

**Grado en Ingeniería Telemática**

**Trabajo Fin de Grado**

Sistema de captura de leads a través de páginas landing para grandes volúmenes de datos

ESCUELA POLITECNICA  
SUPERIOR

**Autor:** Carlos Moreno Guillén

**Tutor/es:** Óscar García Población

2022



# UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

## Grado en Ingeniería Telemática

### Trabajo Fin de Grado

Sistema de captura de leads a través de páginas landing para grandes volúmenes de datos

**Autor:** Carlos Moreno Guillén

**Tutor/es:** Óscar García Población

**TRIBUNAL:**

**Presidente:** María del Mar Lendínez Chica

**Vocal 1º:** Concepción Batanero Ochaíta

**Vocal 2º:** Óscar García Población

**FECHA:** 2022



*“Hay quienes buscan el conocimiento por el bien del conocimiento; eso es **curiosidad**.  
Hay quienes buscan el conocimiento para ser reconocidos por los demás; eso es **vanidad**.  
Hay quienes buscan el conocimiento para servir; eso es **amor**.”*

San Bernardo de Claraval



# Agradecimientos

A mi familia y en especial a mi madre por apoyarme siempre en los momentos más difíciles.

A mis amigos y a mis compañeros de la carrera por compartir buenos momentos y por estar ahí en los malos también.

A los profesores de varias asignaturas que me he encontrado a lo largo de la carrera y que con su entrega y sus ganas de enseñar he conseguido llegar hasta aquí.

A mi tutor Óscar por ayudarme a sacar adelante este proyecto.





# Resumen

Una de las partes esenciales del Inbound Marketing es el de la captura de los leads procedentes de las landing pages para su posterior procesamiento y almacenamiento en una base de datos de la empresa. Los leads son los datos de clientes potenciales, que voluntariamente ceden para que una empresa se ponga en contacto con ellos en relación a un producto o servicio de su interés.

El objetivo de este proyecto es construir un sistema software que abarque todo el proceso de captura de los leads desde la creación de la página donde se ofrecen los productos o servicios (landing page) hasta su almacenamiento en una base de datos y comprobando que responde de manera adecuada ante un posible gran número de peticiones a través de un test de carga.



# Abstract

One of the essential parts of inbound marketing is the lead capture from landing pages for further processing and storage in a company database. Leads are potential customer data, which they voluntarily give so that a company can contact them in relation to a product or service of their interest.

The objective of this project is to build a software system that covers the entire process of lead capture from the creation of the page where the products or services are offered (landing page) to their storage in a database and checking that it responds adequately to a possible large number of requests through a load test.



# Palabras clave

Inbound Marketing, lead, Wagtail, Django, Python, CMS, Locust



# Índice

<b>Agradecimientos</b>	<b>7</b>
<b>Resumen</b>	<b>9</b>
<b>Abstract</b>	<b>11</b>
<b>Palabras clave</b>	<b>13</b>
<b>Índice</b>	<b>15</b>
<b>1. Introducción</b>	<b>19</b>
1.1 Objetivos y desarrollo del proyecto	19
<b>2. Análisis del problema</b>	<b>21</b>
2.1 Outbound Marketing: problemas para alcanzar a nuevos clientes	21
2.2 Inbound Marketing como solución a las deficiencias del Outbound Marketing	21
2.2.1 Orígenes del Inbound Marketing	22
2.3 La metodología inbound	23
2.3.1 Atraer	24
Crear un prototipo de cliente ideal o buyer persona	25
Optimización para motores de búsqueda o SEO	25
Conectar a través de las redes sociales	27
2.3.2 Involucrar	27
Generación de leads	27
Los CRM como herramienta para gestionar los leads	28
Lead scoring y lead nurturing	30
2.3.3 Deleitar	30
Conocer el nivel de satisfacción de los clientes	31
Herramientas para la fidelización de clientes	32
2.4 Inbound Marketing frente al Outbound	33
Principales diferencias	33
Datos Inbound vs. Outbound	34
2.5 Generación de leads	37
2.5.1 ¿Qué es un lead?	37

2.5.2 Valoración de leads o lead scoring	37
2.5.3 El proceso de generación de leads	38
2.5.4 Call-to-action (CTA)	39
2.5.5 Las landing pages	40
2.6 Problema a resolver	42
<b>3. Diseño de la solución</b>	<b>45</b>
3.1 Generación de landing page utilizando Wagtail	45
3.1.1 ¿Qué es un CMS?	45
3.1.2 ¿Qué es Django?	47
El procesamiento de una solicitud al servidor	48
3.2 Aplicación de Wagtail en la solución	49
3.2.1 Instalación de Wagtail	49
3.2.2 Creación del index page	51
3.2.3 Creación de las páginas de formulario	52
3.2.4 Página principal	55
3.3 Procesar los leads procedentes de las landings con Celery	56
3.3.2 ¿Qué es Celery y cuál es su utilidad?	58
3.4 Aplicación de Celery en la solución	59
3.4.1 Instalación y configuración	59
3.4.2 Creación de tareas	60
3.4.3 Ejecución de Celery	60
3.5 Pruebas de carga con Locust	61
3.5.1 ¿Qué es una prueba de carga?	61
3.5.2 ¿Qué es Locust?	61
3.5.3 Instalación y configuración de Locust	62
¿Qué es un token csrf y para qué sirve?	65
3.5.4 Ejecución de Locust	66
3.6 Garantizar medidas de seguridad con Recaptcha	69
3.6.1 ¿Qué es Recaptcha?	69
3.6.2 Configuración de Recaptcha	69
3.7 Docker y Docker Compose	72



3.7.1 ¿Qué es Docker y Docker Compose?	72
Diferencia entre máquina virtual y Docker	72
3.7.2 Dockerizando nuestra aplicación	73
<b>4. Conclusiones</b>	<b>77</b>
<b>5. Líneas de trabajo futuro</b>	<b>79</b>
<b>Citas y referencias</b>	<b>81</b>
<b>Bibliografía</b>	<b>83</b>



# 1. Introducción

Dentro del mundo del marketing existen diferentes metodologías comerciales para atraer posibles clientes a una empresa para que compren un determinado producto o servicio. Por un lado, estaría el Outbound Marketing [15] que interrumpe constantemente al usuario con publicidad poco relevante y sin su consentimiento. Mientras, por otro lado, está el **Inbound Marketing** [16][17][18], cuyo propósito es generar contenido valioso para el usuario con publicidad hecha a medida que pueda satisfacer sus necesidades, lo cual permite crear relaciones significativas y duraderas con los clientes.

Para poder llevar a cabo el Inbound Marketing necesitamos generar **leads** [20][21][22]. Los leads son datos proporcionados por un usuario a una empresa, los cuales pasan a formar parte de un registro en una base de datos que permite conocer mejor las necesidades de un potencial cliente y ofrecerle productos de manera más eficaz, lo cual lleva a un aumento de los beneficios de dicha empresa. Los leads son una herramienta fundamental dentro de las empresas para promocionar sus productos dentro de su target (público objetivo que estaría interesado en nuestro producto) de la forma más acertada posible. Por ello resulta de vital importancia que estos leads sean procesados y almacenados de manera adecuada.

## 1.1 Objetivos y desarrollo del proyecto

El objetivo del proyecto es construir un sistema software para facilitar la captura de los leads de usuarios con el fin de convertir estos potenciales clientes en contactos con los que poder mantener una relación a largo plazo. Esto lo haremos generando unas **landings** [27][28][29], que son páginas web creadas específicamente para una campaña de marketing donde un visitante llega, normalmente, después de hacer clic en un enlace de un correo electrónico o de algún anuncio. Su objetivo principal es la llamada a la acción (o **CTA** por sus siglas en inglés) [24][25][26], es decir, el siguiente paso o acción que el vendedor quiere que tome el consumidor (por ejemplo, enviar un formulario con sus datos o perfil de gustos). De esta forma, las empresas obtienen información valiosa sobre potenciales clientes para poder ofrecerles los productos que mejor se ajusten a sus necesidades y, de esa forma, incrementar las ventas y la fidelización del cliente.

Además de lo mencionado anteriormente, debemos tener en cuenta que, si damos mucha difusión a nuestra campaña de marketing, es posible que mucha gente acabe accediendo a nuestra landing page, se produzcan avalanchas de datos en periodos cortos de tiempo. Por ese motivo debemos estar preparados para manejar un gran volumen de datos y gestionarlos adecuadamente y de esa forma evitar que nuestro sistema se vea desbordado. Para ello, los objetivos marcados serán los siguientes:

- **Creación de landings** que resulten atrayentes para el usuario, pero además de forma rápida, ya que los ciclos en las campañas de marketing tienen, por lo general, una duración corta. Para lograr este objetivo deberemos:
  - Partir de un diseño HTML generado por otro departamento (por ejemplo, el de UX/UI [Capítulo 2.6]).
  - Gestionar altas, bajas y modificaciones de dichas páginas. Para facilitar dicha gestión usaremos un CMS basado en Django llamado **Wagtail** [Capítulo 3.1].
- **Procesar los leads** procedentes de las landings de tal forma que el usuario no vea reflejado el tiempo de procesamiento, que puede ser largo, en el tiempo de carga de la página web. Para ello:
  - Enviaremos el lead a un sistema intermediario (también llamado broker) como **REDIS** que se encargará simplemente de almacenar los leads y devolver un OK al servidor web. Esto lo haremos a través de un gestor de mensajes distribuidos llamado

**Celery.** En conjunto actuarán como una cola de mensajes de tipo publicador/suscriptor que permitirá tratar los leads de manera asíncrona.

- Finalmente, lo almacenaremos en algún soporte persistente como una base de datos o una hoja de cálculo, como puede ser **Google Sheets**.
- **Garantizar unas medidas mínimas de seguridad informática:**
  - Debido a que los leads recibidos son anónimos, algún usuario malintencionado podría generar un gran número de datos falsos. Lo cual sería claramente contraproducente para nuestra labor de recolección de información. Para poder controlar este tipo de acciones haremos uso de la herramienta **Recaptcha**.
  - Proteger posibles datos sensibles de los usuarios, ya que, lógicamente, a nadie le gustaría ver sus datos personales expuestos al público, además que significaría una pérdida de la confianza en nuestros servicios por parte del cliente.
- Para finalizar haremos **pruebas simulando que hay un número alto de usuarios** que consultan las landings y que dejan leads en las mismas. Esto lo haremos valiéndonos de una herramienta de testeo de carga llamada **Locust**.

Todo esto se hará usando **Docker** y **Docker Compose** [Capítulo 3.7] para facilitar el despliegue de las aplicaciones.

## 2. Análisis del problema

En este apartado vamos a comenzar desarrollando qué es el Inbound Marketing y qué lo diferencia del Outbound Marketing, para de esa forma entender mejor cuál es el motivo de la existencia de las landing pages y la utilidad de la generación de los leads. Por último, vamos a describir cuál sería el problema a resolver, el cual además es el objetivo final de este proyecto.

### 2.1 Outbound Marketing: problemas para alcanzar a nuevos clientes

Como consumidores sabemos lo molesto que es ser expuestos a diario a llamadas de determinados servicios que no tenemos ninguna intención de comprar o de recibir numerosos correos electrónicos que publicitan una serie de productos en los que no estamos interesados y que con alta probabilidad acabaremos mandando a la carpeta de spam<sup>1</sup>. Esto es lo que se conoce como Outbound Marketing.

Entendemos por **Outbound Marketing** aquel en el que la empresa se anuncia a un gran grupo de personas con la esperanza de alcanzar de esa forma a algún potencial cliente que esté interesado en su producto. Esta estrategia se basa en que cuanto mayor sea el número de personas que alcance nuestro mensaje, mayor será la respuesta que recibamos. Entre los ejemplos de Outbound Marketing se incluyen algunos de los métodos tradicionales de publicitarse como pueden ser anuncios de televisión, anuncios de radio, periódicos, vallas publicitarias, folletos, llamadas en frío<sup>2</sup>.

Aunque el Outbound Marketing también se sigue aplicando haciendo uso de tecnologías más modernas como la publicidad de pago por clic<sup>3</sup> o vía correo electrónico. Aunque en la práctica este método resulta poco efectivo [31], ya que con frecuencia los consumidores que reciben este tipo de publicidad ni siquiera están interesados en el producto anunciado, lo cual se traduce en pérdidas de dinero invertido en publicidad que no llega a los potenciales clientes.

### 2.2 Inbound Marketing como solución a las deficiencias del Outbound Marketing

Por otro lado, está el **Inbound Marketing**. El cual viene a solucionar las deficiencias que tenía el Outbound Marketing. Se define como “una forma de marketing orgánico, basado en la estrecha relación entre la empresa y sus prospectos<sup>4</sup> o clientes, quienes han manifestado su interés en los productos de la empresa de forma voluntaria (basada en la suscripción a boletines, blog , redes sociales, etc.) y que se han sentido atraídos e implicados por una alta calidad del marketing de contenidos.” [1]

---

<sup>1</sup> **Spam** (correo basura): “correo electrónico de distribución masiva y contenido normalmente publicitario o malicioso, que se recibe sin haberlo solicitado.” [2]

<sup>2</sup> “Las **llamadas en frío** son una técnica que sirve para iniciar un proceso publicitario, y así captar clientes potenciales. «En frío» refiere a que el prospecto no esperaba al agente de ventas y no se sabe si está interesado en la oferta.” [3]

<sup>3</sup> “El PPC o **pago por clic** es un modelo de publicidad digital en el que el anunciante paga una cantidad (fija o determinada por subasta) cada vez que un usuario hace clic en uno de sus anuncios para visitar su página web. La idea de una campaña de pay per click es “comprar visitas” a un sitio determinado”, por ejemplo la publicidad en Google está basada en este método. [4]

<sup>4</sup> “Un prospecto es un cliente potencial, alguien que está en el mercado de su producto y tiene los recursos necesarios para comprarlo, pero aún no lo ha comprado. Una vez que el prospecto compra el producto, que siempre es el objetivo final, se convierte en cliente.” [5]

Así, la principal finalidad del Inbound Marketing [30] es el acompañar al consumidor desde que conocen nuestro producto hasta que finalmente lo compran siempre de una forma amigable y poco intrusiva para lograr la fidelización del cliente. La idea es que si ofreces algo valioso para tus clientes podrás obtener a cambio su confianza en tu empresa.

El objetivo es no tener que gastar esfuerzos en llamar la atención a grupos de personas que no tienen interés en tu producto, sino que se trata de ofrecer el producto adecuado en el momento correcto para atraer solo a los consumidores que estén interesados o estén buscando un producto o servicio que tú como empresa puedes ofrecerles. [34]

### 2.2.1 Orígenes del Inbound Marketing

El origen del término “Inbound Marketing” fue acuñado en 2005 por Brian Halligan, quien es cofundador y director de la empresa de marketing HubSpot<sup>5</sup>. Sus nuevas ideas en conjunto con las teorías del marketing de su compañero Dharmesh Shah y el asesor de la empresa David Meerman Scott dieron como resultado un concepto innovador que traería consigo una nueva forma de hacer marketing. [32][33]

Aun así, existen algunos precedentes al desarrollo de la metodología inbound que ya utilizaban este tipo de prácticas. Antes de la llegada de este revolucionario concepto, el marketing era casi en su totalidad outbound.

Aunque este término fue difundido recientemente, las bases de esta idea se remontan a los años 50, cuando Peter F. Drucker, quien es considerado uno de los padres del marketing moderno, resaltaba la importancia de conocer los intereses de los consumidores y hacía especial énfasis en la heterogeneidad del mercado y en el hecho de hacer un marketing centrado en el cliente y sus necesidades.

Esta idea fue adoptada posteriormente por el empresario Seth Godin, quien concibió el término de “permission marketing” o marketing de permiso. Con sus ideas quería dejar a un lado el marketing basado en la intrusión de manera agresiva con el cliente. En su lugar, buscar un enfoque más amigable en el que los vendedores buscan el permiso de los clientes para enviar emails, productos promocionales, etc.

Más adelante, en la década de los 90 llegó el concepto de “marketing relacional” de la mano del consultor y experto en marketing Regis McKenna que destacaba la importancia del deber de las empresas de satisfacer a los clientes para poder retenerlos y así crear relaciones mucho más duraderas.

Luego llegó internet que revolucionó por completo el mundo del marketing. Comenzaron a nacer los buscadores, lo que trajo consigo el concepto de SEO (Search Engine Optimization). Más tarde aparecieron las redes sociales y los dispositivos móviles con el que cualquiera tenía acceso a internet desde cualquier parte y en cualquier momento. Rápidamente, internet se hizo omnipresente en la vida de las personas, lo que cambió de forma drástica los hábitos de los consumidores y su forma de comprar.

Y fue en ese momento en 2004 cuando Brian Halligan, David Meerman Scott y Dharmesh Shah que en ese momento estudiaban en el MIT vieron que el modelo del Outbound Marketing estaba en decadencia y fue entonces cuando Brian Halligan y Dharmesh Shah se lanzaron a crear su propia empresa llamada HubSpot. Poco tiempo después se les unió Meerman Scott. Su finalidad era asesorar a otras empresas para comenzar a poner a sus clientes en el centro y tratarlos como personas y no solo como números. De esa forma consiguen atraer a clientes ideales y convertirlos en compradores satisfechos que nos recompensará con su fidelidad y nos recomendarán a otros, lo que generará un círculo virtuoso en el que nuestra empresa no hará más que aumentar sus beneficios.

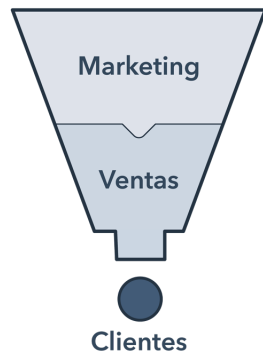
---

<sup>5</sup> HubSpot es un software de ventas e Inbound Marketing que ayuda a las empresas a atraer potenciales clientes y convertirlos en clientes satisfechos que posteriormente promuevan los productos de dicha empresa.

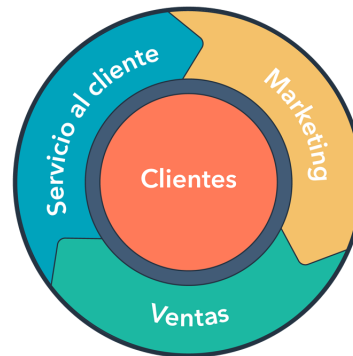
Originalmente, el concepto del Inbound Marketing giraba en torno al concepto del embudo, donde los clientes entraban, compraban y salían por el otro lado del embudo, lo que significaba que salían del proceso de ventas (el ciclo tenía un principio y un final). Pero recientemente en HubSpot cambiaron el concepto de este modelo y pasó a ser un volante o flywheel.

## Del embudo al ciclo

**Antes, con el embudo...**  
los clientes eran la  
última prioridad



**Ahora, con el ciclo...**  
los clientes están en el  
centro del proceso



El cambio del modelo de Inbound Marketing del embudo al volante. (Fuente: [www.hubspot.es](http://www.hubspot.es))

En este modelo, en lugar de salir por el fondo del embudo para no volver, ahora regresan siempre en lo que es un ciclo sin fin en el que los clientes satisfechos vuelven a comprar de forma recurrente y además se convierten en promotores de tu marca atrayendo a nuevos clientes. En resumen, los clientes felices atraen a nuevos clientes.

## 2.3 La metodología inbound

El Inbound Marketing está basado en la denominada **metodología inbound** [19][35][36] que es un proceso que inicialmente estaba dividido en cuatro fases (atraer, convertir, cerrar y fidelizar) cuyo fin es convertir a los visitantes en clientes para que, posteriormente, pasen a ser promotores de tu producto

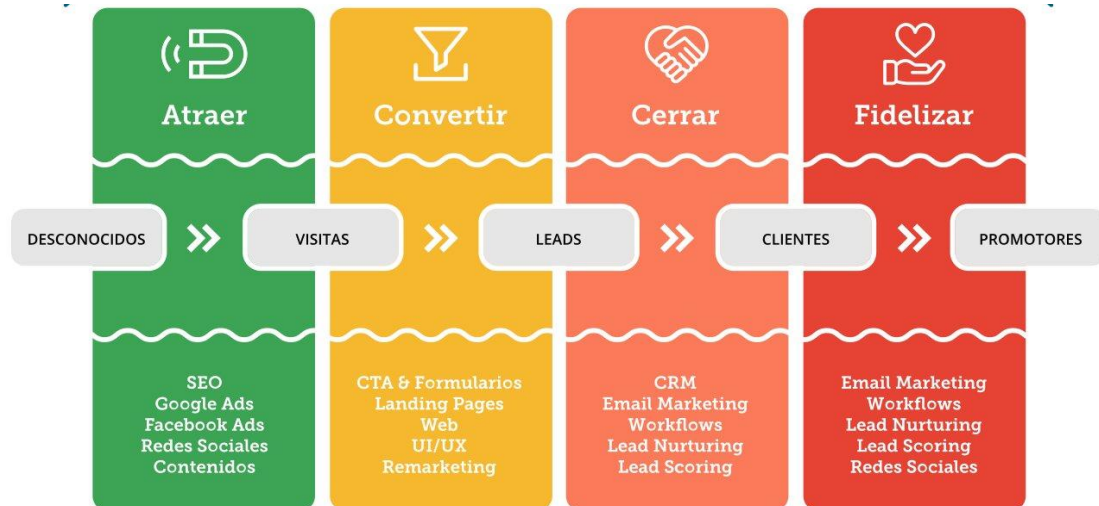


Gráfico que representa las cuatro etapas originales de la metodología inbound. (Fuente: wanaleads.com)

Según esta idea, el proceso tenía un principio (el consumidor conoce nuestra empresa) y un final (el consumidor se convierte en un cliente fidelizado). Pero recientemente la empresa HubSpot (los creadores de este método) ideó un nuevo sistema en el que la relación de la empresa con el cliente no finaliza cuando este se fideliza, sino que se mantiene en el tiempo de forma cíclica, es decir, que nunca termina. Esto es lo que se conoce como **volante** o flywheel, en el que el proceso se divide ahora en tres fases: atraer, involucrar y deleitar. [37]

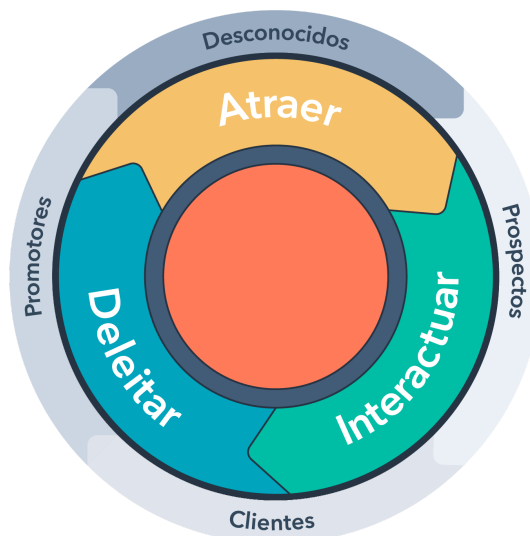


Gráfico que representa el volante o flywheel ideado por la empresa HubSpot. (Fuente: www.hubspot.com)

Pasemos ahora a describir cada uno de las etapas de este proceso:

### 2.3.1 Atraer

En esta primera etapa los consumidores entran en el ciclo por primera vez, lo cual significa que nuestra empresa les resulta completamente desconocida, por lo que nuestro objetivo es conseguir



atraer a las personas adecuadas presentándoles nuestro producto en el lugar y momento apropiados, esto es cuando lo buscan o tienen una necesidad que nuestro producto puede satisfacer.

### Crear un prototipo de cliente ideal o buyer persona

Para este proceso de encontrar cuál es nuestro tipo de cliente objetivo resulta de gran utilidad inventarnos un prototipo de cliente ideal, creando una historia de fondo, así como otras características relevantes como pueden ser edad, trabajo, sexo, nivel de salario, etc. Debemos indicar también cuáles son sus aspiraciones a corto y largo plazo, así como su forma de pensar o sus comportamientos regulares.

## Antonio Rodríguez

Desarrollador web, 29 años

**Datos demográficos**

Lugar de residencia: Madrid

Sector laboral: Desarrollo web

Profesión: Desarrollador web back-end

Nivel de responsabilidad: empleado

Nivel de estudios: Universitario



<b><u>Le gustaría</u></b>	<b><u>No le gustaría</u></b>	<b><u>Qué necesita</u></b>	<b><u>Con qué fin</u></b>	<b><u>Gustos y aficiones</u></b>
Poder comprar un producto que le ayude a mejorar su nivel de vida y ahorrar tiempo, todo sin gastar demasiado	Tener que ir a una tienda física, si puede comprarlo online.	No tener que dedicar tanto tiempo a la limpieza de la casa.	Poder compaginar mejor el trabajo con la vida personal para tener mas tiempo de descanso.	Le gustan los animales, salir con sus amigos y las últimas tecnologías.

Ejemplo de buyer persona para vender un robot aspirador.

Ahora que conocemos cuál es nuestro tipo de cliente objetivo, podemos idear una estrategia para poder atraerlos, mostrándoles el contenido adecuado en el momento idóneo.

Aunque hayamos creado un cliente ideal del que conocemos sus necesidades y preocupaciones, existen ciertos aspectos en relación con nuestro producto que deben estar presentes a la hora de promocionarlo. Y es que debe transmitir sensación de calidad, para destacar entre otros productos similares; debe aportar valor a los clientes y renovarlo periódicamente para no perder el interés de los consumidores. Todo esto ayuda a generar confianza en el cliente y que este se sienta atraído hacia nuestro producto.

### Optimización para motores de búsqueda o SEO

Search Engine Optimization [39] por sus siglas en inglés, se centra en mejorar el posicionamiento orgánico en los motores de búsqueda de internet como pueden ser Google o Bing. Los algoritmos de estos sitios están diseñados para leer el código y todo el contenido de los sitios web y a través de una serie de algoritmos determinar su posición en los resultados de búsqueda en función de su relevancia.

Cuando hablamos de resultados orgánicos nos referimos a aquellos por lo que no hemos pagado a la plataforma de búsqueda para salir en los primeros resultados. A esta otra táctica se la conoce como SEM o Search Engine Marketing [45] que consiste en pagar a servicios como Google Ads<sup>6</sup> para que pongan la web de la empresa que ha pagado al principio de los resultados de búsqueda que contienen determinadas palabras clave, independientemente de que no sean los resultados más relevantes. Las empresas pagan a los buscadores en función del número de personas que accedan a esos enlaces, es lo que se conoce como pago por clic (PPC).

Existen una serie de prácticas que ayudan a mejorar el SEO de un determinado sitio web [85], como pueden ser:

- **Optimización de las URL<sup>7</sup>:** las URL deben ser lo más cortas y simples que sea posible. Además, deben contener alguna palabra clave que se refiere al contenido de dicha página web. Un ejemplo de una página que hablase sobre las mejores plataformas para ver películas en streaming sería:

```
https://www.misitioweb.com/cine/mejores-plataformas-series-peliculas-netflix-amazon-prime-hbo-max-disney-apple-tv.html
```

- **Cuidar la estructura HTML<sup>8</sup> de nuestra página:** Los algoritmos de los buscadores tienen muy en cuenta el contenido que se encuentra dentro de las etiquetas de encabezado (H1, H2, H3, etc.). Por lo que debemos asegurarnos de que estos contengan palabras claves que ayuden a posicionar nuestra web en los resultados de búsqueda.
- **Mejora el SEO técnico [38]:** Los buscadores tienden a mostrar menos aquellos sitios que tienen tiempos de carga lentos o rechazan las peticiones y en general aquellos factores que lastran la experiencia de usuario. Por lo tanto, conviene llevar a cabo una serie de optimizaciones en nuestro sitio web, como utilizar SSL<sup>9</sup>, tener un diseño adaptado a dispositivos móviles, reducir los tiempos de carga o crear un mapa de nuestro sitio web en formato XML<sup>10</sup>, lo cual ayudará al buscador a conocer la estructura de este.
- **Crear contenido de calidad:** A final de cuentas lo más importante es tener un contenido que atraiga a los usuarios, ya que esto hará que sea más relevante para el algoritmo de Google y esto determinará que esté mejor posicionado. Por lo que nuestra página debe tener un contenido bien estructurado, de fácil lectura y que aporten respuestas a lo que el usuario busca.

---

<sup>6</sup> Google Ads es un servicio de publicidad de Google para empresas que desean mostrar anuncios en los resultados de búsqueda de Google y su red publicitaria. [10]

<sup>7</sup> Una URL (Localizador uniforme de recursos) es un identificador único que se utiliza para localizar un recurso en Internet. También se conoce como dirección web. Las direcciones URL constan de varias partes, incluido un protocolo y un nombre de dominio, que le indican al navegador web cómo y dónde recuperar un recurso. [11]

<sup>8</sup> HTML (Hyper Text Markup Language ) es un lenguaje que se utiliza para estructurar una página web y su contenido a través de etiquetas (lenguaje de marcado).

<sup>9</sup> SSL (Secure Sockets Layer) es la tecnología estándar para mantener segura una conexión a Internet y proteger los datos confidenciales que se envían entre dos sistemas, evitando que los delincuentes lean y modifiquen cualquier información transferida, incluidos los posibles datos personales. [12]

<sup>10</sup> XML (Lenguaje de marcado extensible) es un lenguaje de marcado similar a HTML, pero sin etiquetas predefinidas para usar. En su lugar, se definen etiquetas personalizadas diseñadas específicamente para una necesidad concreta. [13]

## Conectar a través de las redes sociales

Ya hemos hablado sobre **cómo** anunciar nuestro producto para atraer a posibles clientes objetivo, pero queda otro elemento indispensable a tener en cuenta como es el **cuándo** y el **dónde**. Tan importante es una cosa como la otra, ya que de nada sirve producir el contenido perfecto para ese cliente ideal si luego no conseguimos hacérselo llegar o lo hacemos en un momento en el que no lo necesita. Por esto es importante conocer qué páginas web o redes sociales frecuenta para saber dónde debemos mostrar nuestro producto, así como el tipo de contenido que busca, el cual puede darnos información valiosa acerca de cuáles son sus necesidades en un momento determinado.

Atraer a los potenciales clientes de las redes sociales resulta una de las estrategias más eficaces y rentables. Si nuestro público objetivo se encuentra activo en las redes sociales, promocionar nuestra marca poniendo publicaciones en estas redes puede tener muy buenos resultados. Para lograr atraer al público objetivo necesitamos crear contenido relevante para los usuarios y que llame su atención.

Las redes sociales también nos ayudan a construir nuestra imagen de marca que dé la sensación de que tenemos una autoridad dentro del sector y que somos una empresa confiable, mostrando información creíble, además de útil y que se muestra cercana a los consumidores interactuando con ellos a través de las redes.

De esta forma conseguiremos que a través del contenido publicado en las redes sociales cada vez más gente conozca nuestra marca y tengan una buena referencia de ésta, todo ello con unos costes muy bajos en comparación con otros métodos de marketing.

### 2.3.2 Involucrar

Una vez hemos conseguido atraer a nuestro cliente objetivo hasta nuestra empresa, tenemos que lograr establecer una relación continua mientras dure el proceso de compra del cliente. Para ello debemos conseguir su información de contacto y almacenarla en una base de datos para poder mantener una vía de comunicación entre ambas partes. Esto nos permitirá crear relaciones duraderas y significativas en lugar de reducir dicha relación cliente-empresa a un mero trámite de venta para después cortar cualquier forma de comunicación de forma definitiva.

#### Generación de leads

Esta fase se basa en la generación de **leads**, es decir, de encontrar candidatos reales que estén interesados en nuestro producto, consiguiendo que confíen en nuestra marca, de forma que nos envíen un formulario con sus datos de contacto para poder iniciar una comunicación directa con el consumidor. Estos leads en esencia son clientes potenciales, y se consideran oportunidades reales de conseguir una venta, por lo que tienen un gran valor para una empresa.

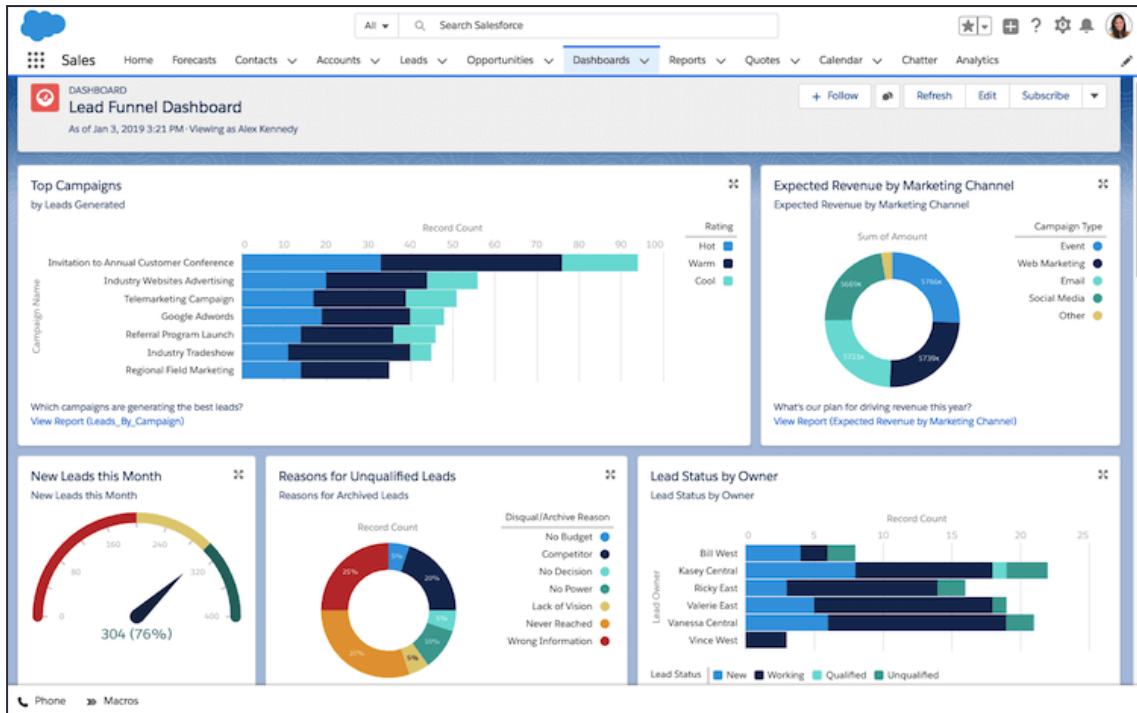
Parece claro que nadie está dispuesto a dar sus datos personales a cambio de nada, por lo que para conseguir que un visitante de nuestra página se convierta en un lead es importante ofrecer algo de valor a cambio. Esto puede ser contenido exclusivo a cambio de una suscripción en nuestra web, suscripción a un boletín de noticias de interés o a un blog, etc. Más adelante profundizaremos más en los leads y en su obtención debido a su importancia dentro de este proceso.



Gráfico que representa la generación de leads.

### Los CRM como herramienta para gestionar los leads

Una vez obtenida la información a partir de los leads generados, es fundamental almacenar estos de manera adecuada dentro de un sistema **CRM** (Gestión de Relaciones con el Cliente o Customer Relationship Management por sus siglas en inglés) [40]. Cuando hablamos de CRM nos referimos a un software de base de datos (normalmente basado en la nube) donde se almacena toda la información de los clientes potenciales y actuales interesados en nuestra marca y que hemos obtenido a través de los leads. La idea es tener toda esa información centralizada en un mismo lugar para poder acceder a ella fácilmente. Esto nos ayuda a poder comprender mejor el comportamiento del cliente y a simplificar el proceso de venta. Un ejemplo de CRM muy popular sería Salesforce.



Ejemplo de interfaz de usuario de Salesforce. (Fuente: [www.salesforce.com](http://www.salesforce.com))

Entre las ventajas que aporta un CRM destacan las siguientes [41]:

- **Mejor calidad de servicio al cliente:** la función principal de un CRM es la de mejorar la relación entre la empresa y los clientes. Este es capaz de gestionar todos nuestros contactos y almacenar toda la información relevante acerca de los clientes, como pueden ser sus datos demográficos (edad, nombre, sexo, etc.), su historial de compras, o cualquier interacción que haya tenido anteriormente con la empresa, y hace que toda esta información sea fácilmente accesible para cualquier empleado que quiera consultarla. De esta forma se puede garantizar una estrategia de ventas más personalizada para cada cliente, obteniendo así una experiencia más satisfactoria.
- **Incremento de los beneficios:** lógicamente el objetivo final de todo este proceso es una mayor obtención de beneficios, y un CRM puede ser muy útil en este sentido, ya que automatiza ciertas tareas importantes y permite tener toda la información en un lugar centralizado, lo que optimiza mucho todo el proceso de ventas.
- **Fidelización del cliente:** Una vez que hemos convertido un lead en un cliente de facto, es necesario retenerlos como clientes recurrentes para conseguir aumentar las ventas. Aquí entra la utilidad del CRM, ya que, basándose en los datos disponibles y a un seguimiento continuo de la actividad y de los intereses del cliente, podremos detectar cuáles son sus necesidades y así satisfacerlas lo antes posible.
- **Gestión de comunicaciones con potenciales clientes:** saber cómo manejar los leads y el momento idóneo para comunicarse con un potencial cliente puede ser una tarea compleja para una persona. Sin embargo, un CRM dispone de toda la información necesaria y puede automatizar este proceso, avisando cuando es el momento idóneo para comunicarse con un potencial cliente para, de esa forma, conseguir una venta con mayor probabilidad. Del mismo modo, nos ayudará a segmentar nuestros contactos en distintos grupos según sus preferencias para poder ofrecer una comunicación más personalizada.

- **Previsión de ventas:** nos da la posibilidad de conocer cómo de cerca está un prospecto de llegar a realizar una compra basándonos en la información obtenida.

### Lead scoring y lead nurturing

Una vez que tenemos toda la información de nuestros potenciales clientes bien almacenada y gestionada, existen otras formas de involucrar que podemos llevar a cabo:

- **Lead scoring** [42]: traducido sería “puntuación de los potenciales clientes” es un sistema que nos permite asignar una puntuación a cada potencial cliente para saber cómo de probable es que dicha persona acabe comprando nuestro producto. De esta forma, si en una determinada campaña de marketing recibimos una cantidad muy grande de leads podremos saber cuáles de ellos están realmente interesados en comprar nuestro producto y cuáles únicamente querían obtener algún beneficio a cambio de enviar el formulario, pero no quieren comprar nada realmente. Esto nos permite centrar nuestros esfuerzos en el primer grupo, lo que aumenta la eficiencia de nuestra inversión.



Ejemplo de modelo de lead scoring.

- **Lead nurturing** [43]: se traduciría como “nutrición de potenciales clientes” y es el proceso por el cual se desarrollan y refuerzan las relaciones con nuestros contactos, independientemente del momento del proceso de conversión<sup>11</sup> en el que se encuentren, preparándose y acompañándolos durante todo el ciclo de compra, proporcionándoles información relevante de nuestra marca regularmente para llamar su atención y que finalmente decidan comprar nuestro producto. Por poner un ejemplo, imaginemos que una persona quiere contratar un determinado servicio, por ejemplo, suscribirse a una plataforma de vídeo en streaming. Esta persona encuentra nuestra empresa que ofrece dicho servicio y deja un lead con sus datos, pero aún no se ha decidido a comprarlo. Lo que conseguimos con el lead nurturing es allanar el camino para que esa persona decida finalmente adquirir nuestro servicio pasando a ser nuestro cliente. Algunas herramientas que pueden ayudarnos en esta labor son la automatización del marketing (por ejemplo, enviando correos personalizados de forma automática) o aplicando descuentos y promociones para dar un último empujón a aquellos consumidores que estén al final del proceso y se decidan a realizar la compra para no perder esa oportunidad.

### 2.3.3 Deleitar

Una vez realizada la venta nuestro trabajo no termina, sino que es necesario seguir empleando recursos para que los clientes estén completamente satisfechos con los servicios ofrecidos, para de esta forma conseguir mantener su fidelidad y confianza en nuestra marca y que vuelvan a elegirnos a

<sup>11</sup> En el marketing, una conversión se da cuando el destinatario de un mensaje de marketing (por ejemplo, un visitante de nuestro sitio web) realiza una acción deseada (como completar un formulario o realizar una compra).

la hora de hacer una compra futura. Retener a los clientes tiene un costo muy inferior al de crear nuevos clientes desde cero, por lo que resulta muy beneficioso para la empresa.

Los clientes fidelizados son una gran oportunidad de crecimiento, ya que no solo están dispuestos a volver a comprar nuestro producto o servicio, sino que también es muy probable que recomienden tu marca a amigos y conocidos, por lo que tu exposición crecerá rápidamente. Lograr convertir un cliente en un entusiasta promotor de tu marca es uno de los principales beneficios del Inbound Marketing.

### Conocer el nivel de satisfacción de los clientes

Pero para poder saber cuál es el nivel de satisfacción de los clientes, es necesario obtener algún tipo de feedback de estos. Es decir, necesitamos recibir comentarios de estos a través de encuestas para saber qué es lo que realmente funciona y qué es lo que no les convence. Debemos dar respuesta a algunas cuestiones como ¿Cuál es su nivel de satisfacción con el producto?, ¿Cómo de decisivo fue el contenido que le ofreciste para su decisión de comprar tu producto?, ¿Hay algo que debería cambiar o añadir?, o ¿Con qué probabilidad recomendaría la marca a un familiar o amigo?

Para saber cuál es el estado de fidelización en el que se encuentran nuestros clientes resulta de gran ayuda contar con algunas métricas que puntúen su nivel de satisfacción:

- **NPS:** Net Promoter Score (por sus siglas en inglés) es la forma más típica de medir el nivel de satisfacción de un cliente. Se trata de que puntúe del 1 al 10 **cómo de probable es que recomiende el producto o servicio a otras personas**. Significando 10 que lo recomendaría seguro y 1 que no lo recomendaría.
- **CSAT:** que significa Customer Satisfaction Score por sus siglas en inglés o en español puntuación de satisfacción del cliente o también índice de satisfacción del cliente. Esta métrica adopta una perspectiva más directa y le pregunta al consumidor del 1 al 5 **cuál es su nivel de satisfacción con el producto o servicio recibido**.

Obteniendo estas métricas de manera asidua podemos obtener una mejor comprensión del nivel de satisfacción de los clientes y ver que cosas se tienen margen de mejora.

Estos datos nos permiten conocer en qué situación se encuentra la relación de nuestra empresa con cada uno de nuestros clientes y de esa forma poder reaccionar a tiempo en caso de que esta empiece a decaer.

Haciendo uso de estas métricas también podemos saber qué cantidad aproximada de clientes vamos a retener, lo que ayuda a realizar una previsión de ingresos con una mayor precisión.

Una vez conozcamos el nivel de satisfacción de nuestros clientes, podremos saber en qué situación se encuentra su relación con nuestra marca y de esa forma corregir nuestra estrategia de marketing.

## Herramientas para la fidelización de clientes

Si conseguimos mantener a los consumidores contentos después de efectuar una compra, lograremos mejorar la tasa de retención de clientes<sup>12</sup> y de esa forma aumentar las ventas y abrir la posibilidad de realizar ventas cruzadas<sup>13</sup>.

Algunos medios que podemos usar en esta fase de deleite para fidelizar clientes son [44]:

- **A través del correo electrónico:** El correo electrónico sigue siendo una de las herramientas más útiles de cara a comunicarnos con nuestros clientes, ya que es usado por una amplia mayoría de la gente. Puedes enviar un correo personalizado de agradecimiento por poner su confianza en ti al haber comprado tu producto. También puedes enviar ofertas especiales por su fidelidad hacia tu marca o algún otro tipo de recompensa que les aporte algún valor, como una invitación a algún evento o tu newsletter.
- **A través de redes sociales:** Las redes sociales son una excelente vía para mantener el contacto con los clientes, esto se debe a que la gente pasa una gran parte del tiempo navegando en las redes sociales, por lo que es muy probable que después de tener una buena experiencia con nuestra empresa dejen algún post o comentario positivo hacia nuestra marca, lo que hace que otras personas las vean y conozcan también nuestro producto. Al tener presencia activa en las redes sociales podremos interactuar con nuestros clientes, aumentando así su buena opinión de nuestra marca, ya que nos verán como alguien más cercano.

Otra manera de deleitar a través de las redes sociales es a partir del servicio de atención al cliente. Los clientes quieren poder contactar con las empresas de la forma más rápida y cómoda posible y por ello muchas personas optan por hacerlo directamente a través de las redes sociales. De esta forma, los clientes dejarán preguntas o comentarios en nuestro perfil que deberemos responder de forma rápida y precisa para dar una sensación de estar siempre preocupados por lo que nuestros clientes opinan o preguntan y nos dará una imagen de responsabilidad y compromiso con nuestra labor. De esta forma también animaremos a los clientes a hablar bien de nuestra marca y otras personas verán que respondes de manera rápida y profesional y te tendrás más posibilidades de que se conviertan también en tus clientes.

Otra de las razones por las que las redes sociales tienen una gran importancia es atender a los comentarios de los usuarios y ver si hay opiniones críticas, lo que nos podría ayudar a encontrar fallos en nuestro servicio y de esa forma darles solución para mejorar la experiencia de nuestros clientes, aumentando su fidelidad.

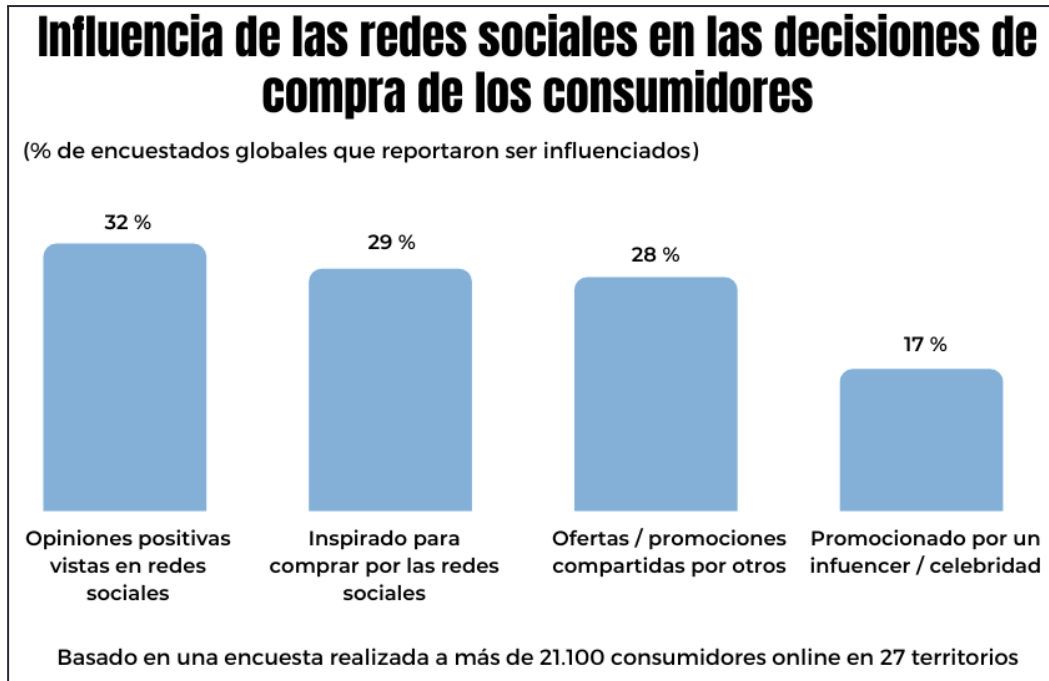
Por último, resulta de gran importancia publicar regularmente contenido que aporte valor para los clientes, ya que esto hará que no pierdan el interés en tu marca y mejore su percepción de esta porque la verán como una empresa activa y dinámica.

---

<sup>12</sup> La tasa de retención de clientes (Customer Retention Rate) es un porcentaje que mide el nivel de fidelidad de los clientes hacia un determinado negocio durante un intervalo de tiempo concreto. Cuanto mayor sea dicha tasa, mayor será la capacidad de una empresa para retener a sus clientes durante un tiempo determinado. [14]

<sup>13</sup> Las **ventas cruzadas** (o cross-selling en inglés) se refiere a la venta de productos relacionados o complementarios a un cliente que ya ha comprado otro de nuestros productos. Es uno de los métodos de marketing más efectivos. Un ejemplo sería cuando contratamos un servicio de telefonía y fibra que además nos ofrece por un importe extra un servicio de vídeo en streaming o de canales de fútbol. No confundir con el up-selling que consiste en añadir una mejora del servicio. En el ejemplo anterior sería ofrecer una red de fibra de 600Mb en lugar de una de 300Mb a cambio de un aumento del precio.





El porcentaje de consumidores de todo el mundo que afirman que sus decisiones de compra se vieron influenciadas por críticas positivas leídas en las redes sociales, ofertas y promociones compartidas por otros usuarios o verlo promocionado por influencers y celebridades.

(Fuente: marketingcharts.com, Abril 2019)

En resumen, con el nuevo modelo del flywheel o volante llegó un cambio de paradigma por el cual se pasó de ver el modelo de ventas como un embudo en el que los consumidores entran y salen a verlo como una rueda en la los clientes una vez entran se mantiene dentro “girando” durante un tiempo indefinido. Al tratarse de un método centrado en el cliente, te asegurarás de brindarle una experiencia de calidad que seguro terminará por recomendar a otros.

## 2.4 Inbound Marketing frente al Outbound

En este apartado vamos a describir cuáles son las principales diferencias entre el Outbound Marketing y el Inbound Marketing. Además de reforzar con datos reales, la afirmación de que la metodología inbound es más óptima de cara a obtener mejores resultados en una campaña de marketing.

### Principales diferencias

Ya hemos descrito en los apartados anteriores con detalle en qué consisten el Outbound y el Inbound Marketing, así como cuáles son sus principales características. Ahora, a modo de resumen, pasaremos a compararlos punto por punto para ver cuáles son sus diferencias más significativas [46][47][48]:

<b>Inbound</b>	<b>Outbound</b>
Centrado en el marketing de contenidos <sup>14</sup> aportando contenido adicional de valor para su público objetivo.	No aporta ningún contenido digital de valor para el público objetivo.
Se promociona usando medios para interactuar con su público como redes sociales, seminarios web, newsletters o blogs.	Se promocionan a partir de medios más pasivos como correo spam, anuncios de TV, radio o vallas publicitarias.
Mensajes personalizados para cada tipo de consumidor.	Mensajes genéricos. No tienen en cuenta las preferencias del consumidor.
Fácil de medir el impacto de una campaña a través de software de marketing digital.	Difícil de medir las campañas hechas a partir de medios físicos, como anuncios de TV o carteles publicitarios.
No se enfoca únicamente en las ventas, sino que pone al cliente en el centro, dándole información de valor y comprendiendo sus necesidades, para fidelizar clientes.	Se centra en conseguir el mayor número de ventas.
Existe una interacción mutua entre la empresa y el cliente, lo que hace que este sienta que se le tiene en cuenta.	Comunicación unidireccional de la empresa hacia el cliente.
La campaña se focaliza en llegar al público que realmente pueda estar interesado en el producto, a partir de datos recogidos y analizados previamente.	La campaña llega al público general sin importar si es de su interés o no.
Intenta no interrumpir al cliente, al contrario, lo atrae de una forma más orgánica.	No le importa interrumpir al cliente de manera abrupta, lo que puede resultar molesto.
Las inversiones en campañas resultan más eficientes a nivel de gastos al centrarse en un público objetivo más reducido.	Al referirse a un público más amplio, requiere de mayores costes.

## Datos Inbound vs. Outbound

Todos los años HubSpot crea un informe llamado State of Inbound, donde recopila datos entre miles de especialistas del marketing y ventas con el fin de conocer la evolución de la industria del Inbound Marketing y cuáles son las principales tendencias de marketing entre las empresas.

---

<sup>14</sup> El marketing de contenidos es una estrategia de marketing que se centra en la creación y distribución de contenido relevante para nuestro público objetivo (a través de redes sociales, blogs, páginas web, etc.) para lograr atraer su atención y, finalmente, conseguir que hagan una acción determinada como que envíen un formulario o que realicen una compra.

A continuación se muestran una serie de datos extraídos de estos informes que ponen de manifiesto los mejores resultados que ofrece el Inbound Marketing frente al outbound marketing.

- La mayoría de empresas que aplican técnicas de Inbound Marketing dicen que **obtienen leads de mayor calidad**, esto es, la probabilidad de que un prospecto acabe convirtiéndose en un cliente. A mayor calidad, más probabilidades de obtener una conversión.

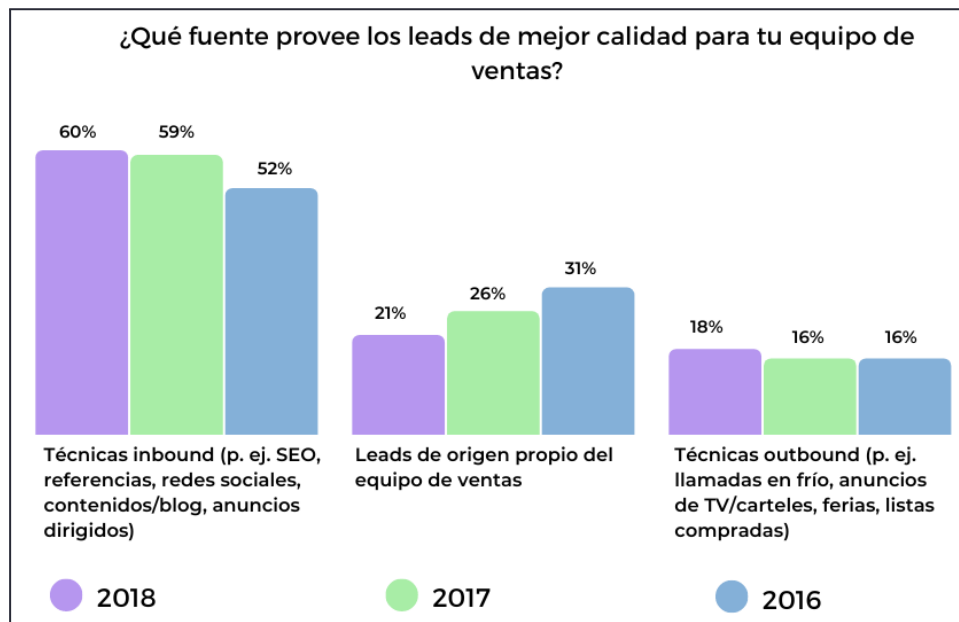


Gráfico que muestra cómo las compañías que usan Inbound Marketing obtienen leads de mayor calidad y cómo esta tendencia va en aumento. (Fuente: HubSpot - State of Inbound 2018 [50])

- Las empresas que aplican el Inbound Marketing por lo general obtienen un **mejor ROI**<sup>15</sup> que las empresas que no lo aplican.



Gráfico que muestra cómo las compañías que utilizan Inbound Marketing obtienen un mejor ROI. (Fuente: HubSpot - State of Inbound 2018 [50])

- En cuanto a la relación **coste por lead generado**, el Inbound Marketing es (de media) un 42,7% más eficiente que el Outbound Marketing, según un estudio realizado por HubSpot entre compañías B2B<sup>16</sup> norteamericanas.

## Coste medio por lead según el tamaño de la empresa

Los inbound leads son más eficientes en coste para las empresas B2B de América del Norte, independientemente de su tamaño.

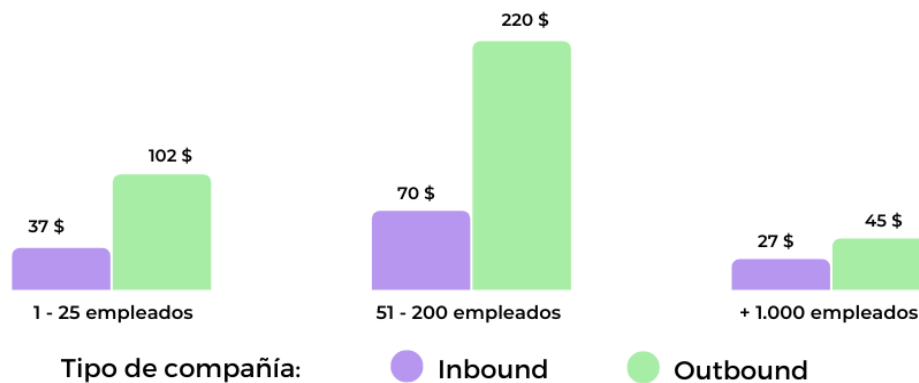


Gráfico de un estudio realizado a compañías B2B norteamericanas que muestra el coste por lead generado. (Fuente: HubSpot - State of Inbound 2014 [49])

<sup>15</sup> El **ROI** (Return on Investment) es una forma de medir el retorno de beneficios que obtenemos de la inversión realizada en una campaña de marketing.

<sup>16</sup> Una empresa B2B (Business-to-business) es aquella que vende su producto directamente a otras empresas en lugar de a un cliente individual.

## 2.5 Generación de leads

El proceso de generación de leads [51][52] resulta clave en el Inbound Marketing, ya que es el punto de unión entre un potencial cliente y la empresa. Dicho proceso comienza atrayendo nuevos visitantes interesados a nuestro sitio web. Desde ahí redirigiremos al usuario a una página destino (también llamada landing page) cuya función es generar nuevos contactos a través del envío de formularios a cambio de ofrecer a los usuarios algo de valor como un ebook con información de su interés, una prueba gratuita o algo similar.

Una vez el usuario ha rellenado un formulario con sus datos de contacto, estos serán incorporados a una base de datos. Acto seguido, dicho usuario será redirigido a una página de agradecimiento y recibirá su contenido. De esa forma pasará a ser un lead. Procederemos ahora a desarrollar este concepto.

### 2.5.1 ¿Qué es un lead?

La definición de lead según HubSpot es: “Un lead es un usuario que se encuentra en la fase inicial del ciclo de compra. El término hace referencia a aquellos contactos que dejan sus datos a través de un formulario a cambio de un contenido de valor. En ese momento pasan a formar parte de la base de datos de tu empresa.” [6]

Un lead, en definitiva, es cualquier persona que muestra interés de alguna forma por un producto o servicio ofrecido por una empresa. En el Inbound Marketing son los leads los que reciben información de la empresa después de haber aceptado iniciar una comunicación de manera voluntaria, normalmente rellenando un formulario con información personal para recibir algo a cambio, ya sea por alguna oferta, para acceder a un periodo de prueba de algún servicio o realizar una suscripción. Esto marca una diferencia notable con el outbound marketing, en el que es la empresa la que empieza la comunicación sin permiso del consumidor enviando correo spam o haciendo llamadas en frío [53].

Pongamos, por ejemplo, que rellenamos una encuesta en una página de video en streaming para recibir recomendaciones de películas que podríamos interesarnos y al rato recibimos un email con la información que pedimos. Esto resulta mucho menos invasivo que recibir una llamada sin previo aviso y sin saber si realmente estamos interesados en recibir recomendaciones de películas.

Esto resulta también beneficioso desde el punto de vista de la empresa que obtiene el contacto de una persona que puede estar interesada en comprar nuestro servicio, evitando perder tiempo y dinero en contactar con personas que no tienen interés alguno en nuestro producto.

### 2.5.2 Valoración de leads o lead scoring

No todos los leads que nos llegan parten de las mismas condiciones. Puede que haya algunos que ya estén dispuestos a comprar nuestro producto y otros que solo quieran informarse por el momento. Por eso es necesario utilizar algún sistema que nos permita saber en qué situación se encuentra cada lead entrante y de esa forma emplear una estrategia de marketing más personalizada. Aquí es donde entra en juego la valoración de leads o lead scoring. El lead scoring es un método que consiste en asignar a cada lead una puntuación en función de diversos factores, de tal forma que nos permita poder centrarnos en aquellos que tengan más probabilidades de convertirse en clientes nuestros. Normalmente, se distinguen tres tipos de leads en función de su puntuación [54][55][56]:

- **Lead frío:** se refiere a una persona que, aunque presenta un interés por la marca y ha dejado sus datos, apenas está comenzando a conocer nuestro producto y, por tanto, no está preparado para realizar compra alguna. Está aún en una etapa inicial del proceso, ya que apenas ha comenzado a investigar cómo dar solución a una determinada necesidad o problema, pero igualmente tiene cierta predisposición a ser persuadido.

- **Lead calificado para marketing:** Marketing Qualified Lead (MQL por sus siglas en inglés), también llamado lead templado. Es una persona que ya ha mostrado interés por la marca en varias ocasiones y de la que tenemos información más concreta y detallada. Se encuentran en un punto intermedio del proceso de conversión, porque están interesados en el servicio que ofrecemos, pero aún no están preparados para comprar. Se considera que ya forma parte de nuestro público objetivo y que, por tanto, tiene una alta probabilidad de convertirse en nuestro cliente. Dicha persona ya ha pasado el proceso de conocer nuestra empresa y ya nos valora como una opción a tener en cuenta. Es el momento de enviar estos leads al equipo de marketing, el cual se encargará de enviarles mensajes y promociones para acercarlos más al final del proceso.
- **Lead calificado para ventas:** Sales Qualified Lead (SQL por sus siglas en inglés), también se le denomina lead caliente. Se trata de un contacto que ya ha pasado todas las fases anteriores y ha dejado claro su interés por convertirse en un cliente, por lo que está en el punto adecuado para recibir una llamada telefónica y poder cerrar una venta.

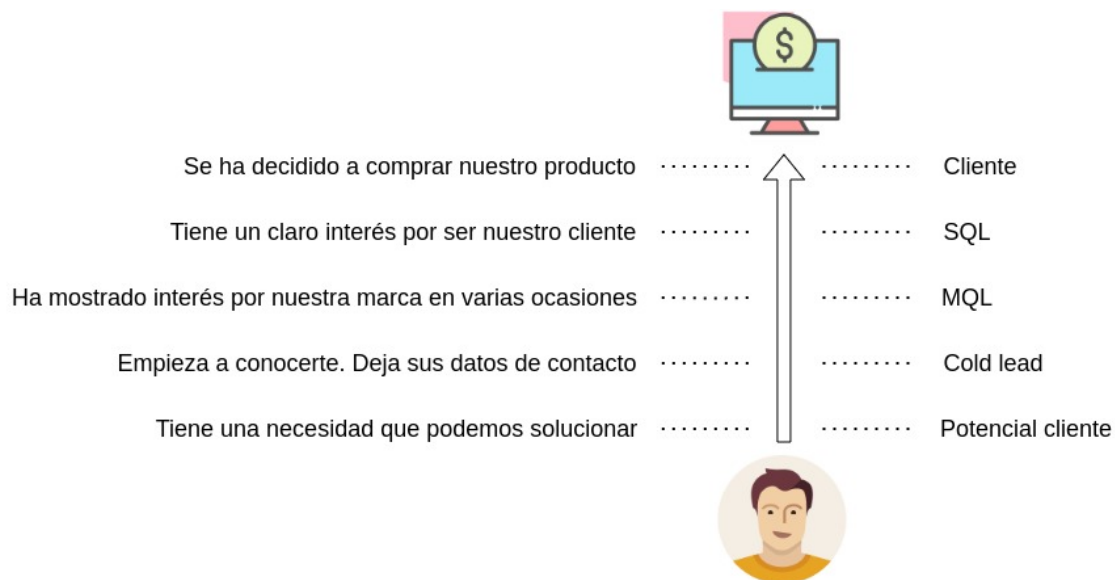


Gráfico que representa los distintos estados por los que pasa la valoración de los leads.

### 2.5.3 El proceso de generación de leads

Según HubSpot: “Lead generation o generación de leads es el proceso de atracción y conversión de extraños en alguien que indica interés en el producto o servicio de tu empresa. Algunos ejemplos de generadores de leads son solicitudes de empleo, publicaciones de blog, cupones, eventos en directo y contenido en línea.” [7] En otras palabras, es una manera de acercar a potenciales clientes a nuestro negocio y guiarlos para que finalmente acaben comprando.

La generación de leads forma parte de la segunda etapa del proceso inbound (es decir, el de involucrar) como vimos anteriormente. Esto sucede una vez hemos conseguido atraer a nuestro público objetivo, pasando a ser clientes potenciales concretos con nombre y apellidos, con los cuales podemos contactar y aplicar estrategias de marketing más personalizadas según la situación y las necesidades de cada individuo.

Los pasos que sigue el proceso de generación de leads es el siguiente:

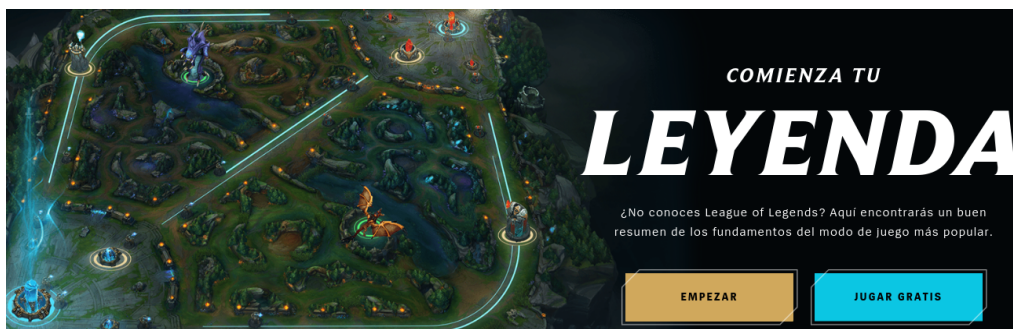
1. Un usuario llega a nuestro sitio web a través de uno de los canales de atracción de los que disponemos, como pueden ser a través de las redes sociales, un blog, o haciendo una consulta en cualquier motor de búsqueda como Google o Bing.
2. Acto seguido, el visitante hace clic en una llamada a la acción o call-to-action (CTA). Este puede ser un botón, una imagen o cualquier elemento que mueva a los visitantes a acceder a un determinado enlace al que nosotros buscamos que entren.
3. Dicho CTA llevará al visitante a una landing page o página destino, que se trata de una página con un formulario que pedirá al usuario cierta información de contacto y algunos otros datos relevantes acerca del visitante. A cambio de dicha información se entregará algo, como puede ser un e-book, acceso a un periodo de prueba de un servicio o a una conferencia online. En definitiva, se trata de ofrecer cualquier cosa que el visitante considere que tiene el suficiente valor como para prestar a cambio su información de contacto.
4. Una vez el usuario ha rellenado el formulario de nuestra landing page ya tenemos un nuevo lead generado que pasará a formar parte de nuestra base de datos de potenciales clientes.

#### 2.5.4 Call-to-action (CTA)

La función principal de un CTA es provocar un estímulo para realizar una acción determinada, como puede ser captar la atención de clientes potenciales y conseguir que dejen un lead en nuestra web.

Para crear un call-to-action que cumpla su función de manera efectiva conviene tener en cuenta algunas condiciones [57]:

- **Contener un verbo y alguna palabra clave:** debe incorporar algún verbo al inicio de la frase que invite al usuario a llevar a cabo dicha acción, además de palabras clave que hagan ver que dicha acción está relacionada con lo que el visitante está buscando. Algunos ejemplos pueden ser “Descarga aquí la versión de prueba”, “Pide más información aquí” o “Haz clic para ver el video completo”.
- **Resaltarlo con algún color que destaque:** Los botones CTA deben destacar y conviene que no tengan un texto demasiado largo. Debemos usar colores que contrasten con el resto del sitio web, así como un diseño atractivo. Para ello conviene tener en cuenta cuáles son los gustos de los usuarios a los que va dirigido nuestra web.
- **Posicionarlo de manera adecuada dentro de la página:** un CTA debe estar siempre colocado de manera estratégica en un lugar visible y muy fácil de localizar, de manera que al entrar en el sitio web nos llame la atención casi al momento. Para ello, el botón debe ser lo suficientemente grande para poder identificarlo rápidamente, pero no tanto como para que resulte molesto y le quite al usuario la atención del resto del contenido.



Dos ejemplos de CTA. Arriba de la plataforma de streaming Disney+ y debajo del videojuego League of Legends (Fuentes: disneyplus.com y leagueoflegends.com)

Los CTA son una herramienta muy importante dentro del proceso inbound, ya que son el punto inicio del proceso de generación de leads, que terminará traduciéndose en ventas.

### 2.5.5 Las landing pages

Según HubSpot “Una landing page, también conocida como página de destino, es una página web que sirve para convertir al visitante en lead por medio de un llamado a la acción (CTA). Puede incluir un formulario y otras formas de captación, de acuerdo con su objetivo particular.” [8]

Se trata de una página web independiente, creada expresamente para una campaña de marketing. Es donde un visitante “aterriza” (de ahí el término landing que se traduce como aterrizar) [60] después de hacer clic en algún CTA de nuestras redes sociales, de algún email que recibió o de nuestra propia página web. Su función es recopilar la información de contacto del visitante a cambio de algo de valor para el usuario. Dicha información se obtiene a partir de un formulario de captura de leads, en el que los visitantes introducen datos personales tales como nombre, edad, lugar de residencia y datos de contacto como el correo electrónico.

La finalidad de una landing page es la de producir leads y enviarlos a una base de datos de clientes potenciales para los que poder aplicar una estrategia de marketing dirigida a esas personas enviando emails personalizados, anuncios pagados u otro tipo de métodos de Inbound Marketing.

Es importante aplicar prácticas de SEO adecuadas a nuestras landing pages, ya que su principal objetivo es el de atraer al mayor número de potenciales clientes posibles de forma orgánica, lo que nos ayudará a reducir costes en publicidad pagada. Algunas prácticas que ayudan a mejorar el SEO de una landing page son [58][59]:



- **Mantener una URL corta y clara** que represente de forma precisa el contenido de la página.
- **Incluir palabras clave** en la página que expresen cuál es la temática de dicha página, de forma que cuando los usuarios busquen esas palabras, nuestra web aparezca en los resultados.
- **Estructurar el contenido de la página adecuadamente:** es necesario que aparte de incluir las palabras claves, la estructura de la página sea clara de forma que un motor de búsqueda sepa qué palabras tienen mayor importancia. La palabra clave principal debe estar en la etiqueta del título. Es recomendable añadir texto alternativo en las imágenes que describan el contenido de esta, ya que el motor de búsqueda no sabrá cuál es su contenido de otra forma. También hay que tener en cuenta la jerarquía de las etiquetas de encabezado de nuestra página, de forma que el encabezado principal (H1) coincida con el título y el resto (H2, H3, etc.) contengan igualmente palabras clave.

- **H1:** El título con la idea principal del post
  - **H2:** Primera sección del post
    - **H3:** Subsección
      - **H4:** Subsección
      - **H4:** Subsección
    - **H3:** Subsección
      - **H4:** Subsección
  - **H2:** Segunda sección del post
    - **H3:** Subsección
      - **H4:** Subsección
      - **H4:** Subsección
      - **H4:** Subsección

Ejemplo de uso de las etiquetas de encabezado, siendo H1 la de mayor importancia y H4 la menos relevante.  
(Fuente: hubspot.com)

- **Optimizar las velocidades de carga:** la velocidad de carga de las páginas es otro factor que tienen en cuenta los motores de búsqueda a la hora de posicionar un determinado sitio web en los resultados de búsqueda. Una alta velocidad de carga mejora la experiencia de usuario y evita que los usuarios se cansen y cierren la página antes de que se cargue. Acciones como reducir el tamaño de las imágenes o el número de peticiones que esta realiza pueden ayudar a disminuir el tiempo de carga.
- **Crear backlinks:** los backlinks son enlaces desde una página de un sitio web que redirigen a otro sitio web. Es importante que consigamos que páginas web conocidas y con prestigio generen backlink que apunten a nuestro sitio web. De igual forma, conviene que pongamos enlaces a otros sitios web que sean de confianza y que estén relacionados con el contenido de nuestra página.
- **Realizar pruebas de A/B testing:** teniendo en cuenta el corto tiempo de vida de las landing pages, resulta el entorno ideal para llevar a cabo pruebas para descubrir qué estrategias dan

mejor resultado. El A/B testing consiste en mostrar diferentes versiones de la misma página, cambiando ciertos elementos clave como pueden ser algunos estilos, colores, textos de nuestros CTA, etc. De esta forma podremos medir qué modificaciones han sido más eficaces a la hora de llamar la atención de los usuarios para conseguir de esa forma optimizar nuestras páginas y lograr un mayor éxito en nuestras campañas de marketing.

**SUSCRÍBETE PARA RECIBIR NOTICIAS  
SOBRE EA SPORTS FIFA**

¿Ya tienes una cuenta EA? Comprueba [tus preferencias de correo electrónico aquí](#).

Suscríbete hoy mismo para recibir las últimas noticias, vídeos, ofertas y mucho más sobre EA SPORTS FIFA por correo electrónico (incluyendo otros productos, noticias, eventos y promociones de EA).

Puedo retirar mi consentimiento en cualquier momento. Para ello, debo cambiar mis [preferencias de correo electrónico](#), ponerme en contacto con [privacyadmin.ea.com](#) o enviar una carta a Electronic Arts Inc., ATTN: Email Opt-Out, 209 Redwood Shores Pkwy, Redwood City, CA, 94065 (EE. UU.).

**Suscribirse**

Ejemplo de landing page para la suscripción al newsletter de EA Sports. (Fuente: ea.com)

## 2.6 Problema a resolver

El objetivo del proyecto es construir un sistema software para facilitar la captura de los **leads** de usuarios con el fin de convertir a estos potenciales clientes en contactos con los que poder mantener una relación a largo plazo. Esto lo haremos generando unas **landing pages**, que son páginas web creadas específicamente para una campaña de marketing donde un visitante llega, normalmente, después de hacer clic en un enlace de un correo electrónico o de algún anuncio y cuyo objetivo principal es que el usuario envíe un formulario con sus datos. Para ello utilizamos la llamada a la acción (o **CTA** por sus siglas en inglés), es decir, el siguiente paso o acción que el vendedor quiere que tome el consumidor. De esta forma, las empresas obtienen información valiosa sobre potenciales clientes para poder ofrecerles los productos que mejor se ajusten a sus necesidades y, de esa forma, incrementar las ventas y la fidelización del cliente. Además, debemos tener en cuenta que, si damos mucha difusión, es posible que se produzcan avalanchas de datos en periodos cortos de tiempo debido al gran número de usuarios que realizan el envío de formularios, por lo que debemos estar preparados para manejar un gran volumen de datos y gestionarlos adecuadamente para evitar que nuestro sistema se vea desbordado. Para ello, los objetivos marcados serán los siguientes:

- Creación de landings que resulten atractivos para el usuario, pero además de forma rápida, ya que los ciclos en las campañas de marketing tienen, por lo general, una duración corta. Para lograr este objetivo deberemos:
  - Partir de un diseño HTML generado por otro departamento (por ejemplo, el de UX/UI<sup>17</sup>).
  - Gestionar altas, bajas y modificaciones de dichas páginas.

<sup>17</sup> UX (experiencia de usuario) y UI (interfaz de usuario) se centran en que la interacción entre los usuarios y los sistemas informáticos sea lo más amigable y amena posible. UI se centra más en el aspecto visual y estético de la interfaz, mientras que UX se enfoca en la usabilidad para que el usuario tenga una fácil navegación.

- Procesar los leads procedentes de las landings de tal forma que el usuario no vea reflejado el tiempo de procesamiento, que puede ser largo, en el tiempo de carga de la página web.
- Facilitar el acceso a los leads por parte de quien vaya a procesarlos:
  - Operadores humanos (haciendo uso de Google Sheets).
  - Herramientas de marketing automáticas como Salesforce Marketing Cloud.
  - Integración con un CRM como p. ej. Salesforce
- Garantizar unas medidas mínimas de seguridad informática:
  - Debido a que los leads recibidos son anónimos, algún usuario malintencionado podría generar un gran número de datos falsos. Lo cual sería claramente contraproducente para nuestra labor de recolección de información. Por lo que es necesario usar alguna herramienta para la autenticación de los leads.
  - Proteger posibles datos sensibles de los usuarios, ya que, lógicamente, a nadie le gustaría ver sus datos personales expuestos al público. Además de que habría una pérdida de la confianza en nuestros servicios por parte del cliente.
- Para finalizar haremos pruebas simulando que hay un número alto de usuarios que consultan las landings y que dejan leads en las mismas.



## 3. Diseño de la solución

Una vez descrito cuál es el problema a resolver, pasemos a explicar qué herramientas hemos utilizado para resolver dicho problema, desarrollando además la forma en la que las hemos implementado dentro del proyecto.

### 3.1 Generación de landing page utilizando Wagtail

En el análisis del problema hablábamos de la necesidad de gestionar altas, bajas y modificaciones de nuestras páginas landing. Para facilitar dicha gestión usaremos **Wagtail**.

Wagtail es un CMS gratuito y de código abierto<sup>18</sup> escrito en Python<sup>19</sup> y basado en el framework<sup>20</sup> de Django. [66]

Puede que la definición anterior contenga varios términos que resulten desconocidos en un principio. Por lo que a continuación vamos a explicar en qué consiste cada uno de ellos.

#### 3.1.1 ¿Qué es un CMS?

Un sistema de administración de contenido (o CMS por sus siglas en inglés) [61][62] es una aplicación software que nos permite crear un sitio web y administrar su contenido sin necesidad de codificarlo desde cero y en algunos casos incluso no es necesario tener conocimiento alguno sobre programación.

Sin un CMS deberíamos diseñar nuestra página desde cero, eso supone que tendríamos primero que generar un archivo html al que agregar texto, una barra de navegación y algunas imágenes. Después habría que añadir algunos estilos con CSS<sup>21</sup> de forma que tome el estilo que mejor se adapte a lo que buscamos. Además, sería recomendable añadir algunas funcionalidades usando scripts. Por último, hay que subir dicho archivo a una base de datos online que sea pública y accesible a través de un navegador.

Luego, si quisiéramos modificar una página, deberíamos modificar directamente el código html de dicha página para después volver a subirla, lo cual tiene el riesgo de que modifiquemos alguna parte del código que afecte a la funcionalidad correcta de nuestro sitio.

Un sistema de gestión de contenido se compone principalmente de dos partes:

- Una aplicación de gestión de contenido (CMA): nos permite añadir contenido a nuestra página y gestionarlo. Todo a través de una interfaz muy intuitiva.
- Una aplicación de entrega de contenido (CDA): Esta parte se encargará de tomar los cambios que hemos realizado en el CMA, codificarlo y almacenarlo. De forma que los visitantes

---

<sup>18</sup> Un software de **código abierto** es aquel cuyo código fuente está abierto al público de tal forma que cualquiera pueda verlo y modificarlo como considere.

<sup>19</sup> **Python** es uno de los lenguajes de programación más populares. Es un lenguaje de propósito general, lo que significa que se puede usar en una gran variedad de aplicaciones. Tiene una sintaxis simple, fácil legibilidad y cuyo objetivo es facilitar la tarea del desarrollador.

<sup>20</sup> Un **framework**, es una plataforma de software diseñada para facilitar el desarrollo de un tipo de aplicaciones específico. Esto agiliza enormemente el trabajo de los desarrolladores, que no deben empezar de cero cada vez que quieran crear una nueva aplicación.

<sup>21</sup> **CSS** o Cascading Style Sheets es un lenguaje que se utiliza principalmente para dar formato a páginas web. Al poder ser reutilizado en distintas páginas nos permite dar una apariencia uniforme a nuestro sitio web

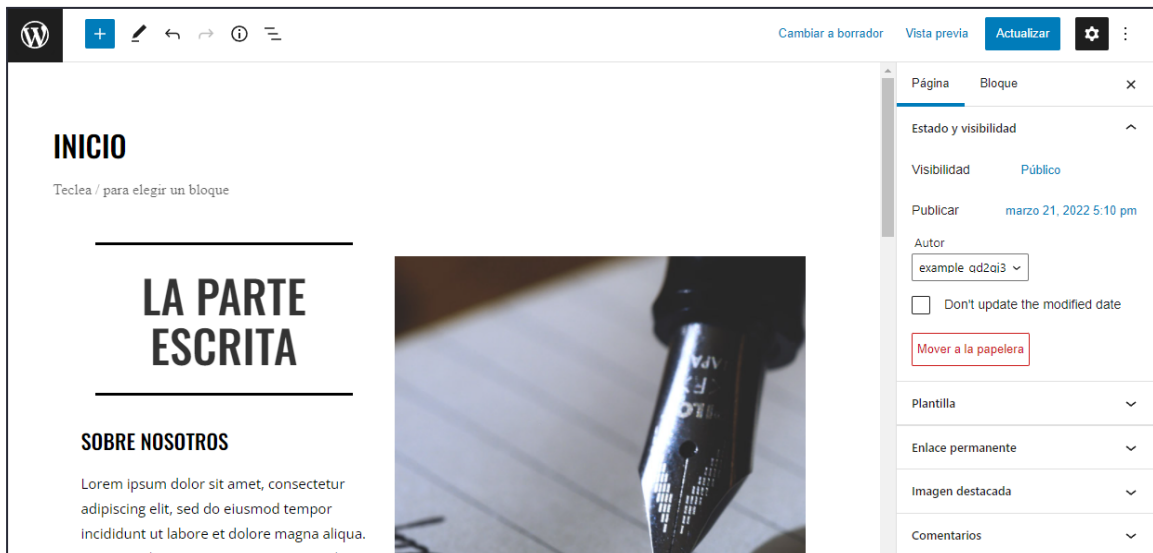
puedan visualizar dicho contenido.

Las principales características que definen a cualquier CMS son:

- Creación de contenido: Permite a los editores generar y gestionar el contenido de su página web de manera fácil e intuitiva.
- Almacenamiento: Poder guardar todo nuestro contenido de forma centralizada y organizada.
- Nos permite generar distintos roles, de forma que cada persona pueda acceder solo a la parte del CMS que le corresponda, evitando de esa forma que de manera accidental o malintencionada modifique algo que esté fuera de sus funciones.
- Publicar nuestro contenido online de forma sencilla.

Algunas de las ventajas de usar un CMS son:

- Uso colaborativo: distintas personas en diferentes lugares pueden acceder al editor iniciando sesión para contribuir en la creación y administración del sitio web.
- No requiere de conocimientos técnicos: permite a las personas poder editar una página web de manera intuitiva sin necesidad de saber lo que pasa por debajo o de tener conocimientos de lenguajes de programación web, un ejemplo de esto sería Wordpress. Aun así, hay algunos CMS que sí dan la posibilidad de editar nuestro panel de control, de manera que las posibilidades de personalización son casi infinitas. Este es por ejemplo el caso de Wagtail.
- Agilizar el proceso de generación de nuevo contenido, ya que no tenemos necesidad de codificar nada cada vez que queramos crear una nueva página y centrarnos simplemente en desarrollar el contenido como tal de nuestra web.



Ejemplo de CMS de Wordpress. (Fuente: help.dreamhost.com)

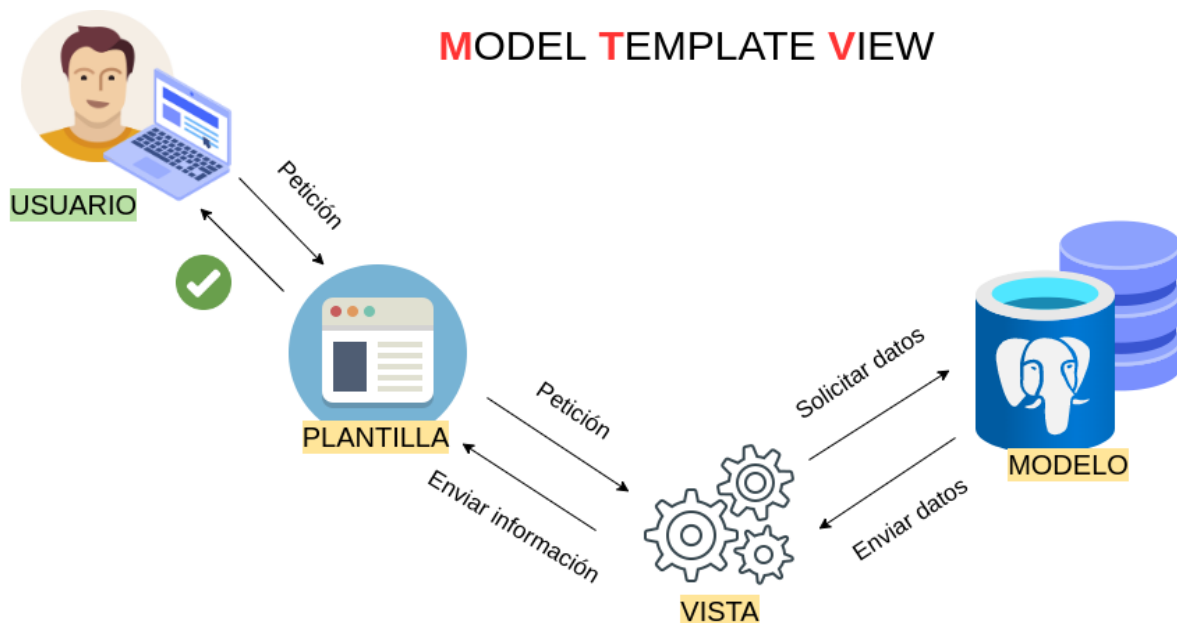
### 3.1.2 ¿Qué es Django?

Django es un framework de desarrollo web de alto nivel escrito en el lenguaje Python y de código abierto que permite crear sitios web seguros y de fácil mantenimiento. Django se aprovecha de una inmensa cantidad de módulos prefabricados, que nos permiten incluir de una manera sencilla funcionalidades avanzadas como autenticación de usuarios, panel de administración, creación de formularios, etc. de forma que solo requiera adaptarlos a nuestra web, ahorrando mucho tiempo en la implementación de estos. [63][64][65]

Django está diseñado para generar sitios web complejos de forma rápida y sencilla. Nos permite reutilizar código en diferentes sitios web.

Está basado en el modelo-plantilla-vista (MTV por sus siglas en inglés) que consiste en dividir cualquier aplicación en tres grandes bloques:

- El **modelo**: se encarga de gestionar los distintos elementos que conforman nuestro sitio web y que están almacenados en una base de datos.
- La **plantilla**: su función es la de mostrar la información al usuario, es decir, es con lo que el usuario interactúa (por ejemplo la interfaz de nuestra página web).
- La **vista**: se encarga de gestionar todas las comunicaciones entre la plantilla y el modelo. Es donde se implementan todas las funcionalidades principales de nuestro sitio web. Su función es almacenar los datos introducidos a través de la interfaz de usuario (plantilla) y enviar a las plantillas los datos necesarios desde la base de datos (modelo).



Representación gráfica del modelo-plantilla-vista.

Gracias a este modelo podemos separar fácilmente la parte de front-end<sup>22</sup> y back-end<sup>23</sup> de la aplicación.

<sup>22</sup> **Front-end** es la parte de la programación web que se encarga de los elementos visuales y de todo con lo que el usuario interactúa de forma directa.

<sup>23</sup> **Back-end** es la parte del sitio web con la que el usuario no puede interactuar. Es la parte que realiza todas las operaciones para que la aplicación funcione correctamente.

Algunas de las características principales de Django son:

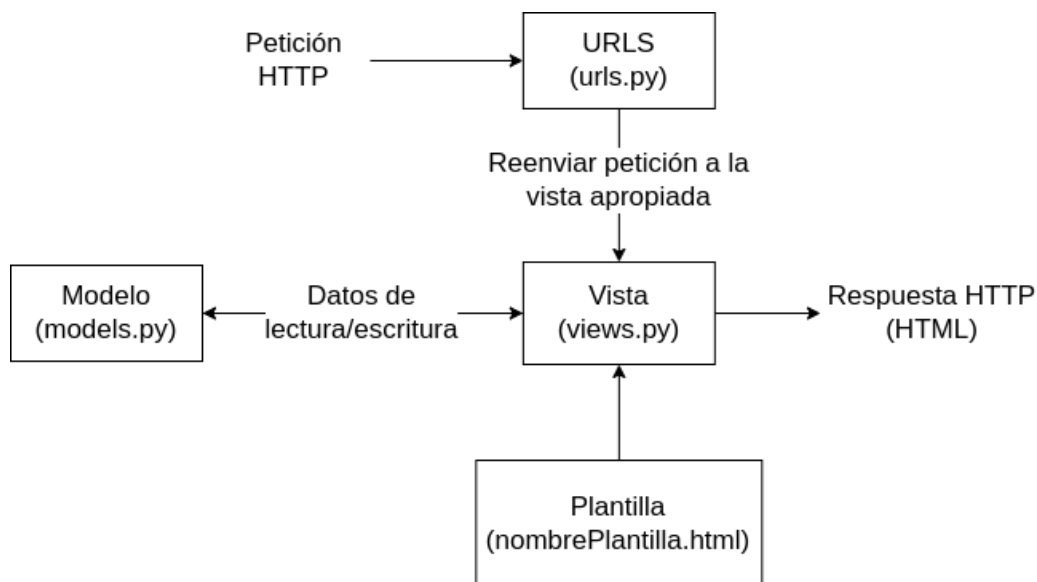
- **Reutilización:** Django sigue la filosofía DRY (Don't Repeat Yourself), lo que significa que los desarrolladores pueden reutilizar parte o todo el código de una aplicación para otro proyecto. Además, al contar con una enorme comunidad detrás, Django tiene un amplio directorio de paquetes prefabricados listos para usar, lo que es muy útil a la hora de aplicar nuevas funcionalidades a nuestra aplicación.
- **Framework completo:** esto significa que Django viene con todos los elementos necesarios para desarrollar una aplicación completa. Desde plantillas html, enrutamiento URL o hasta un sistema de administración de usuarios. Lo que nos evita tener que emplear herramientas de terceros en nuestra aplicación.
- **Escalabilidad:** al no estar fuertemente acoplados, los componentes de una aplicación de Django se pueden quitar y poner cuando sea necesario. Lo que permite una gran flexibilidad a la hora de escalar nuestro proyecto.
- **Actualizado:** tiene una gran comunidad alrededor que constantemente está actualizando, probando y mejorando el framework lo que lo hace más seguro y estable.

El procesamiento de una solicitud al servidor

Cuando llega una solicitud a nuestro servidor web, esta se pasa a Django, que a través del "urlresolver" decide a qué vista corresponde dicha URL y una vez la ha encontrado le envía la petición a dicha vista.

La función de la vista se encarga de realizar las acciones que tenga programadas, como por ejemplo, modificar algún elemento del modelo (base de datos).

Una vez la función de la vista haya efectuado todas las acciones que tuviera que hacer, genera una respuesta que Django envía de vuelta al navegador web del usuario que hizo la solicitud inicial.



Representación gráfica del proceso de petición a un servidor web de Django. (Fuente: developer.mozilla.org)



## 3.2 Aplicación de Wagtail en la solución

Para el diseño de nuestro sitio web vamos a usar Wagtail. Primero comenzaremos explicando cuál fue el proceso de instalación.

### 3.2.1 Instalación de Wagtail

Comenzamos creando un entorno virtual en el directorio donde vayamos a crear la aplicación y lo abrimos para poder comenzar a trabajar en él.

```
python3 -m venv env
source env/bin/activate
```

Donde env sería el nombre del entorno virtual que hemos elegido.

Instalamos wagtail.

```
pip install wagtail
```

Una vez instalado generamos nuestro proyecto.

```
wagtail start mysite
```

Esto creará un nuevo directorio llamado mysite que contendrá todo lo necesario para comenzar a crear nuestro sitio web.

Una vez hecho esto, dentro del directorio mysite ejecutamos los siguientes comandos para iniciar nuestro servidor.

```
pip install -r requirements.txt
./manage.py migrate
./manage.py createsuperuser
./manage.py runserver
```

De esta forma podremos acceder a nuestro servidor a través de la URL <http://localhost:8000> y a la interfaz del administrador desde la dirección <http://localhost:8000/admin/>



## Welcome to your new Wagtail site!

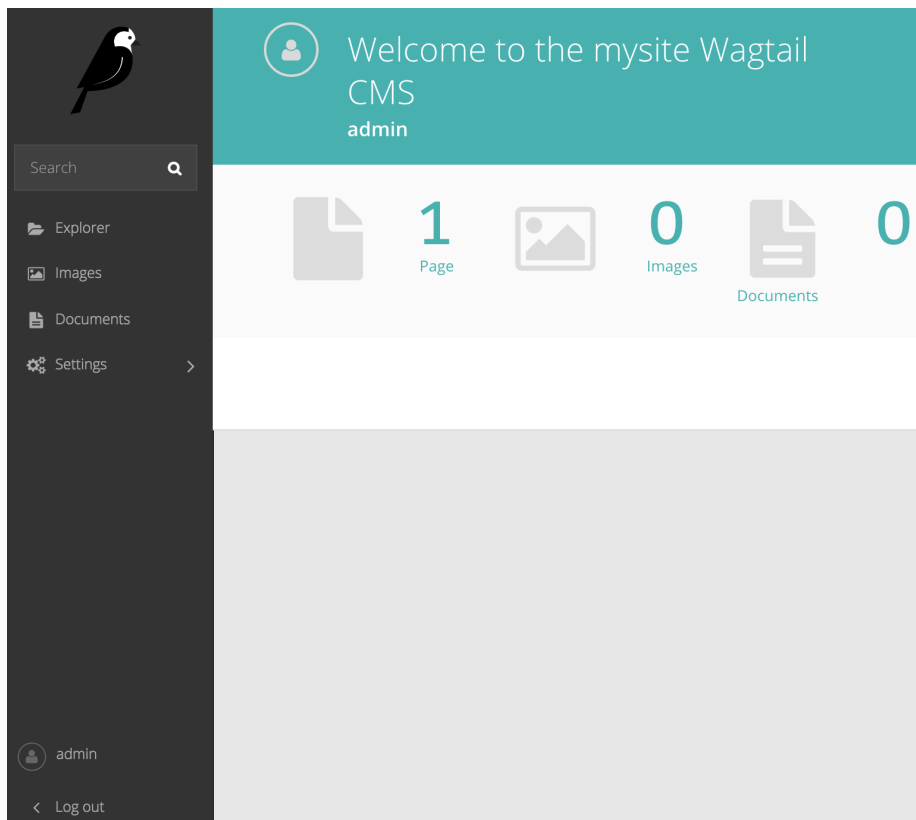
Please feel free to [join our community on Slack](#), or get started with one of the links below.

[Wagtail Documentation](#)  
Topics, references, & how-tos

[Tutorial](#)  
Build your first Wagtail site

[Admin Interface](#)  
Create your superuser first!

Página de bienvenida de Wagtail a través de la url `http://localhost:8000`



Página del administrador de Wagtail a través de la url `http://localhost:8000/admin/`

Creemos una nueva aplicación que contendrá nuestros formularios y otras páginas que creemos:

```
python manage.py startapp blog
```

También debemos añadir la nueva aplicación a la lista de aplicaciones instaladas en el archivo de configuración `mysite/settings/base.py`

### 3.2.2 Creación del index page

Vamos a crear una página que actúe a modo de índice y que contenga un listado de todos los posts y formularios de nuestro sitio web.

En el archivo `blog/models.py` añadimos una nueva clase:

```
class BlogIndexPage(Page):
    intro = RichTextField(blank=True)

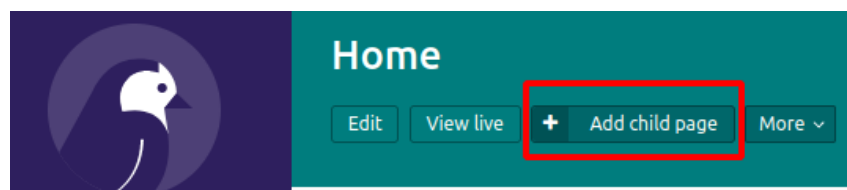
    def get_context(self, request):
        context = super().get_context(request)

        blogpages = self.get_children().live().order_by('-first_published_at')
        context['blogpages'] = blogpages
        return context
```

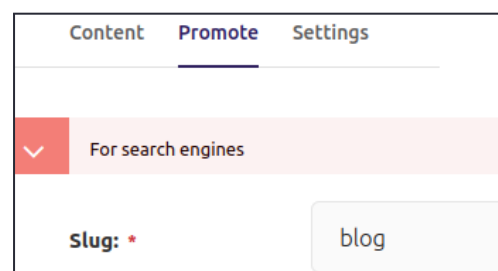
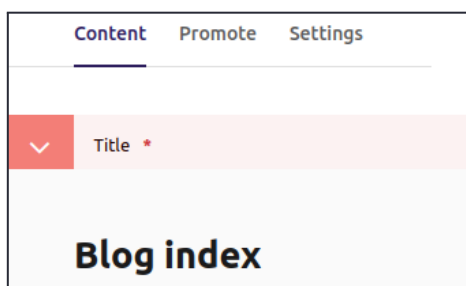
Hemos modificado el método `get_context`, recuperando el contexto original y cambiando una de las consultas del QuerySet que trae por defecto (`blogpages`) por una nueva cuya función es ordenar los resultados por orden cronológico, para que, de esta forma, aparezcan al principio las páginas más recientes.

Ahora debemos crear una plantilla que en Wagtail debe llevar el mismo nombre que la clase que acabamos de crear, en la ruta `blog/templates/blog/blog_index_page.html`

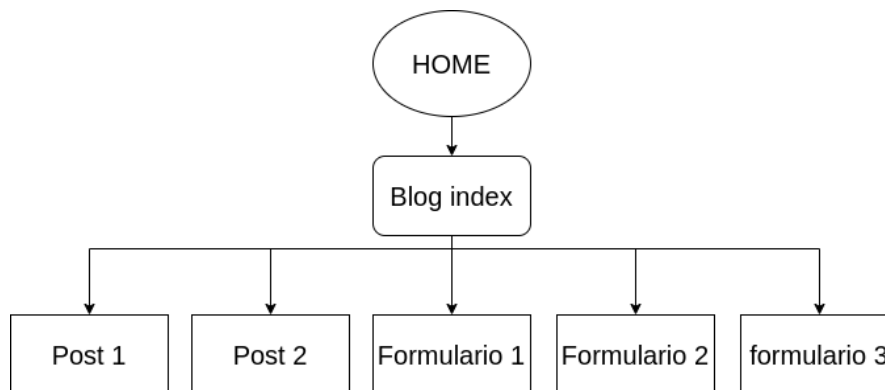
Una vez hecho esto, ya podemos ir a nuestro admin para crear nuestra index page.



Aquí podemos añadir el título y también modificar el nombre que aparecerá en la url desde la pestaña 'promote'. Por defecto, Wagtail pone en promote el texto que hayamos puesto en el campo del título, aunque podemos modificarlo.



De esta forma, una vez hayamos añadido nuevas páginas a nuestro índice, la estructura de nuestro sitio web quedaría de la siguiente forma:



Ejemplo de estructura de nuestro sitio web en el que los formularios y otros posts cuelgan de la página del índice y esta a su vez cuelga del Home.

### 3.2.3 Creación de las páginas de formulario

Ya podemos crear nuestras **landing pages** que incluirán un formulario para rellenar con los datos del usuario.

Volvemos al archivo `blog/models.py` y añadimos dos clases nuevas:

```
class FormField(AbstractFormField):
    page = ParentalKey('FormPage', on_delete=models.CASCADE,
related_name='form_fields')
```

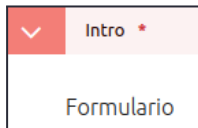
```
class FormPage(AbstractEmailForm):
    # Use custom form builder to override behaviour
    form_builder = CustomFormBuilder

    intro = models.CharField(max_length=250)
    thank_you_text = RichTextField(blank=True)
    body = models.CharField(max_length=250)
    content_panels = AbstractEmailForm.content_panels + [
        FormSubmissionsPanel(),
        FieldPanel('intro', classname="full"),
        FieldPanel('body'),
        InlinePanel('form_fields', label="Form fields"),
        FieldPanel('thank_you_text', classname="full"),
        MultiFieldPanel([
            FieldRowPanel([
                FieldPanel('from_address', classname="col6"),
                FieldPanel('to_address', classname="col6"),
            ]),
            FieldPanel('subject'),
        ], "Email notification"),
    ]
```

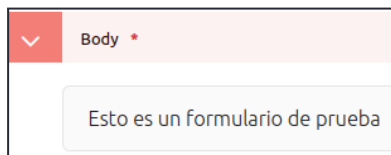
La primera clase (`FormField`) es necesaria para construir los campos del formulario. Mientras que la segunda (`FormPage`) corresponde a una página de formulario que también puede enviar los resultados a la dirección de correo que indiquemos si queremos.

En nuestra página de formulario hemos añadido los siguientes elementos:

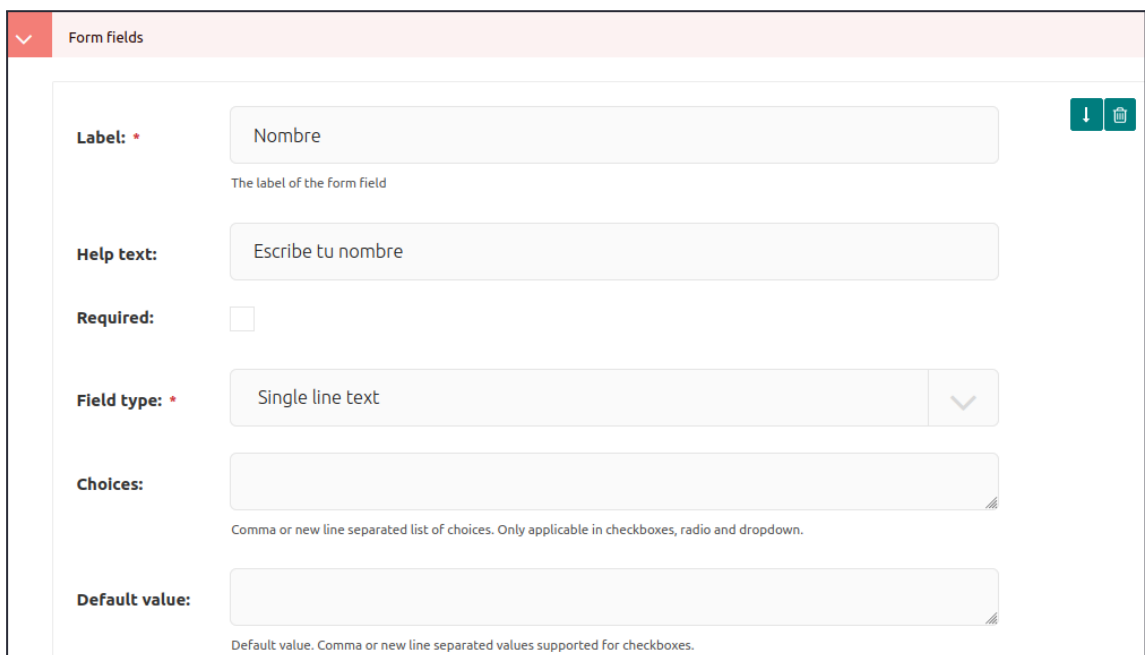
- **intro**: corresponde al texto principal de la página.



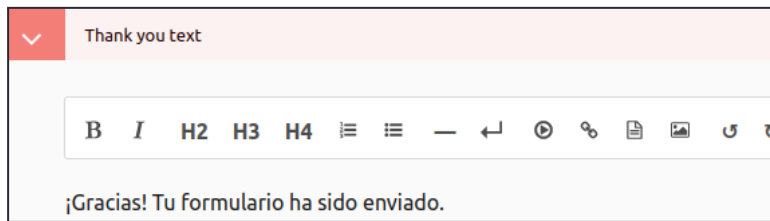
- **body**: texto secundario de la página.



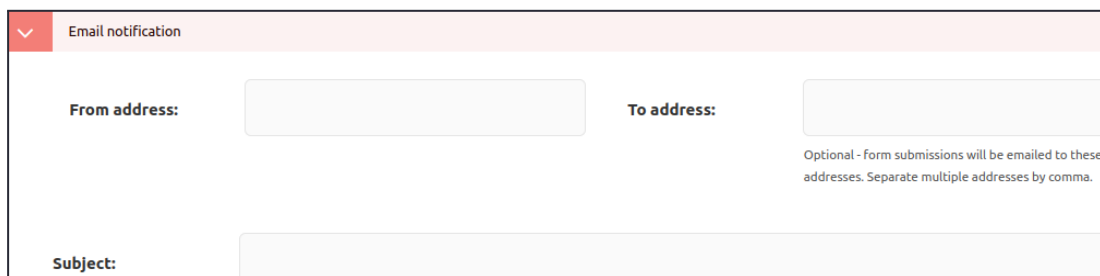
- **form\_fields**: Aquí podemos ir añadiendo los distintos campos de nuestro formulario (texto, número, email, url, checkbox, etc.). Podemos también indicar qué elementos son obligatorios y cuáles no.



- **thank\_you\_text**: texto que se mostrará en la página de agradecimiento una vez que se haya enviado el formulario.



- **from\_address** y **to\_address**: direcciones de origen y destino a la que se enviará (opcionalmente) las respuestas del formulario.
- **subject**: asunto del correo que se envía.



Cabe destacar que tanto **intro** como **body** son del tipo 'CharField' lo que significa que Wagtail no agregará ningún tipo de estilo ni encabezado al texto, aunque luego podemos modificarlo a nuestro gusto desde la plantilla. Mientras que **thank\_you\_text** es del tipo 'RichTextField' al cual podemos añadir estilos como negrita, cursiva o algún tipo de encabezado (H2, H3...), además permite añadir vídeos incrustados, imágenes o documentos entre otros, todo directamente desde el panel de edición.

Con 'CharField' es el desarrollador el que decide cuál será el estilo final del texto, mientras que con 'RichTextField' damos más libertad al editor para decidir el formato del contenido mostrado.

Una vez aplicados los cambios desde el panel de administrador, podemos acceder al formulario.

127.0.0.1:8010/blog/Formulario2/

# FORMULARIO

Esto es un formulario de prueba

**Nombre**  
Escribe tu nombre

**Apellidos**  
Escribe tus apellidos

**Email**  
ejemplo@gmail.com

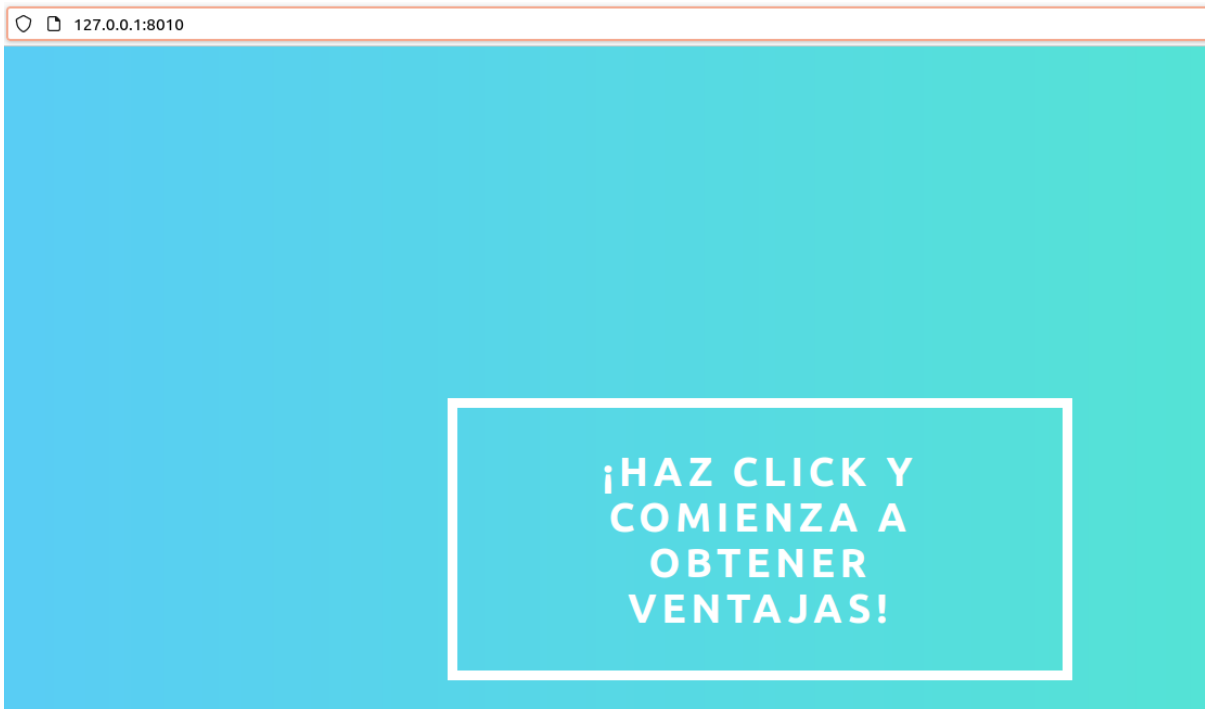
**Nacionalidad**

- Española
- Otra

ENVIAR CONSULTA

### 3.2.4 Página principal

Además de una página con el formulario, se ha diseñado una página de inicio que contiene un botón CTA de forma que el usuario que entrase a nuestra web se viera motivado a hacer clic para acceder a la landing page, que en este caso es el formulario anterior.



### 3.3 Procesar los leads procedentes de las landings con Celery

Para resolver el problema del procesamiento de los leads de manera que los usuarios no vean cómo se demoran los tiempos de carga de nuestro sitio web, vamos a aplicar la siguiente solución:

Enviaremos el lead a un broker<sup>24</sup> (como **Redis**<sup>25</sup>) que se encargará simplemente de almacenar los leads y devolver un OK al servidor web. Esto lo haremos a través de un gestor de mensajes distribuidos llamado **Celery**<sup>26</sup>. En conjunto actuarán como una cola de mensajes de tipo publicador/suscriptor que permitirá tratar los leads de manera asíncrona. La figura inferior muestra un diagrama de secuencia que representa esta forma asíncrona de procesar los datos.

---

<sup>24</sup> Traducido como “**intermediario**”. En este caso entre Celery que almacena temporalmente un formulario en el broker para que posteriormente un worker lo recoja y lo envíe a una base de datos.

<sup>25</sup> **Redis** es una base de datos del tipo clave-valor de código abierto. A menudo se usa como memoria caché para acelerar aplicaciones web. El nombre de Redis viene de **RE**remote **DI**ctionary **S**erver o servidor de diccionario remoto.

<sup>26</sup> **Celery** es una biblioteca Python de código abierto que se utiliza para ejecutar tareas de forma asíncrona fuera del ciclo de petición-respuesta de HTTP, de forma que nuestro servidor web responda a las peticiones entrantes lo más rápido posible, realizando otras tareas más pesadas en segundo plano.



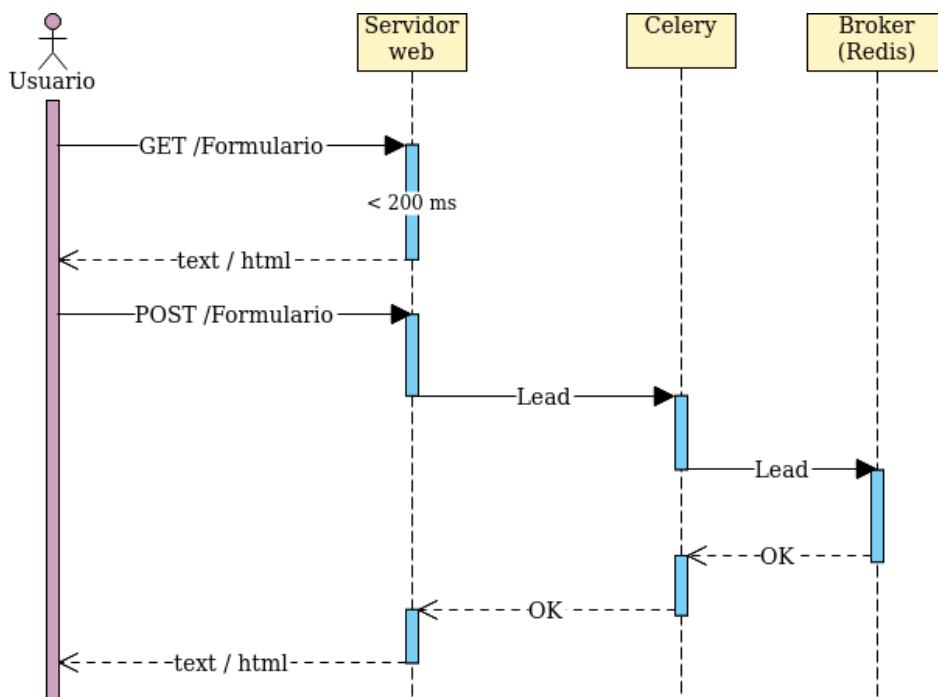


Diagrama de secuencia de petición del formulario y posterior envío. Devolviendo al servidor la página de agradecimiento.

En el diagrama anterior vemos como un usuario accede a una landing page de nuestro sitio web y una vez rellenado el formulario lo envía de vuelta al servidor. En este momento entra en juego Celery, que toma dicho formulario y lo almacena en el broker para su posterior procesamiento de manera asíncrona. Una vez almacenado el formulario en el broker le devuelve al usuario una nueva página web de agradecimiento.

Finalmente, almacenaremos los datos en una base de datos o soporte persistente, como puede ser **Google Sheets**<sup>27</sup>, tal y como se puede ver en el diagrama de secuencia de abajo.

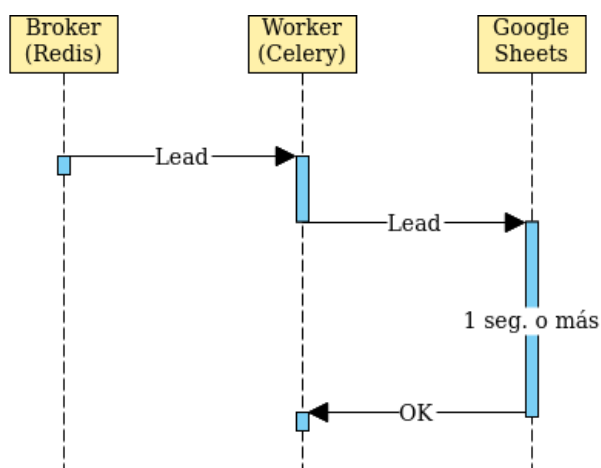


Diagrama de secuencia que muestra el procesamiento asíncrono del lead por parte del worker de Celery.

<sup>27</sup> **Google Sheets** es una aplicación web que permite crear y editar hojas de cálculo.

En este diagrama observamos que cuando llega un nuevo mensaje al broker, se activa un worker de Celery, el cual se encargará de procesar dicho formulario para almacenarlo en una base de datos, que en este caso será Google Sheets.

### 3.3.2 ¿Qué es Celery y cuál es su utilidad?

Trabajar con una aplicación que va a recibir un flujo intenso de datos puede ralentizar el tiempo de carga de nuestro sitio web, haciendo que la experiencia de los usuarios no sea la mejor. Hoy en día esperamos que las páginas carguen al instante, pero ejecutar tareas con muchos datos puede suponer un retardo de varios segundos o incluso más. [67][68][69]

Para evitar ese tipo de situaciones necesitamos quitar parte del trabajo pesado a nuestro servidor web. Una manera de lograrlo es mediante las tareas asíncronas. Mientras el servidor web va cargando la siguiente página, hay otro proceso en segundo plano que se encarga de realizar esas tareas que requieren más tiempo. A estos procesos se les llaman **'workers'** o trabajadores.

Estos trabajadores pueden ejecutar tareas que no requieren ser atendidas en el momento y que pueden lastrar otras que sí requieren ser procesadas en ese instante. Algunas de esas tareas pueden ser efectuar algún cambio en la base de datos, procesar información, enviar un correo electrónico (o muchos). Esto permite a nuestro servidor web poder atender las peticiones de los usuarios mucho más rápido.

La forma que tenemos de decirle a los trabajadores que deben hacer es a través de una cola de mensajes, que como su nombre indica, se trata de una estructura de datos en la que el primer elemento o tarea puesto en la cola será el primero en ser procesado siguiendo el método FIFO (First Input First Output). Así, cuando un trabajador no esté realizando ninguna tarea, cogerá la primera tarea en la cola y la comenzará a procesar. De esta forma garantizamos que cada trabajador haga una sola tarea a la vez, que se hagan según el orden de entrada y que cada tarea se ejecute una sola vez.

Aquí es donde entra **Celery** que es una cola de tareas distribuida, gestionada de manera asíncrona (es decir, fuera del programa principal) de código abierto y desarrollado en Python. Esto lo hace a través de un intermediario (**broker**) que media entre la aplicación y los trabajadores y que se encarga de ir encolando los elementos que después serán distribuidos a los workers quienes se encargarán de procesarlos. En este caso el broker usado es **Redis**.

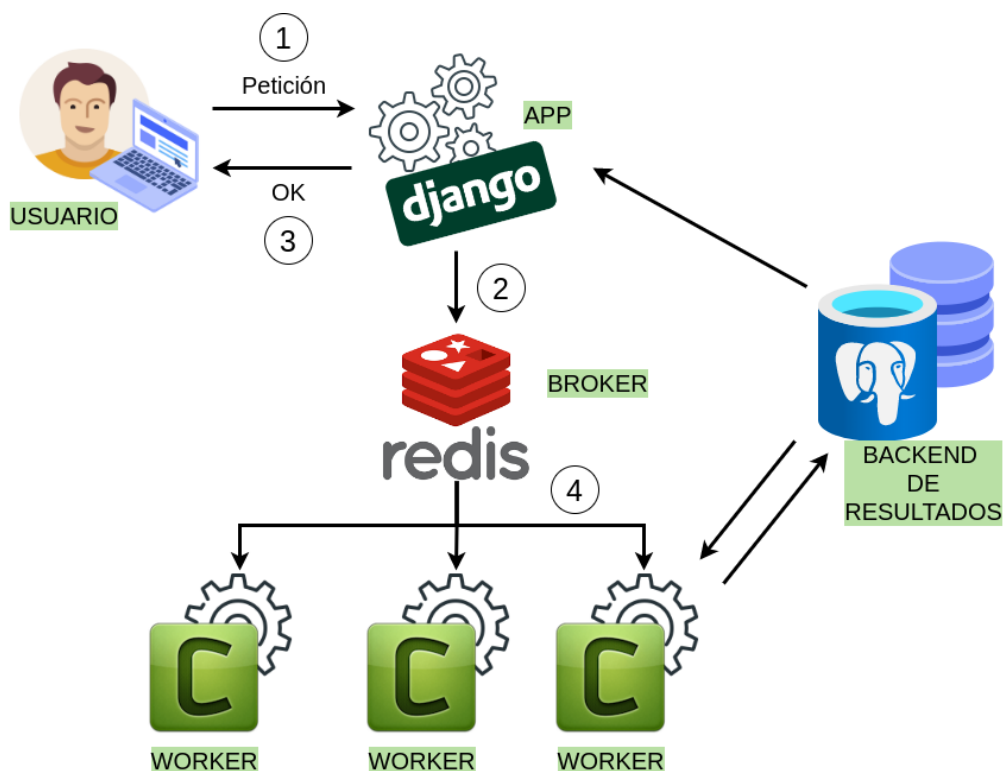
Por otro lado, Celery requiere de un **backend**, el cual almacenará información relevante sobre cada tarea que mandemos. Contiene, entre otras cosas, información sobre el estado en el que se encuentra la tarea en cada momento (pendiente, iniciada, finalizada, fallida, etc.) además de los datos retornados por la tarea (si los hubiera).

Con todo lo descrito anteriormente, puede verse que el funcionamiento de Celery se asemeja al de productor-consumidor<sup>28</sup>. Donde el productor sería la aplicación que genera las tareas y el consumidor serían los trabajadores de Celery que toman dichas tareas y las procesan.

Por último destacar que Celery soporta una gran variedad de frameworks entre los que se encuentra Django.

---

<sup>28</sup> Sistema de comunicación asíncrono entre servicios en el que los productores publican mensajes en un intermediario referenciados por un tema determinado, de forma que todos los consumidores que estén suscritos a ese tema recibirán dicho mensaje. Normalmente, se usa en arquitecturas sin servidor y de microservicios.



Esquema del funcionamiento de Celery.

## 3.4 Aplicación de Celery en la solución

En este apartado explicaremos por partes cómo hemos utilizado Celery para poder procesar los leads de forma asíncrona.

### 3.4.1 Instalación y configuración

Comenzamos instalando los paquetes necesarios:

```
pip install celery
pip install redis
pip install django-celery-results
```

Además de Celery y Redis necesitamos instalar **django-celery-results** el cual es una extensión que permite almacenar los resultados de las tareas de Celery usando el ORM<sup>29</sup> de Django. Esta extensión define un nuevo modelo (`django_celery_results.models.TaskResult`) al cual podemos hacer consultas como si se tratara de cualquier base de datos de Django.

En el archivo de configuración debemos añadir `django_celery_results` a la lista de aplicaciones instaladas e indicar que backend utilizará Celery.

```
CELERY_RESULT_BACKEND = "django-db"
```

<sup>29</sup> El ORM (Object-Relational-Mapper) de Django es una herramienta de software que nos permite hacer consultas a una base de datos SQL sin necesidad de emplear código SQL. Pudiendo hacerlo a través de código en Python orientado a objetos.

Es necesario actualizar la base de datos para que se creen las tablas nuevas.

Creamos un nuevo archivo llamado `celery.py` para configurar Celery. En este archivo indicaremos el broker que emplearemos, en este caso Redis y el puerto en el que escucha.

```
BASE_REDIS_URL = os.environ.get('REDIS_URL', 'redis://localhost:6379')
```

### 3.4.2 Creación de tareas

Generamos un nuevo archivo llamado `tasks.py` el cual contendrá las definiciones de las tareas que realizarán los workers. Vamos a definir una tarea que guarde los campos de un formulario en una hoja de cálculo de Google Sheets

```
from celery import shared_task
import gspread

@shared_task(name="almacenar_formulario")
def storeForm(form):
    sa = gspread.service_account()
    sh = sa.open("HojaDePrueba")
    wks = sh.worksheet("Hoja 1")

    res_list = form.values()
    new_value = list(res_list)
    wks.append_row((list(map(str, new_value))), table_range='A1')
```

A continuación necesitamos hacer que cada vez que un usuario envíe un formulario, este sea almacenado en el broker para que pueda ser procesado por el worker de forma asíncrona. Para ello hay que sobrescribir el método `process_form_submission` del modelo `FormPage`.

```
def process_form_submission(self, form):
    from blog.tasks import storeForm

    submission = super().process_form_submission(form)

    storeForm.delay(form.cleaned_data)

    return submission
```

Aquí vemos que en la tercera línea indicamos que se realicen las acciones por defecto del método `process_form_submission`. Pero además en la cuarta línea añadimos la tarea asíncrona a ejecutar con el `“.delay”` y pasándole como argumento el formulario enviado.

### 3.4.3 Ejecución de Celery

Para iniciar el worker de Celery es necesario ejecutar el siguiente comando:

```
celery -A mysite worker -l info
```

Ahora, cada vez que un usuario envíe un formulario, este se almacenará en la hoja de Google Sheets que hemos indicado.

	A	B	C	D
1	nombre	apellidos	email	nacionalidad
2	Carlos	Moreno Guillén	midireccion@gmail.com	Española
3				

## 3.5 Pruebas de carga con Locust

Para comprobar que todo funcionaría de la manera esperada en una situación real en la que muchos usuarios accediesen a nuestro servidor al mismo tiempo, debemos realizar una serie de pruebas de carga para comprobar que no se produce ningún fallo inesperado que nos pudiera sorprender cuando la aplicación se encuentre disponible al público general.

### 3.5.1 ¿Qué es una prueba de carga?

Una vez el desarrollo de la aplicación web está cerca de terminar, conviene cerciorarse de que nuestra aplicación se comportará de manera correcta, incluso durante periodos de alta carga de peticiones. Esto se consigue a través de las pruebas de carga, las cuales además nos ayudarán a saber en caso de que algo no vaya de la forma esperada, cuál es el punto que está generando dicho fallo o el punto de ruptura. [72]

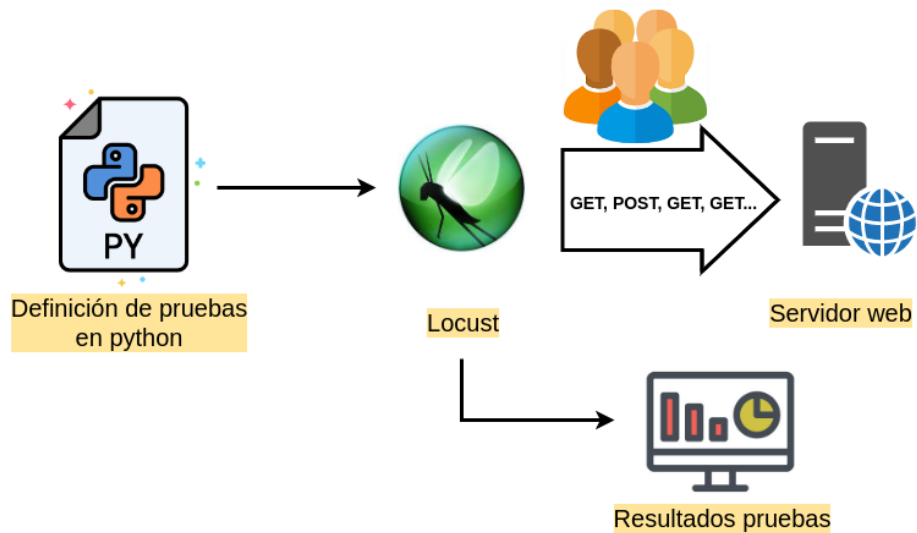
Estas pruebas nos permiten saber si nuestro sitio web cumple con unas condiciones mínimas de rendimiento para poder ofrecer un servicio que cumpla con unos estándares de calidad de cara al usuario.

Con las pruebas de carga podemos simular que pasaría en el caso de recibir una alta demanda en nuestra web. De esta forma podemos prevenir que se diese dicho fallo en una situación real, lo que podría suponer la pérdida de la confianza de los usuarios y, en el caso de que fuera la web de un negocio, incluso la pérdida de ingresos.

### 3.5.2 ¿Qué es Locust?

Locust es una herramienta de pruebas desarrollada en Python que es utilizada para realizar tests de carga (principalmente para servidores web) simulando el comportamiento de un usuario que visita nuestra web. Para ejecutar dichas pruebas ejecuta un conjunto de funciones en Python que simulan diferentes acciones de un gran número de usuarios web. Todo esto con el objetivo de ver cuáles son los principales puntos a mejorar de nuestro servicio, ya sea en cuestiones de rendimiento, capacidad o de seguridad. [70][71]

Una vez ejecutadas las pruebas, Locust muestra los resultados de forma gráfica en una tabla. Esta se accede a través de una interfaz web que además nos muestra el proceso de las pruebas en tiempo real. Incluso pudiendo cambiar los parámetros de la prueba mientras está en ejecución. De esta forma facilita el análisis de los mismos. También es posible efectuar las pruebas sin necesidad de una interfaz de usuario en el caso de que no se requiera.



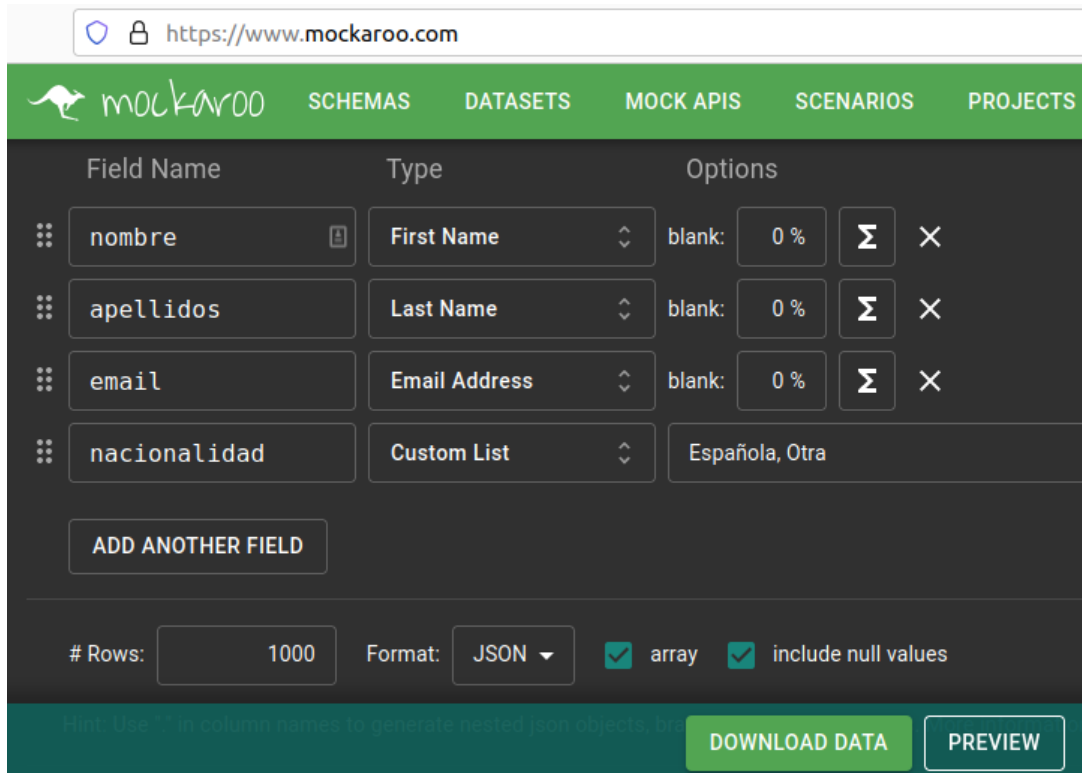
Esquema del funcionamiento de Locust.

### 3.5.3 Instalación y configuración de Locust

Comenzamos instalando Locust.

```
pip3 install locust
```

Ahora vamos a crear un archivo json que contenga un gran número de ejemplos de datos del formulario, para simular que se trata de diferentes usuarios. Para ello vamos a hacer una herramienta llamada Mockaroo que nos permite generar datos aleatorios de diferentes tipos en varios formatos, entre ellos json [73]. Esta herramienta permite generar hasta 100 filas de datos realistas con más de 74 tipos diferentes (nombre, apellido, email, dirección IP o hasta nombres científicos de plantas). Para ello debemos acceder a su página web y seleccionar la clave, el tipo de valor asociado a cada clave, el número de líneas que queremos generar y el formato de salida. En la siguiente imagen se puede ver un ejemplo de las opciones que se han seleccionado para hacer nuestras pruebas.



Interfaz de Mockaroo con las opciones necesarias para realizar nuestras pruebas.

El archivo generado tendrá en aspecto similar al mostrado a continuación:

```
[{"nombre": "Seamus", "apellidos": "Klas", "email": "sklas@earthlink.net", "nacionalidad": "Otra"},
{"nombre": "Nye", "apellidos": "Pilkington", "email": "npilkington1@storify.com", "nacionalidad": "Otra"},
{"nombre": "Robinet", "apellidos": "Wheeldon", "email": "rwheeldon2@discuz.net", "nacionalidad": "Española"},
{"nombre": "Hillie", "apellidos": "Karpets", "email": "hkarpets3@upenn.edu", "nacionalidad": "Española"},
{"nombre": "Gabi", "apellidos": "Crebbin", "email": "gcrebbin4@businessweek.com", "nacionalidad": "Otra"},
{"nombre": "Giffie", "apellidos": "Tilte", "email": "gtilte5@telegraph.co.uk", "nacionalidad": "Española"}]
```

Extracto del archivo traffic.json generado con Mockaroo.

Una vez descargado el archivo con todos los datos de prueba, pasamos a definir las pruebas a realizar con Locust en un nuevo archivo de python que llamaremos **traffic.py**.

Vamos a crear una clase llamada **Reader()** que se encargará de leer los elementos de nuestro archivo **traffic.json**.

```
class Reader():

    def __init__(self):
        self.array = []

    def pickRandom(self):
        lenght = len(self.array)

    if (lenght > 0):
        random_index = randrange(0, lenght - 1) if lenght > 1 else 0
```

```

        return self.array.pop(random_index)
    else:
        return None

    def load(self):
    try:
        with open("traffic.json", 'r') as data_file:
            self.array = json.loads(data_file.read())
    except Exception as error:
        print(f'>> Reader: No se cargaron los datos, error: {error}')

```

Primero definiremos una función **load** que se encargará de leer el archivo json con los datos que hemos generado en el paso anterior y guardarlos en un array.

La segunda función **pickRandom** extraerá una línea aleatoria del array que hemos creado en el paso anterior. De forma que una vez que se hayan extraído todas las líneas, la siguiente vez que llamemos a dicha función devolverá None.

Ahora creamos una segunda clase llamada **WebsiteUser** que será la que se encargue de simular a los usuarios que acceden a nuestra página.

```

class WebsiteUser(HttpUser):
    min_wait = 5000
    max_wait = 9000

    reader = Reader()
    reader.load()

    @task(1)
    def index_page(self):
        random_data = self.reader.pickRandom()

        if (random_data is not None):
            response = self.client.get("/blog/formulario2/")
            csrftoken = response.cookies['csrftoken']

            headers = {
                'Host': '127.0.0.1:8010',
                'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
Gecko/20100101 Firefox/102.0',
                'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*
;q=0.8',
                'Accept-Language': 'es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3',
                'Accept-Encoding': 'gzip, deflate, br',
                'Referer': 'http://127.0.0.1:8010/blog/formulario2/',
                'Content-Type': 'application/x-www-form-urlencoded',
                'Origin': 'http://127.0.0.1:8010',
                'Connection': 'keep-alive',
                'Upgrade-Insecure-Requests': '1',
                'Sec-Fetch-Dest': 'document',
                'Sec-Fetch-Mode': 'navigate',

```



```

'Sec-Fetch-Site': 'same-origin',
'Sec-Fetch-User': '?1'
}

sended = self.client.post('/blog/formulario2/',
                          data={'csrfmiddlewaretoken': csrftoken,
                                'nombre': random_data["nombre"], 'apellidos': random_data["apellidos"], 'email':
                                random_data["email"], 'nacionalidad': random_data["nacionalidad"]},
                          headers=headers)
    else:
        self.stop(True)

```

Primero comenzamos indicando el tiempo mínimo (**min\_wait**) y máximo (**max\_wait**) entre cada una de las tareas a realizar por el usuario simulado, expresado en milisegundos. Ahora cargamos en el objeto **reader** el json con los datos aleatorios que utilizaremos para simular el envío de formularios utilizando el método **load()**.

Una vez hecho esto, ya podemos indicar el contenido de la tarea a realizar denominada **@task(1)**. El argumento que contiene (1) indica la probabilidad de que esa tarea se ejecute en comparación con otras tareas. Si tuviéramos una tarea llamada **@task(3)** y otra **@task(6)**, significaría que la que tiene peso 6 se ejecutaría con el doble de frecuencia que la de peso 3.

Dentro de esta tarea comenzamos llamado al método **pickRandom** para almacenar uno de los elementos de nuestro json en una variable que hemos llamado **random\_data**. Comprobamos si el dato obtenido es **None**, en cuyo caso significaría que ya se han extraído todos los elementos del json, por lo que finalizaría la simulación. En caso de no ser **None** significa que hemos cogido un elemento aleatorio del json correctamente. A continuación hacemos una petición GET a la página del formulario para simular que accedemos a dicha página y además para obtener el token csrf. Normalmente dicho token está oculto, por lo que para poder realizar las pruebas debemos hacer que sea visible en las cookies. Una vez hechas las pruebas se debería volver a ocultar este token. Para hacerlo visible debemos añadir la siguiente línea en el archivo de configuración de Wagtail (**mysite/settings/base.py**):

```
CSRF_COOKIE_USED = True
```

Una vez obtenido el token, sólo falta añadir las cabeceras necesarias en la variable **headers** y lanzar la petición POST que incluirá el token csrf obtenido anteriormente, los datos del formulario y las cabeceras.

Para entender mejor la utilidad del csrf token vamos a ver más en detalle en qué consiste.

¿Qué es un token csrf y para qué sirve?

El token csrf sirve para evitar ataques csrf (Cross-Site Request Forgery, que traducido sería falsificación de solicitudes entre sitios). Este tipo de ataques explotan la confianza que tiene un determinado sitio web en un usuario autenticado, ya que conduce a dicho usuario a realizar peticiones al sitio web que este no tenía intención de hacer, como modificar los datos de un determinado formulario web o en general acciones que sean permanentes como enviar un mensaje, cambiar una contraseña o efectuar una compra. [74][75]

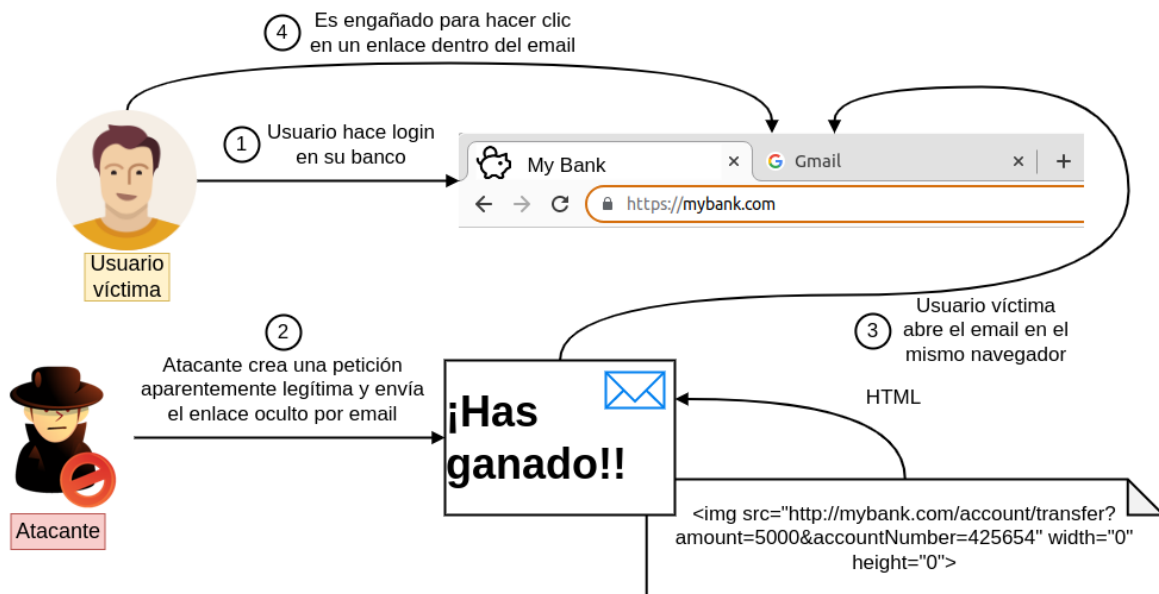
Para entender mejor en qué consiste un **ataque csrf** pongamos un ejemplo. Imaginemos que Bob accede a la cuenta de su banco a través de un navegador. La página web del banco mantendrá la sesión abierta para ese navegador durante un tiempo determinado, de forma que Bob pueda navegar por el sitio sin tener que identificarse cada vez que hace algo. Pero resulta que Bob tiene abierta en otra pestaña su bandeja de correo y ha recibido un correo electrónico aparentemente inofensivo. Lo que

pasa es que dicho correo contiene un fragmento de código que envía una petición al banco de Bob indicando que haga una transferencia desde la cuenta de Bob a la cuenta del atacante. La petición podría tener una forma como la siguiente:

```

```

Como la petición se hace desde el mismo navegador en el que Bob inició sesión en el banco, este se piensa que es el mismo Bob el que ha mandado esa petición intencionadamente, por lo que realiza la transferencia.



Ejemplo de ataque Cross-Site Request Forgery. (Fuente: reflectoring.io)

Para evitar este tipo de ataques, lo que se suele hacer es usar un **token csrf**, que es un conjunto de caracteres aleatorios que se genera cada vez que el usuario hace una petición de un formulario (o cualquier elemento que pueda efectuar modificaciones importantes como borrar una cuenta) y que está oculto dentro del propio formulario. De forma que haciendo clic en el botón enviar de dicho formulario irá adjunto este token oculto. Así, si un atacante intenta hacer un envío de un formulario haciéndose pasar por Bob con una petición desde otra página maliciosa con código oculto, al no poder acceder a dicho token, cualquier petición que envíe será ignorada.

### 3.5.4 Ejecución de Locust

Para ejecutar Locust primero debemos acceder al entorno virtual que hemos creado:

```
source locustenv/bin/activate
```

Una vez hecho esto, ejecutamos el siguiente comando para iniciar Locust:

```
locust -f traffic.py
```

Y ya podemos acceder al navegador y entrar a la interfaz de configuración de las pruebas de Locust a través de la url **http://0.0.0.0:8089/** que tendrá un aspecto como el siguiente:

### Start new load test

Number of users (peak concurrency)

Spawn rate (users started/second)

Host (e.g. http://www.example.com)

[Start swarming](#)

Interfaz de configuración de las pruebas de Locust.

El primer campo hace referencia al número de usuarios que simularemos que acceden a nuestra web. Cada usuario abrirá una conexión TCP distinta y realizará las pruebas programadas.

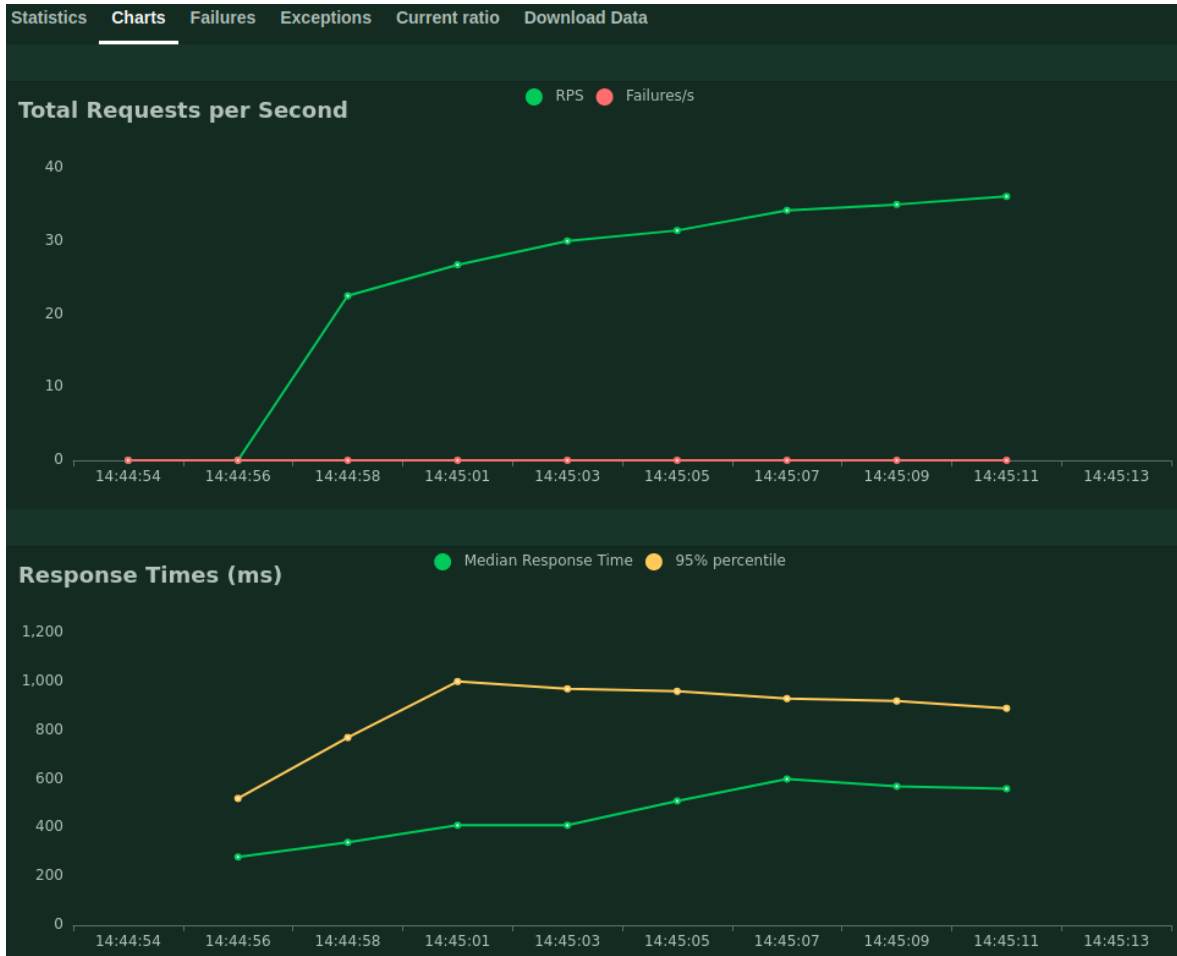
El segundo campo representa el número de usuarios que se irán accediendo a nuestra web cada segundo de manera progresiva hasta alcanzar el total de usuarios del campo anterior.

El último campo indica la url del host sobre el que vamos a hacer el test, enviando las peticiones indicadas en el código de pruebas. En este caso la de nuestro servidor de Wagtail.

Una vez pulsemos en ‘Start swarming’ comenzará el test y nos redirigirá a una página que muestra las estadísticas de los resultados de las pruebas.

<a href="#">Statistics</a> <a href="#">Charts</a> <a href="#">Failures</a> <a href="#">Exceptions</a> <a href="#">Current ratio</a> <a href="#">Download Data</a>												
Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/blog/formulario2/	320	0	730	880	1000	695	133	1077	4541	18	0
POST	/blog/formulario2/	313	0	320	430	1200	341	93	1375	206	18.1	0
<b>Aggregated</b>		<b>633</b>	<b>0</b>	<b>430</b>	<b>840</b>	<b>1100</b>	<b>520</b>	<b>93</b>	<b>1375</b>	<b>2397</b>	<b>36.1</b>	<b>0</b>

Interfaz de Locust que muestra los datos numéricos de los resultados del test.



Interfaz de Locust que muestra dos gráficas. La de arriba, indica el número de peticiones por segundo y la segunda el tiempo de respuesta del servidor.

Podemos ver que las pruebas se han realizado con éxito, ya que la base de datos de Google Sheets se ha rellenado con los datos aleatorios generados al principio.

1	nombre	apellidos	email	nacionalidad
2	Angelica	Lemary	alemary38@diigo.com	Española
3	Alta	Blessed	ablessedi@chicagotribune.com	Española
4	Nikaniki	Davenell	ndavenell39@si.edu	Española
5	Leelah	MacKay	lmackayll@scribd.com	Otra
6	Ned	Wilcott	nwilcottie@feedburner.com	Otra
7	Lolly	Morrell	lmorrellmf@newyorker.com	Española
8	Cherida	Colvill	ccolvillq4@adobe.com	Otra
9	Broderic	Carrane	bcarranern@geocities.com	Española
10	Ferdinanda	Ledgister	fledgisteril@dropbox.com	Española

## 3.6 Garantizar medidas de seguridad con Recaptcha

Debido a que los leads recibidos son anónimos, algún usuario malintencionado podría generar un gran número de datos falsos. Lo cual sería claramente contraproducente para nuestra labor de recolección de información. Para poder controlar este tipo de acciones haremos uso de la herramienta Recaptcha.

### 3.6.1 ¿Qué es Recaptcha?

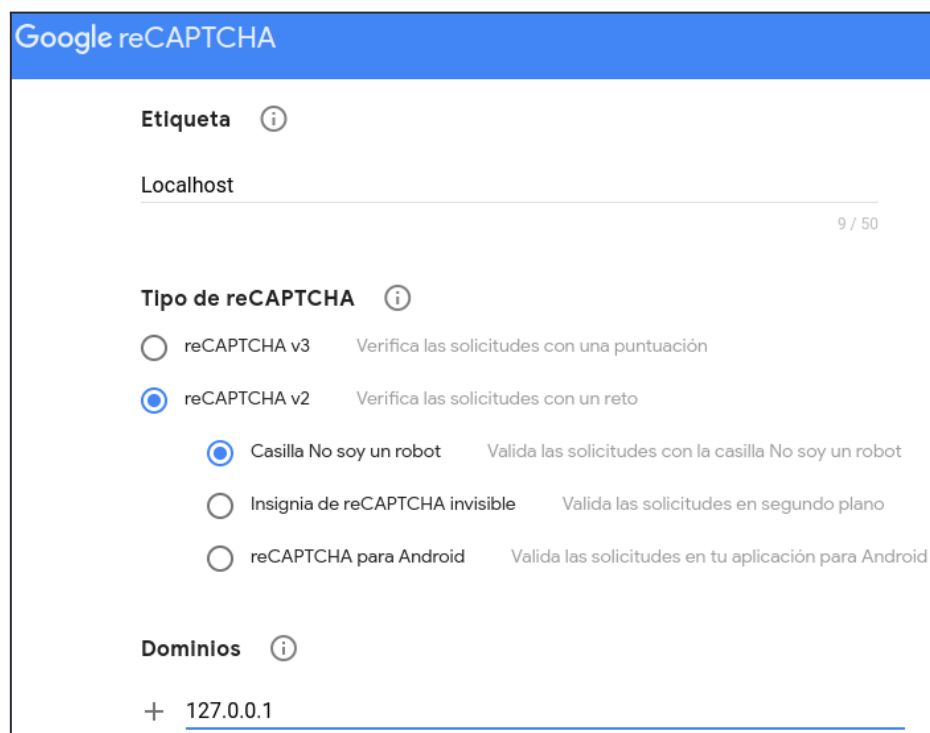
Recaptcha es un servicio de Google que ayuda a proteger los sitios web de las aplicaciones maliciosas. Genera una prueba que es fácil de realizar para los humanos, pero difícil para los bots, para así poder diferenciarlos y bloquear a los bots. [76]

### 3.6.2 Configuración de Recaptcha

En el archivo de configuración base.py añadimos las siguientes aplicaciones:

```
INSTALLED_APPS = [  
    # ...  
    'captcha',  
    'wagtailcaptcha',  
]
```

A continuación vamos a obtener las credenciales de la API de Google. Para ello entramos en la página <https://www.google.com/recaptcha/admin/create> y seleccionamos 'reCAPTCHA v2' con la opción 'Casilla No soy un robot'. Además, en dominios debemos poner el **127.0.0.1**, ya que estamos realizando pruebas en local.



### Propietarios

carlosmore96@gmail.com (Tú)

+ Introduce direcciones de correo electrónico

**Acepta las condiciones del servicio de reCAPTCHA**

Al acceder a las API de reCAPTCHA, o al utilizarlas, aceptas las [condiciones de uso](#) de APIs de Google, las [condiciones de uso](#) de Google y las Condiciones Adicionales que se indican a continuación. Lee y comprende todas las condiciones y políticas aplicables antes de acceder a las API.

Términos del Servicio de reCAPTCHA ▼

**Enviar alertas a los propietarios** ⓘ

[CANCELAR](#) [ENVIAR](#)

También debemos añadir el email del propietario de dicho dominio.

### Propietarios

carlosmore96@gmail.com (Tú)

+ Introduce direcciones de correo electrónico

**Acepta las condiciones del servicio de reCAPTCHA**

Al acceder a las API de reCAPTCHA, o al utilizarlas, aceptas las [condiciones de uso](#) de APIs de Google, las [condiciones de uso](#) de Google y las Condiciones Adicionales que se indican a continuación. Lee y comprende todas las condiciones y políticas aplicables antes de acceder a las API.

Términos del Servicio de reCAPTCHA ▼

**Enviar alertas a los propietarios** ⓘ

[CANCELAR](#) [ENVIAR](#)

Pulsamos en enviar y a continuación veremos dos claves, una pública y otra privada:

Se ha registrado "Localhost".

Usa esta clave de sitio web en el código HTML que tu sitio web sirve a los usuarios. [Más información sobre la integración en el cliente](#)

[COPIAR CLAVE DE SITIO WEB](#)

Usa esta clave secreta para la comunicación entre tu sitio web y el servicio reCAPTCHA. [Más información sobre la integración en el servidor](#)

[COPIAR CLAVE SECRETA](#)

[IR A CONFIGURACIÓN](#) [IR A ANALYTICS](#)

Ahora añadimos dichas claves en el archivo de configuración **base.py** de la siguiente forma:

```
RECAPTCHA_PUBLIC_KEY = "6Lds1Moh"
RECAPTCHA_PRIVATE_KEY = "6Lds1Moh"
```

Cambiar en **models.py** la línea

```
class CustomFormBuilder(FormBuilder):
```

por la siguiente línea:

```
class CustomFormBuilder(WagtailCaptchaFormBuilder):
```

Ahora, si reiniciamos el servidor, veremos que al acceder al formulario hay un nuevo campo que se corresponde con la casilla del reCAPTCHA:

The screenshot shows a web form with a dark blue background. At the top, there is a section titled "Nacionalidad" with two radio button options: "Española" (selected) and "Otra". Below this, there is a red asterisk indicating a required field. The reCAPTCHA widget is displayed, featuring a checkbox labeled "No soy un robot" and the reCAPTCHA logo with links for "Privacidad" and "Términos". At the bottom of the form, there is a white button with the text "ENVIAR CONSULTA".

Solo debemos marcarla y ya podremos enviar el formulario.

Cabe mencionar que he podido comprobar que realmente funciona, ya que cuando el reCAPTCHA está activo no se puede ejecutar el test de Locust. Esto se debe a que precisamente esta herramienta sirve para evitar que un bot pueda enviar formularios de manera automática. Por lo que para ejecutar el test sin problemas debemos volver a modificar la línea que cambiamos del archivo **models.py** en el paso anterior.

## 3.7 Docker y Docker Compose

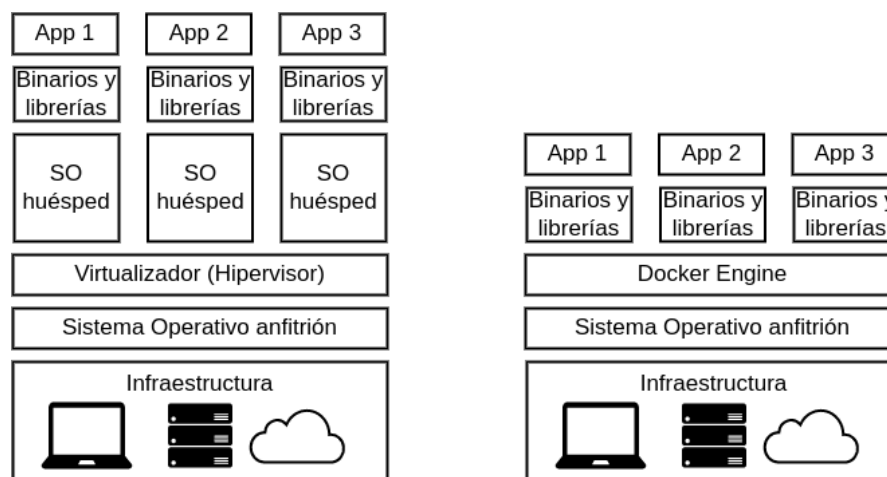
Para facilitar el despliegue y la portabilidad de la aplicación, vamos a hacer uso de Docker y Docker Compose.

### 3.7.1 ¿Qué es Docker y Docker Compose?

**Docker** es una plataforma de software que simplifica el proceso de creación, ejecución, administración y distribución de aplicaciones a través del uso de imágenes que se ejecutan dentro de contenedores. Los contenedores permiten empaquetar una aplicación con todas las bibliotecas y demás dependencias necesarias para su correcto funcionamiento. [77][78]

#### Diferencia entre máquina virtual y Docker

Una máquina virtual es un sistema que trata de simular una computadora completa con su propio sistema operativo independiente y que funciona dentro de nuestra propia computadora.



Diferencia entre MV y contenedor de Docker. (www.infoworld.com)

Como se puede observar en la imagen superior, cada máquina virtual posee su propio sistema operativo huésped que se ejecuta sobre el sistema operativo anfitrión. Esto hace que las máquinas virtuales sean muy pesadas, llegando a ocupar decenas de GB. Por otro lado, los contenedores de Docker no requieren de su propio sistema operativo, sino que comparten el sistema operativo anfitrión, lo que hace que sean mucho más ligeros (del orden de KB o MB).

Al no tener que iniciar el sistema operativo completo cada vez que ejecutamos un contenedor, podemos levantarlo en cuestión de segundos. Este proceso puede llevar minutos en el caso de las máquinas virtuales. [79]

Otra ventaja de Docker es su gran portabilidad, ya que puede llevarse a cualquier sistema que tenga instalado el **Docker Engine**<sup>30</sup> (tecnología de contenedores) (por ejemplo en una computadora, en un servidor remoto o como un servicio en la nube) y ejecutarlo al instante. [81]

A diferencia de las máquinas virtuales, los contenedores de Docker no poseen persistencia de datos, es decir, que cada vez que desplegamos un contenedor, este lo hace a través de una imagen que describe su contenido. Dicha imagen es inmutable, por lo que su contenido no varía una vez que ha sido creada. De forma que si nosotros hemos realizado cambios dentro de un contenedor, si luego

<sup>30</sup> **Docker Engine** es la parte de Docker que se encarga de generar y ejecutar los contenedores a partir de una imagen de Docker.



eliminamos dicho contenedor, todas las modificaciones que hayamos hecho se perderán, ya que si volvemos a levantar un contenedor a partir de una imagen, esta no tiene la información referida a dichos cambios. Si quisiéramos que dichos cambios permanezcan cada vez que levantamos el contenedor, debemos hacer uso de herramientas para añadir persistencia de datos, como pueden ser los **volúmenes** o los **puntos de montaje**. [80]

La diferencia principal entre un punto de montaje y un volumen es que el punto de montaje es un área de almacenamiento en la máquina local que se incluirá en el contenedor al arrancarlo y que puede ser modificado tanto por parte del contenedor como por nuestra máquina local. De forma que todos los cambios que hagamos en el punto de montaje desde fuera del contenedor se verán reflejados también dentro de este. Mientras que el volumen está pensado para ser administrado únicamente por Docker, y pueden estar enlazados a más de un contenedor de forma simultánea, de forma que los cambios que hagamos en un contenedor se verán reflejados en el otro.

**Docker Compose** es una herramienta que sirve para ejecutar múltiples contenedores como un solo servicio, pudiendo gestionarlos de una forma más centralizada, simplificando el proceso de despliegue de los contenedores. Para ello utiliza un archivo de configuración YAML que define los servicios (componentes de nuestra aplicación), volúmenes (para almacenar datos persistentes de los servicios) y redes (para la comunicación entre servicios) que serán utilizados por nuestra aplicación. De esta forma, con un solo comando podemos desplegar todos los componentes y volúmenes necesarios para el correcto funcionamiento de nuestra aplicación. [82][83]

### 3.7.2 Dockerizando nuestra aplicación

Vamos a pasar a dockerizar nuestra aplicación de Wagtail para poder desplegarla rápidamente. La estructura de los principales archivos de nuestro proyecto al final de este proceso será la siguiente:

```
mysite/
├── blog/
│   ├── migrations/
│   ├── static/
│   ├── templates/
│   ├── models.py
│   └── tasks.py
├── compose/
│   └── local/
│       ├── django/
│           ├── celery/
│               └── worker/
│                   └── start
│           ├── Dockerfile
│           ├── entrypoint
│           └── start
├── mysite/
│   ├── celery.py
│   ├── __init__.py
│   ├── urls.py
│   └── wsgi.py
├── docker-compose.yml
├── manage.py
└── requirements.txt
```

Aquí nos limitaremos a presentar la estructura del archivo de configuración de Docker Compose:

```
version: '3.8'

services:
  web:
    build:
      context: .
      dockerfile: ./compose/local/django/Dockerfile
    image: django_celery_example_web
    # '/start' es el script usado para iniciar el servicio
    command: /start
    volumes:
      - ./app
    ports:
      - 8010:8000
    # env_file lo usaremos para gestionar las variables de entorno
    env_file:
      - ../.env/.dev-sample
    depends_on:
      - redis
      - db

  db:
    image: postgres:14-alpine
    volumes:
      - postgres_data:/var/lib/postgresql/data/
    environment:
      - POSTGRES_DB=hello_django_dev
      - POSTGRES_USER=hello_django
      - POSTGRES_PASSWORD=hello_django

  redis:
    image: redis:6-alpine

  celery_worker:
    build:
      context: .
      dockerfile: ./compose/local/django/Dockerfile
    image: django_celery_example_celery_worker
    command: /start-celeryworker
    volumes:
      - ./app
    env_file:
      - ../.env/.dev-sample
    depends_on:
      - redis
      - db
```

Como se puede observar vamos a crear cuatro servicios, correspondiendo cada uno a un contenedor distinto:

- **web**: será el que contenga nuestro servidor de Wagtail y accederemos a este a través del puerto 8010.
- **db**: es la base de datos que utilizará nuestra aplicación de Wagtail. En este caso será **PostgreSQL** en lugar de SQLite (usada por defecto por Django y en consecuencia también por Wagtail). Es recomendable usarla en lugar de SQLite, ya que es más potente y robusta, además de segura. También es una de las más utilizadas por la comunidad de Django y ofrece muchas más posibilidades de cara a escalar la funcionalidad de nuestra aplicación en un futuro. [84] Podemos ver también que la propia documentación de Django recomienda el uso de PostgreSQL:

*“Django provides support for a number of data types which will only work with PostgreSQL. There is no fundamental reason why (for example) a contrib.mysql module does not exist, except that PostgreSQL has the richest feature set of the supported databases so its users have the most to gain.” [9]*

- **redis**: es el servicio de redis que será empleado como broker de Celery.
- **celery\_worker**: corresponde al trabajador de Celery que se encargará de hacer las tareas de forma asíncrona.

Una vez terminada toda la configuración, solo queda desplegarla utilizando el siguiente comando desde la carpeta raíz del proyecto, donde se encuentra el archivo `docker-compose.yml`:

```
docker-compose up -d
```

Tras lo cual se levantarán los cuatro servicios que hemos indicado anteriormente:

```
Starting mysite_db_1      ... done
Starting mysite_redis_1  ... done
Starting mysite_web_1    ... done
Starting mysite_celery_worker_1 ... done
```

Y de esta forma ya podremos usar la aplicación de forma normal a través de la url:

```
http://127.0.0.1:8010/
```

Para parar la aplicación solo debemos ejecutar el siguiente comando desde el mismo directorio:

```
docker-compose down
```

```
Stopping mysite_web_1 ... done
Stopping mysite_celery_worker_1 ... done
Stopping mysite_redis_1 ... done
Stopping mysite_db_1 ... done
Removing mysite_web_1 ... done
Removing mysite_celery_worker_1 ... done
Removing mysite_redis_1 ... done
Removing mysite_db_1 ... done
Removing network mysite_default
```

## 4. Conclusiones

Este proyecto me ha servido para profundizar en desarrollo web y más concretamente en el CMS de Wagtail basado en el framework de Django. También he ampliado mis conocimientos sobre el envío de formularios y en materia de seguridad haciendo uso de distintas herramientas como Recaptcha o el token CSRF. He utilizado conocimientos sobre Python y sobre bases de datos. Además, he aprendido a ejecutar tareas asíncronas con Celery aprovechando también mis conocimientos de Redis. Asimismo, he entendido la importancia de realizar test de carga para comprobar el correcto funcionamiento del servidor web. Por último, he empleado mis conocimientos de Docker obtenidos en la carrera para poder facilitar el despliegue de la aplicación.

Fuera del apartado del desarrollo web, he ampliado mis conocimientos acerca de las diferentes técnicas de marketing, como pueden ser el Inbound y el Outbound Marketing, conociendo sus principales características así como sus ventajas y desventajas.

Para poner cierre a este trabajo, cabe decir que para el estudio de la metodología **inbound** fue necesario consultar una amplia variedad de artículos, ya que era un tema bastante desconocido para mí. Además, al ser un concepto muy reciente, no había demasiados documentos de carácter científico al respecto, por lo que tuve que consultar muchas fuentes para conseguir información relevante que fuera realmente objetiva y realista.

En cuanto al estudio de la parte relacionada con el desarrollo de la aplicación, comencé profundizando en el lenguaje de programación de **Python** porque, aunque tenía ya una base de conocimientos sobre este, creí necesario obtener unos conocimientos más amplios para poder enfrentar el trabajo de la mejor manera. Para ello hice varios cursos online. Lo mismo hice con **Django**, ya que al estar Wagtail basado en este framework, me parecía necesario comprender su funcionamiento para tener una base desde la que comenzar.

Una vez hecho esto ya pude comenzar a desarrollar mi sitio web utilizando **Wagtail**. Al principio fue un reto conseguir que el formulario se viera de forma adecuada, debido a que tuve que comprobar que los diferentes campos que se podían añadir al formulario se vieran bien, independientemente de cuáles se incluyeran desde el panel de administración de Wagtail.

Otro reto fue que a la hora de desarrollar la parte de **Celery**, que se encargaba de procesar los formularios de manera asíncrona, tarde más de lo esperado en conseguir enviar el formulario al worker, debido a que Wagtail procesa los formularios de otra manera y tuve que sobrescribir el método usado por este para añadir esta nueva funcionalidad. Para la elección del broker me decidí por utilizar **Redis** porque ya lo había utilizado en la carrera y estaba familiarizado con algunas de sus funcionalidades.

En lo que respecta a las pruebas de carga con **Locust**, aquí hubo varias cosas que se complicaba. Por un lado, estaba el token csrf que precisamente impedía a Locust poder enviar formularios, por lo que al final llegué a la solución de que era necesario incluirlo en las cookies temporalmente únicamente con propósito de poder realizar las pruebas. Además, tuve que buscar las cabeceras adecuadas, para añadir a la petición de envío del formulario.

Para la parte de seguridad, decidí implementar el **Recaptcha**, lo cual no fue demasiado difícil de integrar en comparación con otros servicios porque Wagtail ya contaba con información detallada al respecto. Aunque más tarde descubrí que, lógicamente, impedía que Locust hiciera el envío de formularios. Por lo que también tuve que desactivarlo de manera temporal para poder ejecutar los tests de carga.

Una vez finalizado el desarrollo de la aplicación y las pruebas, la última parte consistió en usar **Docker** y **Docker Compose** para facilitar el despliegue de la aplicación. En un principio ya contaba con bastantes conocimientos sobre esta herramienta, ya que pude verla en la carrera en el último año,

por lo que el trabajo consistió en profundizar un poco más en su funcionamiento y buscar información sobre cómo implementarla en mi proyecto.

Diría que la parte más complicada fue el comienzo, pues al ser un trabajo libre, no tenía una línea de dirección clara de hacia dónde llevar la investigación, aunque cabe decir que mi tutor me ayudó a decidir hacia dónde era mejor enfocar el proyecto. Por otra parte, es verdad que durante el desarrollo de la aplicación fueron surgiendo bastantes contratiempos debido a que no todo funcionaba de la manera esperada en un principio, por lo que dediqué mucho tiempo a buscar soluciones a estos problemas. Por último, en cuanto a la redacción de la memoria, quizá llevó más tiempo la parte de investigación acerca de la metodología inbound, ya que mis conocimientos de estos eran limitados, mientras que la redacción del procedimiento de desarrollo de la solución fue más asequible, puesto que me había documentado previamente mientras desarrollaba la aplicación.

Haciendo reflexión de todo lo que ha supuesto el desarrollo del proyecto, decir que estoy bastante satisfecho con la elección de las herramientas software usadas y he podido comprender mejor su utilidad haciendo uso de ellas para mi aplicación. He podido ver el potencial que tiene el lenguaje de Python y la versatilidad del framework de Django, así como la utilidad de usar un CMS como el de Wagtail. Además, he podido investigar en otros campos de conocimiento como el marketing, que me ha resultado de lo más interesante y ver como tiene una relación tan estrecha con el mundo de las tecnologías y en concreto de la programación web.

## 5. Líneas de trabajo futuro

Por último, se plantean algunas posibles líneas de mejora para el proyecto que se podrían llevar a cabo de cara a un futuro:

- Añadir la herramienta **Celery Flower** que sirve para monitorizar y gestionar los clusters de Celery. De esa forma podríamos obtener información más detallada sobre el desempeño de nuestras tareas de Celery.
- En lugar de utilizar Google Sheets como base de datos final, podríamos utilizar un **CRM** más potente como el de **Salesforce** o el de **HubSpot**.
- Implementar nuevas técnicas para la **mejora del SEO**, de forma que nuestra web tenga más probabilidades de encontrarse al principio de los resultados de búsqueda. Algunas medidas que se podrían tomar son:
  - La optimización de las URLs.
  - Mejorar la estructura HTML de la página
  - Aumentar la optimización del sitio web para mejorar los tiempos de carga o añadir más contenido relevante (como un blog) relacionado con la temática de nuestro sitio web.
- Además de las pruebas de rendimiento realizadas por Locust podríamos añadir más pruebas para comprobar la robustez de nuestra aplicación y que todo funcione según lo previsto. Algunas de las pruebas que podríamos incorporar son []:
  - Test unitarios, que consisten en probar de manera individual cada una de las funciones y métodos de nuestro software, de manera que nos aseguramos de que todo funciona como debería.
  - Test de integración, que se encargaran de revisar que los diferentes servicios de nuestra aplicación se comuniquen de la forma esperada, por ejemplo que la aplicación web envíe correctamente el formulario al worker de Celery o que Celery le pase la información a la base de datos de la manera adecuada.
  - Pruebas de regresión, verifican que cada vez que añadimos o modificamos un servicio a nuestra aplicación, este no provoque que la funcionalidad de otro servicio que antes de dicha modificación funcionaba deje de hacerlo.
- Utilizar un certificado SSL para mejorar la seguridad de los usuarios y evitar que los datos enviados puedan ser interceptados por un atacante. Además, en el caso de este proyecto resulta especialmente interesante, ya que se envían formularios con datos personales de los usuarios a través de la red y no queremos que caigan en malas manos. Esta medida también ayudaría a mejorar el SEO de nuestra web, puesto que también se tiene en cuenta la seguridad de los sitios a la hora de posicionarlo en los resultados de búsqueda.

El código fuente se encuentra disponible en el siguiente repositorio:

[https://github.com/carlos-2710/TFG-Carlos\\_Moreno](https://github.com/carlos-2710/TFG-Carlos_Moreno)



# Citas y referencias

[1] Patrutiu-Baltes, Loredana. 2016. Inbound Marketing - the most important digital marketing strategy. ISSN: 20652194

[2] Real Academia Española. Diccionario de la Lengua Española.

<https://dle.rae.es/correo>

[3] Miguel Angel Zapatero. 2020. Prospeccionar con llamadas en frío: ¿realmente sirve para vender?. [hubspot.es](https://blog.hubspot.es)

<https://blog.hubspot.es/sales/por-que-tus-llamadas-en-frio-no-funcionan>

[4] Vera Sala. 2018. PPC o pay per click: qué es, ventajas y ejemplos. [cyberclick.es](https://www.cyberclick.es)

<https://www.cyberclick.es/numerical-blog/ppc-o-pay-per-click-que-es-ventajas-y-ejemplos>

[5] Amanda Clevinger. 2021. What is a Prospect: definition, ways to find and convert prospects. [snov.io](https://snov.io)

<https://snov.io/glossary/prospect>

[6] Ana Isabel Sordo. 2022. Qué son los leads y por qué son clave en el inbound marketing. [hubspot.es](https://blog.hubspot.es)

<https://blog.hubspot.es/marketing/leads>

[7] Carolina Samsing. 2021. Lead Generation: cómo conseguir leads de calidad. [hubspot.es](https://blog.hubspot.es)

<https://blog.hubspot.es/marketing/guia-para-la-generacion-de-prospectos-inbound>

[8] Ana Isabel Sordo. 2022. ¿Qué es una landing page y para qué sirve? (y aprende a crear la tuya). [hubspot.es](https://blog.hubspot.es)

<https://blog.hubspot.es/website/landing-page>

[9] Documentación Django.

<https://docs.djangoproject.com/en/4.1/ref/contrib/postgres/>

[10] How Google Adwords became Google Ads. [google.com](https://ads.google.com)

[https://ads.google.com/intl/en\\_uk/home/resources/how-google-adwords-became-google-ads/](https://ads.google.com/intl/en_uk/home/resources/how-google-adwords-became-google-ads/)

[11] Jessica Scarpati. 2021. URL (Uniform Resource Locator). [techtarget.com](https://www.techtarget.com)

<https://www.techtarget.com/searchnetworking/definition/URL>

[12] The Ultimate Guide: What is SSL, TLS and HTTPS?. [websecurity.digicert.com](https://www.websecurity.digicert.com)

<https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>

[13] MDN contributors. 2022. XML introduction. [developer.mozilla.org](https://developer.mozilla.org)

[https://developer.mozilla.org/en-US/docs/Web/XML/XML\\_introduction](https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction)

[14] Chiara Algarotti. Qué es la tasa de retención, la fórmula para calcularla y 3 formas para aumentar (y con ella tus beneficios). [doofinder.com](https://doofinder.com)

<https://www.doofinder.com/es/blog/tasa-retencion-ecommerce>

# Bibliografía

- [15] What Is Outbound Marketing?. wordstream.com  
<https://www.wordstream.com/outbound-marketing>
- [16] Dharmesh Shah, Brian Halligan. 2009. Inbound Marketing: Get Found Using Google, Social Media, and Blogs. Online ISBN: 9781118257838
- [17] Cheryl Baldwin. 2022. What is Inbound Marketing and Why Is It Important?. wsiworld.com  
<https://www.wsiworld.com/blog/what-is-inbound-marketing>
- [18] William Baker. Bringing in the leads: What is inbound marketing and what can it do for you?. fabrikbrands.com  
<https://fabrikbrands.com/what-is-inbound-marketing/>
- [19] Pau Valdés. 2022. Inbound marketing: qué es, fases o etapas y metodología. inboundcycle.com  
<https://www.inboundcycle.com/inbound-marketing-que-es>
- [20] Lea LeBlanc. 2021. What are Inbound Leads?. baremetrics.com  
<https://baremetrics.com/blog/what-are-inbound-leads>
- [21] Esther Varas. 2022. Qué es un lead: tipos y características. occamagenciadigital.com  
<https://www.occamagenciadigital.com/blog/qu%C3%A9-es-un-lead-tipos-y-caracter%C3%ADsticas>
- [22] Oriol Bel. 2022. ¿Qué es un lead, qué tipos hay y para qué sirven?. inboundcycle.com  
<https://www.inboundcycle.com/blog-de-inbound-marketing/que-es-un-lead>
- [24] Liz Moorehead. 2017. What Is a Call-to-Action, and What Should They Do for You?. impactplus.com  
<https://www.impactplus.com/blog/what-is-a-call-to-action>
- [25] Heather Park. 2022. The Role of the Call to Action in Inbound Marketing. geeklymedia.com  
<https://www.geeklymedia.com/blog/its-all-about-the-cta>
- [26] What is a call to action? Definition and examples. marketbusinessnews.com  
<https://marketbusinessnews.com/financial-glossary/call-to-action-definition/>
- [27] Basha Coleman. 2022. What Is a Landing Page? All Your Questions, Answered. hubspot.com  
<https://blog.hubspot.com/blog/tabid/6307/bid/7177/what-is-a-landing-page-and-why-should-you-care.aspx>
- [28] What Is a Landing Page?. unbounce.com  
<https://unbounce.com/landing-page-articles/what-is-a-landing-page/>
- [29] Cheryl Baldwin. 2020. What is a Landing Page & Why is it Important?. wsiworld.com  
<https://www.wsiworld.com/blog/the-importance-of-landing-pages>

- [30] Durán Flores, R.M. (2018). El inbound marketing y la importancia del marketing de contenidos. (Trabajo Fin de Grado Inédito). Universidad de Sevilla, Sevilla.
- [31] Waqar Hassan. 2016. The disadvantages of outbound marketing. b2bmarketing.net  
<https://www.b2bmarketing.net/en-gb/resources/blog/disadvantages-outbound-marketing>
- [32] John McTigue. 2011. A Brief History of Inbound Marketing. kunocreative.com  
<https://www.kunocreative.com/blog/bid/66753/a-brief-history-of-inbound-marketing>
- [33] Chaviva Gordon-Bennett. 2019. The Evolution of Inbound Marketing. smartbugmedia.com  
<https://www.smartbugmedia.com/blog/the-evolution-of-inbound-marketing>
- [34] Zlatko Bezovski. 2015. Inbound Marketing - a new concept in digital business
- [35] Allison Melnik. 2021. What is the Inbound Marketing Methodology?. impactplus.com  
<https://www.impactplus.com/blog/what-is-the-inbound-marketing-methodology>
- [36] Adam Basheer. 2017. What Is The Inbound Methodology?. fit4market.com  
<https://blog.fit4market.com/what-is-the-inbound-methodology>
- [37] What Is Inbound Marketing?. hubspot.com  
<https://www.hubspot.com/inbound-marketing>
- [38] What is Technical SEO?. bigcommerce.com  
<https://www.bigcommerce.com/ecommerce-answers/technical-seo/>
- [39] Laia Cardona. 2022. ¿Qué es el SEO? Definición, cómo funciona y guía completa de posicionamiento en buscadores (2022). cyberclick.es  
<https://www.cyberclick.es/que-es/seo>
- [40] A Beginner's Guide to CRM Systems. salesforce.com  
<https://www.salesforce.com/eu/learning-centre/crm/crm-systems/>
- [41] Kiely Kuligowski. 2022. 11 Benefits of CRM Systems. businessnewsdaily.com  
<https://www.businessnewsdaily.com/15963-benefits-of-crm.html>
- [42] Jesse Scardina. 2018. What is lead scoring?. techtarget.com  
<https://www.techtarget.com/searchcustomerexperience/definition/lead-scoring>
- [43] Lead nurturing: ¿qué es y cómo aplicarlo?. cyberclick.es  
<https://www.cyberclick.es/lead/lead-nurturing>
- [44] Amy Silberman. 2022. How Does Inbound Marketing Delight Customers?. dowhat.works  
<https://www.dowhat.works/blog/how-does-inbound-marketing-delight-customers>

[45] Erika Varagouli. 2020. SEO vs. SEM: What Is the Difference and How It Affects You. semrush.com

<https://www.semrush.com/blog/seo-vs-sem/>

[46] Inbound vs. Outbound Marketing: The Differences Between the Two. marketo.com

<https://www.marketo.com/articles/inbound-vs-outbound/>

[47] Brian Halligan. 2022. Inbound Marketing vs. Outbound Marketing. hubspot.com

<https://blog.hubspot.com/blog/tabid/6307/bid/2989/inbound-marketing-vs-outbound-marketing.aspx>

[48] Sanya Aru. 2022 Inbound vs Outbound: What Is Better For Your Marketing Strategy?. makewebbetter.com

<https://makewebbetter.com/blog/inbound-vs-outbound-marketing/>

[49] Joe Chernov. 2014. State of Inbound 2014. HubSpot

<https://cdn2.hubspot.net/hub/53/file-1589882006-pdf/HubSpot-State-of-Inbound-2014.pdf>

[50] Mimi An. 2018. State of Inbound 2018. HubSpot

<https://cdn2.hubspot.net/hubfs/3476323/State%20of%20Inbound%202018%20Global%20Results.pdf>

[51] Lindsay Kolowich Cox. 2022. Lead Generation: A Beginner's Guide to Generating Business Leads the Inbound Way. hubspot.com

<https://blog.hubspot.com/marketing/beginner-inbound-lead-generation-guide-ht>

[52] 2021. What Is Lead Generation and How Does It Fit Into Inbound Marketing?. cyberclick.net

<https://www.cyberclick.net/inbound-marketing/lead-generation>

[53] Phil Vallender. 2019. Inbound vs. outbound lead generation: The differences and similarities. blendb2b.com

<https://www.blendb2b.com/blog/inbound-vs-outbound-lead-generation-differences-and-similarities>

[54] Karissa Austin. 2019. MQL vs. SQL: How to define lead types in marketing and sales. callrail.com

<https://www.callrail.com/blog/mql-sql-definition-marketing-sales>

[55] Rebecca Riserbato. 2021. SQL vs. MQL: What They Are and How They Differ. hubspot.com

<https://blog.hubspot.com/sales/sales-qualified-lead>

[56] Laia Cardona. 2021. MQL and SQL in Inbound Marketing: How to Generate and Measure Leads. cyberclick.net

<https://www.cyberclick.net/numericalblogen/mql-and-sql-in-inbound-marketing-how-to-generate-and-measure-leads>

[57] Amanda Nielsen. 2018. Inbound Call-To-Action (CTA) Best Practices.newbreedrevenue.com

<https://www.newbreedrevenue.com/blog/inbound-call-to-action-cta-best-practices>

- [58] Erika Varagouli. 2020. SEO for Landing Pages: Best Practices to Rank. [semrush.com](https://www.semrush.com/blog/seo-landing-page/)  
<https://www.semrush.com/blog/seo-landing-page/>
- [59] Ted Vrontas. How to Build SEO Friendly Landing Pages. [instapage.com](https://instapage.com/blog/seo-friendly-landing-pages)  
<https://instapage.com/blog/seo-friendly-landing-pages>
- [60] What is a landing page? Definition and how to use it. [mailchimp.com](https://mailchimp.com/marketing-glossary/landing-pages/)  
<https://mailchimp.com/marketing-glossary/landing-pages/>
- [61] Sarah Amsler. 2021. What is a Content Managed System. [techtarget.com](https://www.techtarget.com/searchcontentmanagement/definition/content-management-system-CMS)  
<https://www.techtarget.com/searchcontentmanagement/definition/content-management-system-CMS>
- [62] Anna Fitzgerald. 2022. What Is a CMS and Why Should You Care?. [hubspot.com](https://blog.hubspot.com/blog/tabid/6307/bid/7969/what-is-a-cms-and-why-should-you-care.aspx)  
<https://blog.hubspot.com/blog/tabid/6307/bid/7969/what-is-a-cms-and-why-should-you-care.aspx>
- [63] MDN contributors. 2022. Django introduction. [developer.mozilla.org](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction)  
<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>
- [64] Cordenne Brewster. What Is Django?. [trio.dev](https://www.trio.dev/blog/what-is-django)  
<https://www.trio.dev/blog/what-is-django>
- [65] What is Django?. [djangogirls.org](https://tutorial.djangogirls.org/en/django/)  
<https://tutorial.djangogirls.org/en/django/>
- [66] Shweta Acharya. 2021. Wagtail CMS: Python CMS for Django. [externlabs.com](https://externlabs.com/blogs/wagtail-cms-for-python-django/)  
<https://externlabs.com/blogs/wagtail-cms-for-python-django/>
- [67] 2021. What is Celery & its Benefits for Python Developers. [appsinvo.com](https://www.appsinvo.com/blog/what-is-celery-its-benefits-for-python-developers/)  
<https://www.appsinvo.com/blog/what-is-celery-its-benefits-for-python-developers/>
- [68] Héctor Canto. 2021. Qué es Celery: Introducción y primeros pasos. [openwebinars.net](https://openwebinars.net/blog/que-es-celery-introduccion-y-primeros-pasos/)  
<https://openwebinars.net/blog/que-es-celery-introduccion-y-primeros-pasos/>
- [69] Filipe Ximenes. 2017. Celery: an overview of the architecture and how it works. [vinta.com](https://www.vinta.com.br/blog/2017/celery-overview-architecture-and-how-it-works/)  
<https://www.vinta.com.br/blog/2017/celery-overview-architecture-and-how-it-works/>
- [70] Mohamed Echout. 2022. What Is Locust Load Testing?. [blazemeter.com](https://www.blazemeter.com/blog/locust-load-testing)  
<https://www.blazemeter.com/blog/locust-load-testing>
- [71] Locust Documentation. What is Locust?  
<https://docs.locust.io/en/stable/what-is-locust.html>
- [72] What is Load Testing?. [microfocus.com](https://www.microfocus.com/en-us/what-is/load-testing)  
<https://www.microfocus.com/en-us/what-is/load-testing>
- [73] Indira Nellutla. 2017. Mockaroo Tutorial: Generate realistic test data. [qxf2.com](https://www.qxf2.com)

<https://qxf2.com/blog/data-generation-using-mockaroo/>

[74] Pratik Das. 2021. Complete Guide to CSRF/XSRF (Cross-Site Request Forgery). reflectoring.io  
<https://reflectoring.io/complete-guide-to-csrf/>

[75] What is Cross-Site Request Forgery. synopsis.com  
<https://www.synopsys.com/glossary/what-is-csrf.html>

[76] What is reCAPTCHA?. google.com  
<https://support.google.com/recaptcha/answer/6080904?hl=en>

[77] Yann Mulonda. 2018. What is Docker? ”In Simple English”. medium.com  
<https://yannmj1.medium.com/what-is-docker-in-simple-english-a24e8136b90b>

[78] Scott Carey. 2021. What is Docker? The spark for the container revolution. infoworld.com  
<https://www.infoworld.com/article/3204171/what-is-docker-the-spark-for-the-container-revolution.html>

[79] Avi. 2022. Docker vs Virtual Machine (VM) – Understanding the Differences. geekflare.com  
<https://geekflare.com/docker-vs-virtual-machine/>

[80] Mohammad Shahzeb Siddiqui. 2022. Bind Mounts & Volumes in Docker. dltlabs.com  
<https://www.dltlabs.com/blog/bind-mounts-volumes-indocker-133067>

[81] Docker Documentation. Docker Engine overview. docs.docker.com  
<https://docs.docker.com/engine/>

[82] 2022. What is Docker Compose: Example, Benefits and Basic Commands. simplilearn.com  
<https://www.simplilearn.com/tutorials/docker-tutorial/docker-compose>

[83] Docker Documentation. Compose specification. docs.docker.com  
<https://docs.docker.com/compose/compose-file/>

[84] Matt Makai. PostgreSQL - Full Stack Python. fullstackpython.com  
<https://www.fullstackpython.com/postgresql.html>

[85] Borja Miguel-Romero Castañer. 2014. Inbound marketing : la combinación de acciones de marketing digital (Trabajo Fin de Grado) Universidad Pontificia Comillas, Facultad de Empresariales (ICADE)

[86] Jess Unrein. 2018. Different types of testing explained. dev.to  
<https://dev.to/thejessleigh/different-types-of-testing-explained-1ljo>

[87] Documentación Django  
<https://docs.djangoproject.com/en/4.0/>

[88] Documentación Wagtail

<https://docs.wagtail.org/en/stable/>

[89] Documentación de Google Sheets API

<https://developers.google.com/sheets/api/guides/concepts>

[90] Documentación de REDIS

<https://redis.io/docs/>

[91] Documentación de Celery

<https://docs.celeryq.dev/en/stable/>

[92] Guía de Recaptcha para desarrolladores

<https://developers.google.com/recaptcha/intro>

[93] Documentación de Locust

<https://docs.locust.io/en/stable/>

[94] Documentación de Docker

<https://docs.docker.com/>







ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá