Consumption'] * df_cuant['Sostenibilidad']
df_cuant['SC_CM'] = df_cuant['Snob_Consumption'] * df_cuant['Conocimiento_Marca']
df_cuant['SC_Si'] = df_cuant['Snob_Consumption'] * df_cuant['Singularidad']
df_cuant['PBC_Ri'] = df_cuant['PBC'] * df_cuant['Riesgo']

variables_interacciones=['BC_CM', 'BC_So', 'SC_CM', 'SC_Si', 'PBC_Ri']
variables_cuadrados= ['Bandwagon_Consumption_2', 'Snob_Consumption_2', 'Conocimiento_Marca_2', 'Sostenibilidad_2', 'Valor_Reventa_2', 'Singularidad_2', 'Creatividad_Emocion_2', 'Nostalgia_2', 'Calidad_Precio_2', 'Riesgo_2']
variables_totales=variables_independientes+variables_interacciones+variables_dependientes+variables_cuadrados
```

```
#Creamos un conjunto para cada modelo con las variables independeintes que le queremos meter, tal y como los definimos en el marco teórico
var_indep_actitudes=['Conocimiento_Marca', 'Sostenibilidad', 'Valor_Reventa', 'Singularidad', 'Creatividad_Emocion', 'Calidad_Precio', 'Nostalgia', 'Riesgo']

var_indep_PBC=['Riesgo']

var_indep_normas_subjetivas=['Bandwagon_Consumption', 'Snob_Consumption', 'BC_CM', 'BC_So', 'SC_CM', 'SC_Si']
```

```
#Creamos la funcion para aplicar la regresion multiple
def regresion_multiple(df, variable_dependiente, variables_independientes):
    # Dividimos los datos en entrenamiento y prueba
    X_train, X_test, y_train, y_test = train_test_split(df_cuant[variables_independientes], df_cuant[variable_dependiente], test_size=0.2, random_state=21)

    # Definir la fórmula del modelo (sin espacios en los nombres)
    formula = f'{variable_dependiente} ~ {" + ".join(variables_independientes)}'

    # Ajustar el modelo
    model = smf.ols(formula=formula, data=df_cuant).fit()

    # Evaluación del modelo
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)

    print(f"\nEvaluación del modelo para {variable_dependiente}:")
    print(f"Mean Squared Error (MSE): {mse:.4f}")
    print(model.summary())  # Resumen del modelo

    return model, mse
```

```
# Aplicar la regresión a cada variable dependiente
regresion_actitudes= regresion_multiple(df_cuant, 'Actitudes', var_indep_actitudes)
regresion_PBC = regresion_multiple(df_cuant, 'PBC', var_indep_PBC)
regresion_normas_subjetivas = regresion_multiple(df_cuant, 'Normas_subjetivas', var_indep_normas_subjetivas)
```

```
Evaluación del modelo para Actitudes:
Mean Squared Error (MSE): 1.1865
                            OLS Regression Results
==============================================================================
Dep. Variable:              Actitudes   R-squared:                       0.687
Model:                            OLS   Adj. R-squared:                  0.660
Method:                 Least Squares   F-statistic:                     25.27
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           3.71e-20
Time:                        09:58:25   Log-Likelihood:                -135.73
No. Observations:                 101   AIC:                             289.5
Df Residuals:                      92   BIC:                             313.0
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.6245      0.419      1.491      0.139      -0.207       1.456
Conocimiento_Marca  0.0613      0.089      0.692      0.491      -0.115       0.237
Sostenibilidad     0.1003      0.092      1.087      0.280      -0.083       0.284
Valor_Reventa      0.0655      0.076      0.861      0.392      -0.086       0.217
Singularidad       0.1828      0.103      1.779      0.079      -0.021       0.387
Creatividad_Emocion  0.3624      0.078      4.676      0.000       0.209       0.516
Calidad_Precio     0.2830      0.105      2.706      0.008       0.075       0.491
Nostalgia         -0.0929      0.090     -1.029      0.306      -0.272       0.086
Riesgo            -0.0711      0.081     -0.877      0.383      -0.232       0.090
==============================================================================
Omnibus:                        5.151   Durbin-Watson:                   1.938
Prob(Omnibus):                  0.076   Jarque-Bera (JB):                6.183
Skew:                           0.221   Prob(JB):                       0.0454
Kurtosis:                       4.129   Cond. No.                         52.7
==============================================================================
```

```
Evaluación del modelo para PBC:
Mean Squared Error (MSE): 1.7315
                            OLS Regression Results
==============================================================================
Dep. Variable:                    PBC   R-squared:                       0.142
Model:                            OLS   Adj. R-squared:                  0.133
Method:                 Least Squares   F-statistic:                     16.38
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           0.000103
Time:                        09:58:25   Log-Likelihood:                -167.93
No. Observations:                 101   AIC:                             339.9
Df Residuals:                      99   BIC:                             345.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.5115      0.394      6.382      0.000       1.731       3.292
Riesgo         0.3963      0.098      4.048      0.000       0.202       0.591
==============================================================================
Omnibus:                        1.518   Durbin-Watson:                   2.171
Prob(Omnibus):                  0.468   Jarque-Bera (JB):                1.282
Skew:                          -0.093   Prob(JB):                        0.527
Kurtosis:                       2.481   Cond. No.                         13.0
==============================================================================
```

```
Evaluación del modelo para Normas_subjetivas:
Mean Squared Error (MSE): 0.4877
                            OLS Regression Results
==============================================================================
Dep. Variable:      Normas_subjetivas   R-squared:                       0.399
Model:                            OLS   Adj. R-squared:                  0.361
Method:                 Least Squares   F-statistic:                     10.40
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           8.06e-09
Time:                        09:58:25   Log-Likelihood:                -131.04
No. Observations:                 101   AIC:                             276.1
Df Residuals:                      94   BIC:                             294.4
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.7168      0.335      5.124      0.000       1.051       2.382
Bandwagon_Consumption  0.2656      0.323      0.824      0.412      -0.375       0.906
Snob_Consumption  -0.0159      0.250     -0.064      0.949      -0.513       0.481
BC_CM             -0.0238      0.068     -0.352      0.726      -0.158       0.110
BC_So              0.0491      0.027      1.833      0.070      -0.004       0.102
SC_CM             -0.0158      0.052     -0.302      0.763      -0.120       0.088
SC_Si              0.0330      0.019      1.730      0.087      -0.005       0.071
==============================================================================
Omnibus:                       11.296   Durbin-Watson:                   2.301
Prob(Omnibus):                  0.004   Jarque-Bera (JB):               11.738
Skew:                           0.723   Prob(JB):                      0.00283
Kurtosis:                       3.834   Cond. No.                         168.
==============================================================================
```

```
[ ] # Tratemos de mejorar el modelo elevando al cuadrado las variables con Pvalor más cercano a cero, y eliminando las que tienen Pvalor muy alto y coeficiente cercano a cero
    #Creamos un conjunto para cada modelo con las variables independeintes que le queremos meter

    #Elevamos al cuadrado Creatividad Emocion y Calidad Precio, y eliminamos Conocimiento de Marca y Riesgo
    var_indep_actitudes=['Sostenibilidad', 'Valor_Reventa', 'Singularidad', 'Creatividad_Emocion', 'Calidad_Precio', 'Nostalgia', 'Creatividad_Emocion_2', 'Calidad_Precio_2']

    #Elevamos al cuadrado Riesgo
    var_indep_PBC=['Riesgo', 'Riesgo_2']

    #Eliminamos BC_CM, Snob_Consumption y SC_CM
    var_indep_normas_subjetivas=['Bandwagon_Consumption', 'BC_So', 'SC_Si']
```

```
    # predecimos y calculamos el error para cada variable dependiente
    model_actitudes, mse_actitudes = regresion_multiple(df_cuant, 'Actitudes', var_indep_actitudes)
    model_PBC, mse_PBC = regresion_multiple(df_cuant, 'PBC', var_indep_PBC)
    model_normas_subjetivas, mse_normas_subjetivas = regresion_multiple(df, 'Normas_subjetivas', var_indep_normas_subjetivas)
```

```
Evaluación del modelo para Actitudes:
Mean Squared Error (MSE): 1.2897
                            OLS Regression Results
==============================================================================
Dep. Variable:              Actitudes   R-squared:                       0.691
Model:                            OLS   Adj. R-squared:                  0.664
Method:                 Least Squares   F-statistic:                     25.71
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           2.18e-20
Time:                        10:20:54   Log-Likelihood:                -135.13
No. Observations:                 101   AIC:                             288.3
Df Residuals:                      92   BIC:                             311.8
Df Model:                           8
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept                 0.9410      0.761      1.237      0.219      -0.570       2.452
Sostenibilidad            0.0837      0.093      0.898      0.372      -0.101       0.269
Valor_Reventa             0.0861      0.075      1.154      0.252      -0.062       0.234
Singularidad              0.1764      0.100      1.758      0.082      -0.023       0.376
Creatividad_Emocion       0.0505      0.232      0.218      0.828      -0.410       0.511
Calidad_Precio            0.2847      0.395      0.722      0.472      -0.499       1.068
Nostalgia                -0.0760      0.089     -0.852      0.397      -0.253       0.101
Creatividad_Emocion_2     0.0429      0.030      1.440      0.153      -0.016       0.102
Calidad_Precio_2          0.0004      0.043      0.008      0.993      -0.085       0.086
==============================================================================
Omnibus:                        6.633   Durbin-Watson:                   1.934
Prob(Omnibus):                  0.036   Jarque-Bera (JB):                7.593
Skew:                           0.363   Prob(JB):                       0.0224
Kurtosis:                       4.130   Cond. No.                         321.
==============================================================================
```

```
Evaluación del modelo para PBC:
Mean Squared Error (MSE): 1.7296
                            OLS Regression Results
==============================================================================
Dep. Variable:                    PBC   R-squared:                       0.144
Model:                            OLS   Adj. R-squared:                  0.127
Method:                 Least Squares   F-statistic:                     8.256
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           0.000486
Time:                        10:20:54   Log-Likelihood:                -167.80
No. Observations:                 101   AIC:                             341.6
Df Residuals:                      98   BIC:                             349.4
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.1391      0.839      2.550      0.012       0.474       3.804
Riesgo         0.6403      0.495      1.294      0.199      -0.341       1.622
Riesgo_2      -0.0343      0.068     -0.503      0.616      -0.170       0.101
==============================================================================
Omnibus:                        1.274   Durbin-Watson:                   2.166
Prob(Omnibus):                  0.529   Jarque-Bera (JB):                1.140
Skew:                          -0.081   Prob(JB):                        0.566
Kurtosis:                       2.506   Cond. No.                         144.
==============================================================================
```

```
Evaluación del modelo para Normas_subjetivas:
Mean Squared Error (MSE): 0.4258
                            OLS Regression Results
==============================================================================
Dep. Variable:      Normas_subjetivas   R-squared:                       0.378
Model:                            OLS   Adj. R-squared:                  0.359
Method:                 Least Squares   F-statistic:                     19.68
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           4.78e-10
Time:                        10:20:54   Log-Likelihood:                -132.74
No. Observations:                 101   AIC:                             273.5
Df Residuals:                      97   BIC:                             283.9
Df Model:                           3
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept                 1.7666      0.261      6.774      0.000       1.249       2.284
Bandwagon_Consumption     0.1404      0.145      0.968      0.335      -0.147       0.428
BC_So                     0.0482      0.025      1.931      0.056      -0.001       0.098
SC_Si                     0.0142      0.011      1.343      0.182      -0.007       0.035
==============================================================================
Omnibus:                       12.275   Durbin-Watson:                   2.329
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               13.035
Skew:                           0.763   Prob(JB):                      0.00148
Kurtosis:                       3.875   Cond. No.                         84.3
==============================================================================
```

En cuanto al modelo de Actitudes, los resultados no mejoran por lo que lo dejaremos como estaba.

Al modelo PBC le sucede lo mismo, lo dejaremos como el original.

El modelo de Normas Subjetivas si ha mejorado un poco más, por lo que lo dejaremos con los cambios.

```
[ ] # Probemos finalmente, sin metemos interacciones que no estaban presentes en el modelo teórico pero que podrían servir para reducir correlaciones
    # Las hemos basado en el cálculo de correlaciones realizado antes de montar los modelos
    df_cuant['BC_SC'] = df_cuant['Bandwagon_Consumption'] * df_cuant['Snob_Consumption']
    df_cuant['Si_CM'] = df_cuant['Singularidad'] * df_cuant['Conocimiento_Marca']
    df_cuant['CE_CM'] = df_cuant['Creatividad_Emocion'] * df_cuant['Conocimiento_Marca']
    df_cuant['No_CM'] = df_cuant['Nostalgia'] * df_cuant['Conocimiento_Marca']
    df_cuant['CE_Si'] = df_cuant['Creatividad_Emocion'] * df_cuant['Singularidad']
    df_cuant['No_Si'] = df_cuant['Nostalgia'] * df_cuant['Singularidad']
    df_cuant['No_CP'] = df_cuant['Nostalgia'] * df_cuant['Calidad_Precio']
```

```
[ ] #Creamos un conjunto para cada modelo con las variables independeintes que le queremos meter
    var_indep_actitudes=['Sostenibilidad', 'Valor_Reventa', 'Singularidad', 'Creatividad_Emocion', 'Calidad_Precio', 'Nostalgia', 'Riesgo', 'Si_CM', 'CE_CM', 'No_CM', 'CE_Si', 'No_Si', 'No_CP']

    var_indep_PBC=['Riesgo']

    var_indep_normas_subjetivas=['Bandwagon_Consumption', 'BC_So', 'SC_Si', 'BC_SC']
```

```
# predecimos y calculamos el error para cada variable dependiente
model_actitudes, mse_actitudes = regresion_multiple(df_cuant, 'Actitudes', var_indep_actitudes)
model_PBC, mse_PBC = regresion_multiple(df_cuant, 'PBC', var_indep_PBC)
model_normas_subjetivas, mse_normas_subjetivas = regresion_multiple(df, 'Normas_subjetivas', var_indep_normas_subjetivas)
```

```
Evaluación del modelo para Actitudes:
Mean Squared Error (MSE): 1.1762
                            OLS Regression Results
==============================================================================
Dep. Variable:              Actitudes   R-squared:                       0.699
Model:                            OLS   Adj. R-squared:                  0.654
Method:                 Least Squares   F-statistic:                     15.57
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           1.53e-17
Time:                        10:23:20   Log-Likelihood:                -133.73
No. Observations:                 101   AIC:                             295.5
Df Residuals:                      87   BIC:                             332.1
Df Model:                          13
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.4631      0.700      2.090      0.040       0.072       2.854
Sostenibilidad 0.1274      0.095      1.339      0.184      -0.062       0.317
Valor_Reventa  0.0795      0.080      0.991      0.324      -0.080       0.239
Singularidad   0.1729      0.348      0.497      0.620      -0.518       0.864
Creatividad_Emocion 0.3278 0.296      1.109      0.271      -0.260       0.915
Calidad_Precio 0.0815      0.246      0.331      0.741      -0.407       0.570
Nostalgia     -0.3776      0.354     -1.067      0.289      -1.081       0.326
Riesgo        -0.0407      0.084     -0.484      0.630      -0.208       0.126
Si_CM          0.0061      0.061      0.101      0.920      -0.115       0.127
CE_CM          0.0691      0.064      1.079      0.284      -0.058       0.197
No_CM         -0.0629      0.076     -0.828      0.410      -0.214       0.088
CE_Si         -0.0642      0.057     -1.131      0.261      -0.177       0.049
No_Si          0.0604      0.066      0.920      0.360      -0.070       0.191
No_CP          0.0601      0.062      0.976      0.332      -0.062       0.183
==============================================================================
Omnibus:                        5.787   Durbin-Watson:                   1.975
Prob(Omnibus):                  0.055   Jarque-Bera (JB):                6.689
Skew:                           0.297   Prob(JB):                       0.0353
Kurtosis:                       4.112   Cond. No.                         416.
==============================================================================
```

```
Evaluación del modelo para PBC:
Mean Squared Error (MSE): 1.7315
                            OLS Regression Results
==============================================================================
Dep. Variable:                    PBC   R-squared:                       0.142
Model:                            OLS   Adj. R-squared:                  0.133
Method:                 Least Squares   F-statistic:                     16.38
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           0.000103
Time:                        10:23:20   Log-Likelihood:                -167.93
No. Observations:                 101   AIC:                             339.9
Df Residuals:                      99   BIC:                             345.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.5115      0.394      6.382      0.000       1.731       3.292
Riesgo         0.3963      0.098      4.048      0.000       0.202       0.591
==============================================================================
Omnibus:                        1.518   Durbin-Watson:                   2.171
Prob(Omnibus):                  0.468   Jarque-Bera (JB):                1.282
Skew:                          -0.093   Prob(JB):                        0.527
Kurtosis:                       2.481   Cond. No.                         13.0
==============================================================================
```

```
Evaluación del modelo para Normas_subjetivas:
Mean Squared Error (MSE): 0.4556
                            OLS Regression Results
==============================================================================
Dep. Variable:      Normas_subjetivas   R-squared:                       0.380
Model:                            OLS   Adj. R-squared:                  0.354
Method:                 Least Squares   F-statistic:                     14.71
Date:                Sun, 16 Feb 2025   Prob (F-statistic):           2.08e-09
Time:                        10:23:20   Log-Likelihood:                -132.61
No. Observations:                 101   AIC:                             275.2
Df Residuals:                      96   BIC:                             288.3
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.6601      0.337      4.926      0.000       0.991       2.329
Bandwagon_Consumption 0.2247 0.222     1.010      0.315      -0.217       0.666
BC_So          0.0471      0.025      1.873      0.064      -0.003       0.097
SC_Si          0.0187      0.014      1.345      0.182      -0.009       0.046
BC_SC         -0.0158      0.032     -0.501      0.617      -0.078       0.047
==============================================================================
Omnibus:                       12.613   Durbin-Watson:                   2.348
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               13.522
Skew:                           0.774   Prob(JB):                      0.00116
Kurtosis:                       3.905   Cond. No.                         136.
==============================================================================
```

Los modelos no mejoran, los dejaremos como habíamos establecido.

3. MODELO GLOBAL

Ahora que ya tenemos los 3 modelos, ahora crearemos uno global y lo evaluaremos. Primeramente, incluiremos solo como variables independientes las tres variables relativas a los componentes de la TPB.

```
[ ] variables_dependientes=['Actitudes', 'Normas_subjetivas', 'PBC']
```

```
[ ] variables_totales2 = variables_dependientes
    variables_totales2
```

```
['Actitudes', 'Normas_subjetivas', 'PBC']
```

```
# En este modelo la variable dependiente será la intención de compra (IC)
model_IC, mse_IC = regresion_multiple(df_cuant, 'Intencion_Compra', variables_totales2)
```

```
Evaluación del modelo para Intencion_Compra:
Mean Squared Error (MSE): 2.1999
                            OLS Regression Results
==============================================================================
Dep. Variable:       Intencion_Compra   R-squared:                       0.441
Model:                           OLS   Adj. R-squared:                  0.424
Method:                Least Squares   F-statistic:                     25.51
Date:               Sun, 16 Feb 2025   Prob (F-statistic):           2.99e-12
Time:                       10:50:53   Log-Likelihood:                -188.01
No. Observations:                101   AIC:                             384.0
Df Residuals:                     97   BIC:                             394.5
Df Model:                          3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.3400      0.570      0.597      0.552      -0.791       1.471
Actitudes      0.5188      0.136      3.814      0.000       0.249       0.789
Normas_subjetivas    -0.2749      0.156     -1.762      0.081      -0.585       0.035
PBC            0.5593      0.159      3.517      0.001       0.244       0.875
==============================================================================
Omnibus:                        2.472   Durbin-Watson:                   2.044
Prob(Omnibus):                  0.290   Jarque-Bera (JB):                1.723
Skew:                           0.100   Prob(JB):                        0.423
Kurtosis:                       2.392   Cond. No.                         26.0
==============================================================================
```

Dado que le resultado no es muy bueno, incluyamos las variables significativas de los modelos anteriores a modo de interacciones con sus variables dependientes respectivas

```
[ ] df_cuant['PBC_Ri'] = df_cuant['Riesgo'] * df_cuant['PBC']
    df_cuant['Si_A'] = df_cuant['Singularidad'] * df_cuant['Actitudes']
    df_cuant['CE_A'] = df_cuant['Creatividad_Emocion'] * df_cuant['Actitudes']
    df_cuant['CP_A'] = df_cuant['Calidad_Precio'] * df_cuant['Actitudes']
    variables_totales2 += ['PBC_Ri', 'Si_A', 'CE_A', 'CP_A', 'BC_So', 'SC_Si']
```

```
[ ] variables_totales2
```

```
['Actitudes',
 'Normas_subjetivas',
 'PBC',
 'PBC_Ri',
 'Si_A',
 'CE_A',
 'CP_A',
 'BC_So',
 'SC_Si']
```

```
# En este modelo la variable dependiente será la intención de compra (IC)
model_IC, mse_IC = regresion_multiple(df_cuant, 'Intencion_Compra', variables_totales2)
```

```
Evaluación del modelo para Intencion_Compra:
Mean Squared Error (MSE): 1.9580
                            OLS Regression Results
==============================================================================
Dep. Variable:       Intencion_Compra   R-squared:                       0.476
Model:                           OLS   Adj. R-squared:                  0.425
Method:                Least Squares   F-statistic:                     9.197
Date:               Sun, 16 Feb 2025   Prob (F-statistic):           8.65e-10
Time:                       10:51:01   Log-Likelihood:                -184.72
No. Observations:                101   AIC:                             389.4
Df Residuals:                     91   BIC:                             415.6
Df Model:                          9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0525      0.761     -0.069      0.945      -1.563       1.458
Actitudes      0.8233      0.306      2.693      0.008       0.216       1.431
Normas_subjetivas    -0.3029      0.188     -1.612      0.111      -0.676       0.070
PBC            0.4111      0.235      1.746      0.084      -0.057       0.879
PBC_Ri         0.0136      0.035      0.393      0.695      -0.055       0.082
Si_A          -0.0912      0.041     -2.199      0.030      -0.174      -0.009
CE_A           0.0096      0.027      0.353      0.725      -0.044       0.064
CP_A           0.0098      0.040      0.246      0.806      -0.069       0.089
BC_So         -0.0158      0.028     -0.553      0.581      -0.072       0.041
SC_Si          0.0639      0.031      2.040      0.044       0.002       0.126
==============================================================================
Omnibus:                        2.867   Durbin-Watson:                   1.982
Prob(Omnibus):                  0.238   Jarque-Bera (JB):                1.905
Skew:                           0.113   Prob(JB):                        0.386
Kurtosis:                       2.367   Cond. No.                         261.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Este es el modelo final