



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

LAIK:
PLATAFORMA SAAS DE MONITOREO DE MARCAS EN INTERNET

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN
COMPUTACIÓN

VICENTE DANIEL DAIE PINILLA

PROFESOR GUÍA:
SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:
NANCY HITSCHFELD KAHLER
PABLO GONZÁLEZ JURE

SANTIAGO DE CHILE

2022

Resumen

Las redes sociales y los sitios web cuentan con una gran cantidad de datos relativos a marcas comerciales. Dichos datos proporcionan información valiosa respecto del alcance de una marca, y el monitoreo de estos resulta fundamental para diseñar estrategias de marketing. Sin embargo, usualmente los datos se encuentran dispersos en diversas fuentes, lo que supone la inversión de una cantidad considerable de tiempo y recursos únicamente en acceder a la información.

LIG Tech es un emprendimiento que busca desarrollar soluciones de software para el monitoreo de marcas en Internet, a través del análisis de medios y redes sociales. Es en dicho contexto que surge este trabajo de memoria, el cual aborda el problema de dispersión de los datos de redes sociales y sitios web. El objetivo general de esta memoria es desarrollar una plataforma que solucione este problema, permitiendo acceder a toda la información necesaria para el monitoreo de marcas de manera centralizada y sencilla.

A modo de cumplir con los objetivos, se desarrolla la aplicación web Laik. Ésta cuenta con variados servicios que despliegan métricas de una marca, entre ellos, el análisis de usuarios de Twitter, de cuentas de Instagram, de páginas de Facebook y de sitios web mediante Google Analytics 4. Se tiene además un servicio de “favoritos”, en el cual el usuario podrá agrupar métricas provenientes de los demás servicios, y construir reportes personalizados sobre el estado de su marca en Internet. Los reportes podrán desplegarse en tiempo real, o exportarse a documentos en formato PDF.

La aplicación integra variadas tecnologías, entre ellas Flask como framework de back-end, React.js como framework de front-end, Celery como herramienta de administración de procesos pesados, y Docker como sistema de integración y despliegue. Una vez que la plataforma se encuentra operativa, ésta se monta en un servidor propio y se expone en una URL pública.

Se realizaron dos rondas de pruebas de usuarios para evaluar el cumplimiento de los objetivos planteados. De estas, se obtuvieron comentarios en su mayoría positivos, junto con ideas para extender la aplicación, por lo que se puede decir que se cumplieron los objetivos de la memoria. Tras hacer entrega del producto terminado a LIG Tech, se optó por continuar trabajando en la plataforma, incluyendo nuevos servicios para que ésta alcance su máximo potencial.

Agradecimientos

Me gustaría agradecer a todas las personas que hicieron posible que llegara hasta acá, apostando por mi educación, apoyándome, y creyendo en mí, a lo largo de todas las fases de mi vida.

A mi padre, Jorge, que sembró las semillas de la curiosidad en mí desde niño, mediante juegos, cofres y estrellas fugaces que recordaré por siempre.

A mi familia, quienes siempre han sido (y serán) un pilar de apoyo fundamental: mi madre Vivian, Rino, Alonso, Colomba, Sofía, Alessandra y Francesca. Y, cómo no, también a mis queridas Cleo, Atenea y Hera.

A Ignacio y a LIG Tech, quienes confiaron en mí para el desarrollo de este producto, y con quienes espero seguir trabajando para alcanzar el máximo potencial de Laik.

Al profesor Sergio, gracias a quien este trabajo de memoria pudo ser realidad, pese a la adversidad.

A la Universidad de Chile, por proporcionarme una formación con la cual puedo sentirme orgulloso y completo, y me permitirá alcanzar cualquier objetivo que me proponga.

Y, finalmente, me gustaría agradecer particularmente a mi madre, Vivian. Ella es la persona gracias a quien estoy acá, y que me ha dado apoyo permanente e incondicional en básicamente cualquier aspecto de mi vida. Esta va para tí, mamá.

Tabla de contenido

1. Introducción	1
1.1. Situación Inicial	2
1.2. Objetivos de la Memoria	2
1.3. Resumen de la Solución	3
1.4. Estructura del Documento	4
2. Marco Teórico	5
2.1. Redes sociales	5
2.2. Tratamiento de los datos	5
2.3. Acceso sistemático vía APIs	6
2.4. Estadísticas de sitios web	7
2.5. Dispersión de los datos	7
3. Concepción de la Solución	9
3.1. Levantamiento inicial de requisitos	9
3.2. Desarrollo del primer prototipo	9
3.3. Plan de desarrollo del producto	10
3.4. Tecnologías a utilizar	11
3.5. Servicios a brindar por la plataforma	12
4. Diseño de la Plataforma	14
4.1. Arquitectura de la solución	14
4.2. Modelo de datos	15
4.2.1. Modelo relacional	15
4.2.2. Archivos estáticos	15
4.2.3. Caché	16
4.2.4. Últimos informes	16
4.2.5. Archivos temporales	16
4.3. Diseño de interfaces de usuario	16
4.4. Sistema de manejo de usuarios	19
5. Implementación de la Plataforma	20
5.1. Desarrollo de base	20
5.1.1. Creación del proyecto	20
5.1.2. Armonización de tecnologías	20
5.1.3. Navegación	21
5.1.4. Implementación del sistema de manejo de usuarios	21
5.2. Tareas livianas	22
5.3. Desarrollo de servicios	23
5.3.1. Generalidades	24
5.3.2. Autenticación de usuarios	24
5.3.3. Administración de procesos pesados	25
5.3.4. Tareas del lado del cliente	26
5.4. Línea gráfica	26
5.5. Puesta en producción	27
5.6. Entrega del producto	28

6. Interfaces del Sistema	29
6.1. Landing page del sitio	29
6.2. Autenticación de usuarios	30
6.3. Vista de administración	31
6.4. Interfaces de servicios	32
6.4.1. Dashboard	32
6.4.2. Twitter	35
6.4.3. Facebook	36
6.4.4. Instagram	37
6.4.5. Google Analytics 4	38
6.4.7. Acceso a los informes	39
6.4.8. Modales de preferencias	40
6.4.9. Modales de autenticación	41
6.5. Configuración de cuenta de usuario	44
6.6. Documentos legales	44
7. Evaluación de la Solución	45
7.1. Proceso de evaluación realizado	45
7.2. Resultados obtenidos	45
8. Conclusiones y Trabajo a Futuro	48
8.1. Cumplimiento de los objetivos	48
8.2. Futuro del proyecto	48
8.3. Reflexión	49
9. Bibliografía	50

1. Introducción

Hoy en día hay muchas marcas comerciales que buscan mejorar su posición en el mercado; particularmente, aumentando su visibilidad en Internet. Para ello primero es necesario conocer cuál es el alcance (o visibilidad) que ésta tiene en Internet, y eso se puede lograr recopilando información acerca de los canales de comunicación de la marca; por ejemplo, redes sociales, prensa y sitios web.

En el caso de las redes sociales, éstas cuentan con métodos para obtener información acerca de las cuentas. En el caso de Twitter, cuenta con un botón en cada tweet para ver las interacciones de éste, mientras que Facebook e Instagram ofrecen reportes aún más detallados acerca de las interacciones de las cuentas. De manera adicional, las redes sociales ofrecen servicios gratuitos para extraer información de manera sistemática.

Respecto de los sitios web, hoy la plataforma por excelencia para recopilar información sobre el tráfico en sitios web es Google Analytics. Ésta ofrece diversas métricas y estadísticas personalizables, las cuales se despliegan en un portal dedicado. Además, se ofrece un servicio para realizar consultas de manera sistemática, similar a las redes sociales.

Es aquí donde se identifica una problemática particular: las mismas fuentes de datos entregan información distinta según el canal de comunicación sobre el que se transmita. Esto implica que las áreas de marketing y prensa de empresas deben invertir una gran cantidad de tiempo en la recopilación de información. También, se requiere de una amplia variedad de herramientas para acceder a todos los datos, lo que a su vez incrementa la curva de aprendizaje al momento de comenzar a analizar los canales de comunicación. Esto también implica mayores costos operacionales para las áreas antes mencionadas.

En este escenario, se identifica una oportunidad para brindar servicios de análisis de la información proveniente de los canales de comunicación, centralizándola en una única plataforma, de acceso sencillo e interfaz intuitiva. Esta plataforma debería permitir a los clientes ahorrar tiempo en el análisis de datos, y aumentar su capacidad de diseñar estrategias para mejorar el posicionamiento de la marca. La información podría además presentarse en diversos formatos, ya sea en tiempo-real para consultas rápidas, o de manera asíncrona para consultas periódicas y personalizables.

Ante esta oportunidad, en este trabajo de memoria se aborda el desarrollo de una aplicación web llamada *Laik*, que ofrece la información de posicionamiento de marca mediante los métodos mencionados, pero desde una única plataforma amigable para el usuario. Este trabajo es patrocinado por LIG Tech SpA¹, emprendimiento que busca desarrollar soluciones de software para monitoreo y análisis de medios y redes sociales.

¹ <https://www.ligtech.cl/>

1.1. Situación Inicial

Al momento de iniciar este trabajo de memoria, había algunas plataformas que realizaban algunos de los servicios mencionados anteriormente. Respecto del análisis de redes sociales, se pueden destacar plataformas como TweetBinder², en que se ofrece un análisis detallado de Twitter. Además, se permite realizar búsquedas por hashtag, cashtag, o analizar cuentas propiamente tal. Sin embargo, la plataforma se limita a analizar únicamente a Twitter (junto con algunas funciones de Instagram).

Un ejemplo más amplio es el de Sprout Social³, que ofrece servicios para múltiples redes sociales (Facebook, Twitter, Instagram, YouTube, Pinterest y LinkedIn), y no sólo de análisis de métricas, sino que también de planificación de contenido. Similar a esta aplicación, también existe Hootsuite⁴, que además incluye métricas de Google Analytics, pero que limita el número de cuentas de redes sociales a las que se puede acceder. Además, Hootsuite es incompatible con Google Analytics 4, que será el estándar desde julio de 2023, una vez que se dé de baja el servicio de Universal Analytics.

Plataformas de este tipo existen por toda la red. Sin embargo, no se observan aplicaciones que centralizan tanto análisis de redes sociales como de sitios web, y que ofrezcan al mismo tiempo análisis tanto en tiempo-real, como mediante informes generados automáticamente.

Una aplicación que cumple una función parecida sería Supermetrics⁵, la cual centraliza consultas a APIs de redes sociales y Google Analytics 4, y permite presentar los resultados como informes automáticos. Sin embargo, los precios son bastante superiores a los de las herramientas mencionadas anteriormente, y está pensada para programadores, debido al lenguaje que utiliza, lo que supone una curva de aprendizaje mayor para personas dedicadas al área de marketing.

Es por ello que, frente a la ausencia de una aplicación que realice estas funciones de manera centralizada para el usuario objetivo, se justifica la necesidad de desarrollar este proyecto.

1.2. Objetivos de la Memoria

El objetivo general de este trabajo de memoria es desarrollar una plataforma que permita a sus usuarios monitorear estadísticas de sus redes sociales y sitios web, tanto en tiempo-real, como mediante informes en PDF. Los objetivos específicos que se desprenden del objetivo general son los siguientes:

² <https://www.tweetbinder.com/es/>

³ <https://sproutsocial.com/es/>

⁴ <https://www.hootsuite.com/es/>

⁵ <https://supermetrics.com/>

- La herramienta debe permitir a sus usuarios:
 - acceder a métricas públicas y privadas de cuentas redes sociales, tales como el número de seguidores, alcance de publicaciones o número de vistas de los vídeos.
 - acceder a estadísticas de sus sitios web mediante Google Analytics 4, como por ejemplo, visitas o tiempo promedio de estadía en el sitio.
 - desplegar las métricas y estadísticas en tiempo real.
 - generar informes personalizados.
- La herramienta debe estar diseñada para ser utilizada por profesionales en las áreas de marketing o prensa.

1.3. Resumen de la Solución

La solución con que se da cumplimiento a los objetivos planteados consiste en una aplicación web llamada Laik. Ésta permite que los usuarios, una vez que hayan iniciado sesión, puedan acceder de manera sencilla a la información de redes sociales y sitios web, pudiendo desplegarse tanto en tiempo real como mediante informes en PDF.

La aplicación se organiza en servicios, cada uno de los cuales permite acceder a cierto tipo de información. Por ejemplo, existe un servicio de análisis de usuario de Twitter, en el que se ingresa un nombre de usuario como entrada, y la aplicación retorna las métricas correspondientes a dicho usuario de Twitter, en el formato solicitado (tiempo real o PDF). El usuario puede también seleccionar qué información le gustaría consultar, personalizarla (eligiendo, por ejemplo, el tipo de gráfico), o proporcionar autenticación para acceder a métricas restringidas.

Los usuarios pueden marcar ciertas métricas como favoritas, las cuales quedarán disponibles para acceso rápido en un dashboard, donde el usuario podrá juntar información proveniente de distintas fuentes, pudiendo actualizarla con un solo clic.

El software está desarrollado a partir de diversas tecnologías que trabajan de manera conjunta: React.js como motor de front-end, Flask como API interna que procesa las solicitudes, Celery como administrador de procesos pesados, Docker para armonizar las tecnologías, entre otras. Laik se encuentra públicamente disponible en la URL <https://laik.ligtech.cl/>.

1.4. Estructura del Documento

Este documento se estructura de la forma siguiente. A continuación, en el Capítulo 2 se expone el marco teórico asociado al proyecto, describiendo las políticas de privacidad de redes sociales y los métodos para conseguir los datos, y posibles alternativas de solución.

En el capítulo 3 se explica el levantamiento inicial de requisitos, el desarrollo del primer prototipo y el plan trazado para el desarrollo del producto. Además se indican y justifican las tecnologías escogidas para la implementación de la plataforma y los servicios a brindar por la misma.

En el capítulo 4 se presenta el diseño de plataforma, cuyos principales componentes son su arquitectura, el modelo de datos, el modelo de interfaces y el sistema de usuarios de la misma. En el capítulo 5 se describe la implementación de la plataforma Laik, particularmente su desarrollo de base, la implementación del sistema de manejo de usuarios, el desarrollo de los servicios comprometidos, y la puesta en producción y entrega del producto.

Luego, en el capítulo 6 se presentan y describen las principales interfaces del sistema, las cuales permiten el acceso a los servicios. El capítulo 7 describe el proceso de evaluación de la solución y los resultados obtenidos. Finalmente, el capítulo 8 presenta las conclusiones y el trabajo a futuro.

2. Marco Teórico

Este trabajo de memoria se enfoca en la recopilación e integración de datos de redes sociales y sitios web, en el contexto de diseño de estrategias de marketing de marcas comerciales. A continuación, en este capítulo se discute el estado actual del tratamiento de datos de las redes sociales, junto con los mecanismos para acceder a estos de manera sistemática. Posteriormente, se hace un análisis similar para los sitios web, y se discuten posibles mecanismos para abordar el problema de la dispersión de los datos.

2.1. Redes sociales

Según un estudio⁶ realizado en enero de 2022, a nivel mundial, las personas pasan alrededor de 147 minutos al día en redes sociales. Éstas han crecido hasta el punto de ser un elemento transversal en la sociedad, son una fuente muy rica en información de usuarios y han llegado a ser una parte importante en las vidas de las personas del mundo contemporáneo.

De la misma forma, las redes sociales resultan fundamentales para las marcas comerciales, las cuales pueden acceder a grandes volúmenes de su público objetivo de manera directa. En muchos casos, especialmente cuando se trata de productos digitales, la totalidad o la gran mayoría de los recursos de marketing deben enfocarse en las redes sociales. Conocer la eficiencia de las publicaciones y del comportamiento de los community managers resulta crucial para definir estrategias de marketing.

A día de hoy, compañías de todo tipo, ya sea antiguas o recientes, están enfrentando un gran número de desafíos para integrarse de lleno al mundo digital. Costos que aumentan, presupuestos que disminuyen, complejidades crecientes y un aumento de demanda de servicios digitales son algunos de estos desafíos, los cuales deben abordarse para llegar a una óptima toma de decisiones (Custy, 2007).

2.2. Tratamiento de los datos

Cuando se habla de datos de redes sociales, en general, el término se refiere a las métricas e información demográfica de las redes sociales, recopilados a través de herramientas de análisis en las plataformas sociales, y a través de las acciones mismas de los usuarios. También, puede hacer referencia a los datos recopilados del contenido que las personas publican en cada una de las redes sociales.

La forma en que las redes sociales manejan sus datos no ha estado exenta de debate, puesto que la línea entre información pública y privada es a veces difusa. La facilidad de uso de los smartphones y la proliferación de redes móviles de alta velocidad, han facilitado una cultura de compartir contenido de manera espontánea y despreocupada. Si bien un usuario puede conocer el nivel de privacidad de su contenido, a gran escala es muy difícil

⁶ <https://datareportal.com/reports/digital-2022-global-overview-report>

discernir, por lo que la extracción de datos de manera sistemática suele cruzar la línea hacia lo privado (Smith, 2012). Sin embargo, éste es un problema que finalmente recae en la responsabilidad de los usuarios, pues las redes sociales cuentan con políticas de privacidad que son aprobadas por los usuarios al momento de registrarse.

La recopilación de estos datos también brinda información clave sobre la audiencia. Se puede saber qué tipo de contenido les gusta, cuándo quieren verlo y dónde pasan su tiempo en línea. Estos datos son fundamentales para poder orientar el contenido hacia dicho foco, y así aumentar el alcance de la marca.

2.3. Acceso sistemático vía APIs

Con el fin de acceder a dicha información para fines particulares, como promocionar marcas y servicios, muchas aplicaciones están actualmente siendo diseñadas de tal forma que tengan la capacidad de establecer una conexión con la API (Application Programming Interface, por sus siglas en inglés) de dichas redes. Una API, en este contexto, consiste en un conjunto de rutinas que provee acceso a funciones de un determinado software. Estas son diseñadas y publicadas por los mismos sitios proveedores de dichos servicios, con el fin de permitir el acceso sistemático y programado a ciertos datos específicos. Cada red social cuenta con su(s) API(s), que permiten acceder a los datos mediante peticiones HTTP, u otro tipo de clientes.

En el caso de Twitter, esta plataforma cuenta con una amplia variedad de APIs que permiten acceder a distintos tipos y volúmenes de información. La principal es Twitter API v2⁷, la cual es gratuita pero cuenta con restricciones de consultas. Esta permite acceder a información reciente de Twitter, por ejemplo, los últimos tweets de una búsqueda o de un usuario, las métricas públicas de un usuario (número de seguidores, de tweets, etc.) o las métricas privadas de un usuario (alcance, número de impresiones, etc.). Entre las restricciones, destacan el límite de poder extraer únicamente los 3200 tweets más recientes de cada resultado, o el tope máximo de dos millones de tweets al mes por aplicación. Además, el número de solicitudes está limitado por intervalos de tiempo, impidiendo que la API se sature.

Si bien estas funcionalidades son suficientes para construir una aplicación que haga uso de esta API para extraer información y tomar decisiones, eventualmente puede ser necesario recurrir a tweets más antiguos, o a volúmenes más grandes de información. Para ello, Twitter provee a los desarrolladores de APIs adicionales⁸, las cuales permiten realizar consultas de mayor magnitud. Estas APIs son de pago, puesto que significan una inversión importante de recursos para Twitter, debido a los grandes volúmenes de información con los que trata.

Facebook, por su parte, cuenta con Facebook Graph API, la cual es gratuita. Este es el método primario para que las aplicaciones puedan acceder a datos de Facebook. Además, todos los SDKs (kits de desarrollo de software, por sus siglas en inglés) y productos de

⁷ <https://developer.twitter.com/en/docs/twitter-api>

⁸ <https://developer.twitter.com/en/docs/twitter-api/enterprise>

Facebook interactúan directamente con esta API, siendo todas las demás APIs extensiones de esta. Dentro del resto de APIs, destacan principalmente Marketing API⁹, que permite integrar las funciones de marketing y anuncios, y Pages API¹⁰, que permite acceder a datos de páginas de Facebook.

Además, debido a que Instagram es propiedad de Meta¹¹, sus APIs también dependen directamente de Facebook. Estas son Instagram Graph API¹², que permite acceder a información de cuentas de tipo Business o Creator, e Instagram Basic Display API¹³, que permite el acceso a cuentas personales, aunque con métricas limitadas. Otras redes sociales, como YouTube, TikTok, LinkedIn o Reddit, también cuentan con APIs para acceder a sus datos.

2.4. Estadísticas de sitios web

De manera similar a las métricas de redes sociales, los sitios web también cuentan con información valiosa relativa a la actividad de los usuarios. Para todo administrador de algún sitio web, o proveedores de información online, hay básicamente tres puntos principales para monitorear al consumidor digital: historial de accesos, contabilidad interna y seguimiento (Clark, 2014).

Hoy en día, la necesidad de conocer el estado de dichos puntos principales, hace que sea necesario un método sistemático para recopilar la información. Si bien existen variados métodos para hacer esta labor, el proveedor por excelencia de servicios de métricas de sitios web es Google, mediante el servicio de Analytics. Hasta octubre de 2020, el servicio de Universal Analytics¹⁴ era el principal motor de monitoreo de actividad en sitios web, pero ha sido reemplazado por Google Analytics 4¹⁵.

Con Google Analytics 4, servicio completamente gratuito, es posible definir métricas personalizadas, permitiendo al usuario combinar la información recopilada, y presentarla en algún formato conveniente al caso. Además, se cuenta con Google Analytics Data API, método que permite extraer la información de los sitios web, e integrarla en todo tipo de aplicaciones.

2.5. Dispersión de los datos

El problema que se observa es que existen muchas fuentes de información para acceder sistemáticamente a datos provenientes de redes sociales y sitios web. Cada una de las APIs mencionadas anteriormente implican una curva de aprendizaje que debe ser

⁹ <https://developers.facebook.com/docs/marketing-apis/>

¹⁰ <https://developers.facebook.com/docs/pages/>

¹¹ <https://www.facebook.com/help/instagram/581066165581870>

¹² <https://developers.facebook.com/docs/instagram-api/>

¹³ <https://developers.facebook.com/docs/instagram-basic-display-api/>

¹⁴ <https://support.google.com/analytics/answer/2790010?hl=es>

¹⁵ <https://support.google.com/analytics/answer/10089681?hl=es>

considerada por los encargados de marketing de las marcas comerciales. Además, estas herramientas están diseñadas para ser utilizadas por desarrolladores.

La solución a este problema de dispersión de los datos, junto con la puesta a disposición de estos en una plataforma amigable para usuarios no familiarizados con programación, corresponde a una aplicación que permita consultar estos datos de manera intuitiva. Esta puede ser dedicada (por ejemplo, a través de un plugin en la interfaz de administración de un sitio web¹⁶), o generalizada y ofrecida como servicio, en forma de aplicación web o móvil, como los ejemplos mencionados en la introducción. Este trabajo de memoria busca abordar este problema a través de la segunda solución, permitiendo al usuario seleccionar la información que le interesa, y consultarla desde cualquier dispositivo a través de una URL con tan solo un click.

¹⁶ <https://wordpress.org/plugins/ga-google-analytics/>

3. Concepción de la Solución

Como se mencionó antes, para cumplir con los objetivos planteados se desarrolló la plataforma web llamada *Laiik*. Esta plataforma permite a los clientes acceder, desde cualquier lugar o dispositivo, a estadísticas de distinto tipo sobre su marca (redes sociales, prensa y sitios web) agrupadas en una única plataforma.

A continuación se presenta el detalle de la planificación del desarrollo, comenzando por el levantamiento de requisitos por parte de LIG Tech, seguido por el primer prototipo desarrollado a modo de prueba de concepto. Posteriormente, se describe el plan de desarrollo y la arquitectura de la solución.

3.1. Levantamiento inicial de requisitos

La primera reunión con LIG Tech se llevó a cabo el día 7 de junio de 2022, en la que se planteó la idea de desarrollar una plataforma, a través de la cual los encargados de marketing de una marca pudiesen acceder a todo tipo de métricas de sus redes sociales, tanto en tiempo-real como mediante informes personalizables.

En reuniones posteriores, se fueron acotando algunos detalles adicionales, que darían forma a los requisitos finales. La plataforma sería una aplicación web, accesible desde cualquier dispositivo, y se organizaría mediante servicios. Estos consistirán en rutinas que permiten recopilar cierto tipo de información, como por ejemplo, analizar una cuenta de Twitter, o una página de Facebook. Adicionalmente, se decidió incorporar el análisis de estadísticas de sitios web (mediante Google Analytics 4), junto al de redes sociales.

3.2. Desarrollo del primer prototipo

Tras las primeras reuniones, y a medida que se definían los requisitos finales, se trabajó en el desarrollo de un primer prototipo de la aplicación, que serviría como prueba de concepto y una primera aproximación a las integraciones necesarias para acceder a los datos de redes sociales. Este prototipo se desarrolló usando el framework Flask, tanto en back-end como en front-end.

La aplicación consistió en un sistema de usuarios sencillo, y dos servicios: análisis por usuario en Twitter y en Instagram. Para esto, fue necesario incorporar los flujos de autenticación de Twitter y Facebook, y asociar los códigos de acceso al usuario correspondiente mediante una base de datos local.

3.3. Plan de desarrollo del producto

Para planificar el desarrollo del producto, posterior al primer prototipo, se organizó el tiempo mediante una carta Gantt, la cual se observa a continuación en la Figura 1. Se contemplaron 15 semanas para el desarrollo.

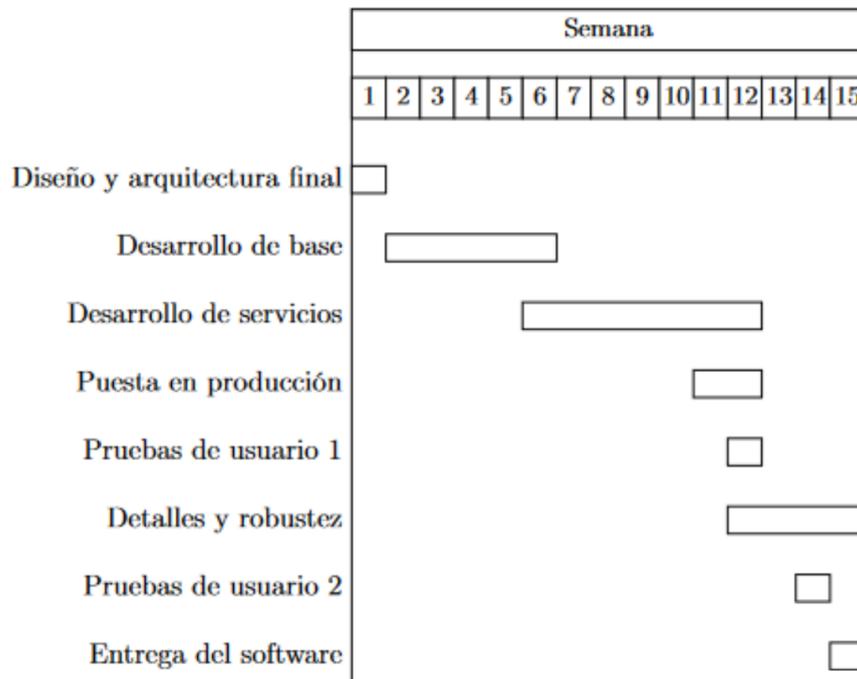


Figura 1: Plan de desarrollo del producto.

La primera semana se destina para afinar los detalles finales del diseño (mediante mock-ups), y para definir las tecnologías a utilizar, que darán forma a la arquitectura descrita en la subsección siguiente. Posteriormente, se destinan cinco semanas para el desarrollo de la base de la plataforma, esto es, los sistemas de usuarios, administración, y la base de los servicios, junto con las características principales del front-end. Luego, hasta la semana N°12 se contempla el desarrollo de los servicios, tras lo cual se tendría un producto funcional y completo.

Una vez que el desarrollo se encuentra en sus etapas finales, se procede a pasar a producción, dejando la plataforma en el servidor proporcionado por LIG Tech, accesible mediante la URL <https://laik.ligtech.cl>. Tras encontrarse accesible, la plataforma es sometida a dos rondas de pruebas de usuarios, descritas en detalle en el capítulo 6. Entre estas rondas de pruebas de usuario, y la última semana, se realizarán las correcciones finales tomando en consideración la retroalimentación recibida. Finalmente, durante la última semana, se realiza la entrega del software a LIG Tech.

A continuación se indican las primeras definiciones que hubo que tomar previo al proceso de desarrollo, tales como las tecnologías a utilizar y los servicios que se ofrecerán al usuario.

3.4. Tecnologías a utilizar

En primer lugar, se requirió contar con un cliente y un servidor, los cuales se debían comunicar mediante el protocolo HTTP. Debido a que la plataforma contaría con múltiples servicios, se decidió que del lado de servidor se contará con una API interna, encargada de recibir y procesar todas las peticiones del cliente. Para esto, se optó por el framework Flask¹⁷ para el desarrollo de esta API, pues cumple con todas las necesidades al ser uno de los frameworks de desarrollo de APIs por excelencia, siendo altamente escalable, flexible y fácil de usar. Dentro de las alternativas también estaban Django¹⁸ y FastAPI¹⁹, pero se optó por Flask debido a que está más orientado a microservicios en comparación a Django (y, además, es más minimalista en términos de código), y porque la comunidad de desarrolladores es extensa, contando con gran variedad de preguntas ya resueltas en Internet, a diferencia de FastAPI.

La aplicación de Flask es servida por Gunicorn²⁰, servidor WSGI HTTP para Unix, que se encarga de la comunicación con el cliente. La plataforma cuenta además con algunas funciones que podrían tardar tiempo considerable en ejecutarse, tales como la generación de informes o las consultas a las APIs de redes sociales.

Para manejar estos procesos pesados, se hace uso de Celery²¹. Esta herramienta, rápida, flexible y sencillamente integrable con Flask, permite organizar los procesos pesados, los cuales se ejecutan de manera asíncrona, liberando así a los *workers* de Gunicorn y evitando que la experiencia del usuario cuente con tiempos de espera muy largos. Para hacer uso de Celery, éste a su vez requiere de Redis²², motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes.

El cliente, por otro lado, fue desarrollado usando el framework React.js²³. La necesidad de utilizar un framework de front-end se justifica, ya que muchos de los componentes de la aplicación se utilizarán en diversas interfaces, variando muy poco su contenido, por lo que usar Jinja²⁴ para servir plantillas HTML directamente con Flask, implicaría tener que escribir mucho más código, lo que se traduce en más tiempo y más flancos para errores. Otras alternativas de frameworks fueron Angular.js²⁵ y Vue.js²⁶, pero en el caso de Angular.js la curva de aprendizaje es más alta (lo que se traduce en mayor tiempo de desarrollo), mientras que Vue.js cuenta con una comunidad de desarrollo más reducida que la de React.js. De todas formas, el cliente es una plataforma cuyo funcionamiento es independiente del framework, pues su función es traducir los llamados a la API de Flask a una interfaz amigable para el usuario. Por lo tanto, podría ser reemplazado, por ejemplo, si React.js dejase de ser gratuito.

¹⁷ <https://flask.palletsprojects.com/en/2.2.x/>

¹⁸ <https://www.djangoproject.com/>

¹⁹ <https://fastapi.tiangolo.com/>

²⁰ <https://gunicorn.org/>

²¹ <https://docs.celeryq.dev/en/stable/>

²² <https://redis.io/>

²³ <https://reactjs.org/>

²⁴ <https://jinja.palletsprojects.com/en/3.1.x/>

²⁵ <https://angular.io/>

²⁶ <https://vuejs.org/>

Para agrupar y desplegar correctamente cada uno de los elementos mencionados anteriormente, se usa Docker²⁷ (junto con docker-compose). Éste permite estandarizar las operaciones de las aplicaciones, transferir el código con facilidad, y pasar de un entorno de desarrollo a uno de producción con sencillez. Además, facilita la comunicación entre los distintos componentes.

Por otra parte, se cuenta con cuatro contenedores: la API interna, el cliente, el worker de Celery, y Redis. Además, se hace uso de un volumen, para algunos archivos con información variable, principalmente de los usuarios.

Como en la mayoría de los sistemas de información, la información más sensible se encuentra en una base de datos relacional. Por lo tanto, para asegurar la integridad de los datos, se optó por mantener la base de datos fuera del esquema de Docker. Respecto de la tecnología del motor de base de datos, se hace uso de PostgreSQL²⁸, debido a su robustez. Otra opción fue MySQL²⁹, pero PostgreSQL cuenta con un ligero mejor manejo de la concurrencia.

Se hace uso de Nginx³⁰ como servidor web, y como enlace entre la API interna y el cliente. Apache2³¹ habría sido también una opción viable, pero debido a experiencia previa con Nginx, se optó por esta opción. Finalmente, se utiliza Git³² como herramienta de versionamiento, junto con GitHub como nube.

3.5. Servicios a brindar por la plataforma

Habiendo definido las tecnologías a usar, se procede a indicar específicamente cuáles son los servicios que ofrece la aplicación. Se acordó que los servicios prioritarios (es decir, los que se incluirían en el producto mínimo viable) fueran los siguientes:

- *Dashboard*: Plataforma en que el usuario podrá consultar por las métricas que ha marcado como favoritas a lo largo de los demás servicios, pudiendo combinar múltiples servicios y consultas.
- *Análisis de usuario de Twitter*: Una vez que el usuario proporcione una cuenta de Twitter, se harán llamados a la API oficial (Twitter API v2) para conseguir y desplegar información relativa a dicha cuenta, como por ejemplo, el número de seguidores, sus likes recibidos durante los últimos días, y una tabla con sus últimos tweets. El usuario podrá además autenticar cuentas, contando así con acceso a métricas exclusivas, como el alcance de sus tweets.
- *Análisis de usuario de Instagram*: De manera análoga al servicio anterior, el usuario podrá proporcionar cuentas de Instagram, y consultar métricas relativas a estas

²⁷ <https://www.docker.com/>

²⁸ <https://www.postgresql.org/>

²⁹ <https://www.mysql.com/>

³⁰ <https://www.nginx.com/>

³¹ <https://httpd.apache.org/>

³² <https://git-scm.com/>

mediante la API oficial: Instagram Graph API. Igualmente, en caso de autenticar una cuenta, contará con acceso a métricas exclusivas de ésta.

- *Análisis de página de Facebook*: En este servicio el usuario podrá proporcionar páginas de Facebook (mediante su respectiva URL), y consultar métricas relativas a esta usando la API oficial: Facebook Graph API. También, en caso de proporcionar autenticación, se contará con métricas exclusivas.
- *Análisis de propiedad de Google Analytics 4*: Este servicio consistirá en el despliegue de métricas relativas a una propiedad, es decir, un sitio web. Éste debe estar previamente registrado en el servicio de Google Analytics 4, y el usuario debe proporcionar acceso a Laik en la plataforma de Google. Los datos se extraerán haciendo uso de la API oficial (Google Analytics Data API).

4. Diseño de la Plataforma

En este capítulo se describen los principales componentes de la solución, desde la arquitectura y el modelo de datos, hasta el modelo de interfaces de la plataforma. Además, se describen los mecanismos de manejo de usuarios y de datos.

4.1. Arquitectura de la solución

La arquitectura de la aplicación involucra a cuatro contenedores de Docker: el *cliente*, la *API interna*, *Redis* y *Celery*. Además, se cuenta con una base de datos relacional PostgreSQL, no dependiente de los contenedores. Los contenedores y la base de datos se encuentran en un único servidor, en el cual se sirve la aplicación mediante Nginx. En la Figura 2 se puede apreciar un diagrama que resume esta arquitectura.

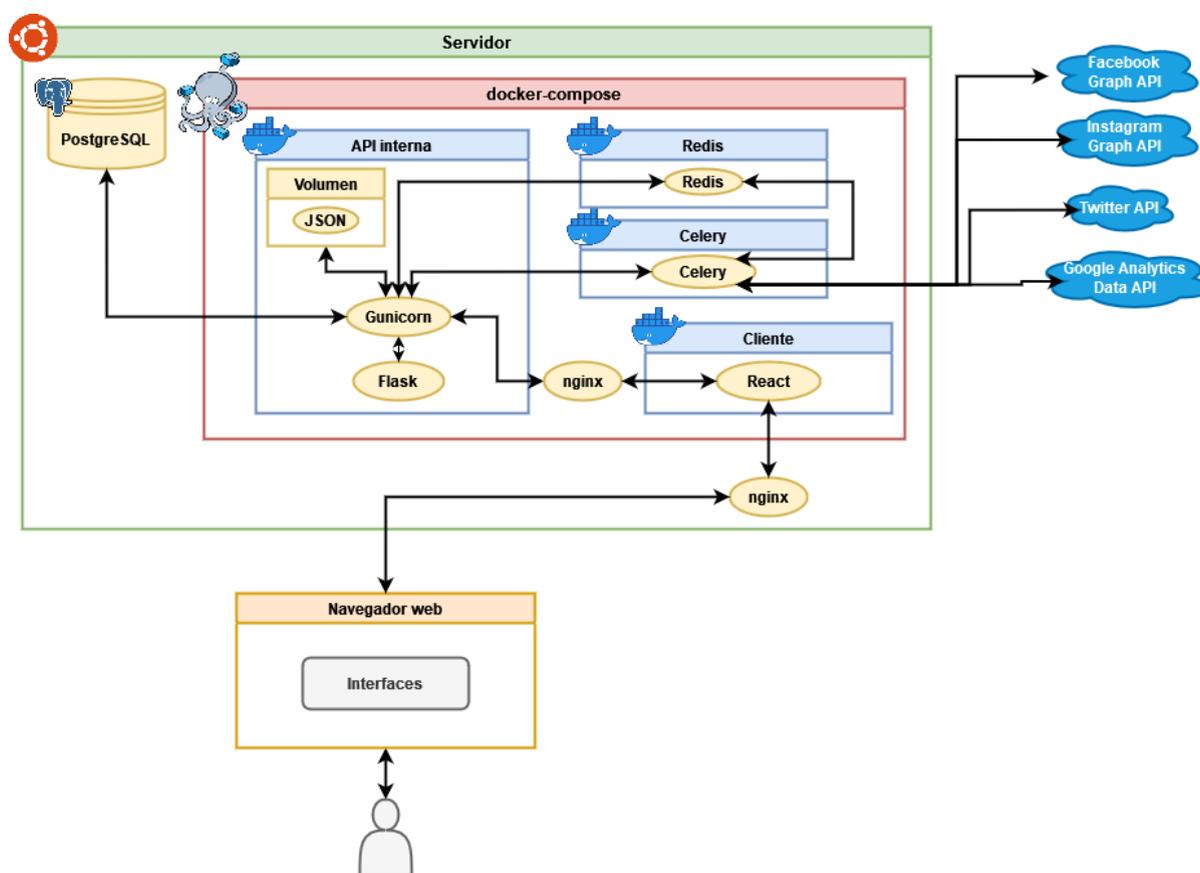


Figura 2: Estructura general de la aplicación.

El cliente (escrito en React) se encarga de administrar las interfaces, y mostrar al usuario las vistas que corresponden en su navegador web. La API interna, por otro lado, es una aplicación en Flask, servida con Gunicorn, que se encarga de procesar las solicitudes hechas por el cliente, y responder con la información correspondiente. Celery se utiliza como administrador de procesos pesados (como algunos de los servicios),

mientras que Redis se encarga de almacenar las sesiones de los usuarios, y de proporcionar servicios auxiliares a Celery.

4.2. Modelo de datos

Laik requiere hacer uso de una gran variedad de información, tanto acerca de los usuarios como de la aplicación, y en algunos casos de carácter sensible. A continuación se describen brevemente los distintos recursos de datos que utiliza la plataforma.

4.2.1. Modelo relacional

Para almacenar la información sensible, relativa a los usuarios, se hace uso de una base de datos relacional. Ésta cuenta con tres tablas (Figura 3): una relativa a los usuarios, otra relativa a los tokens de acceso de Facebook, y otra relativa a los tokens de acceso de Twitter.

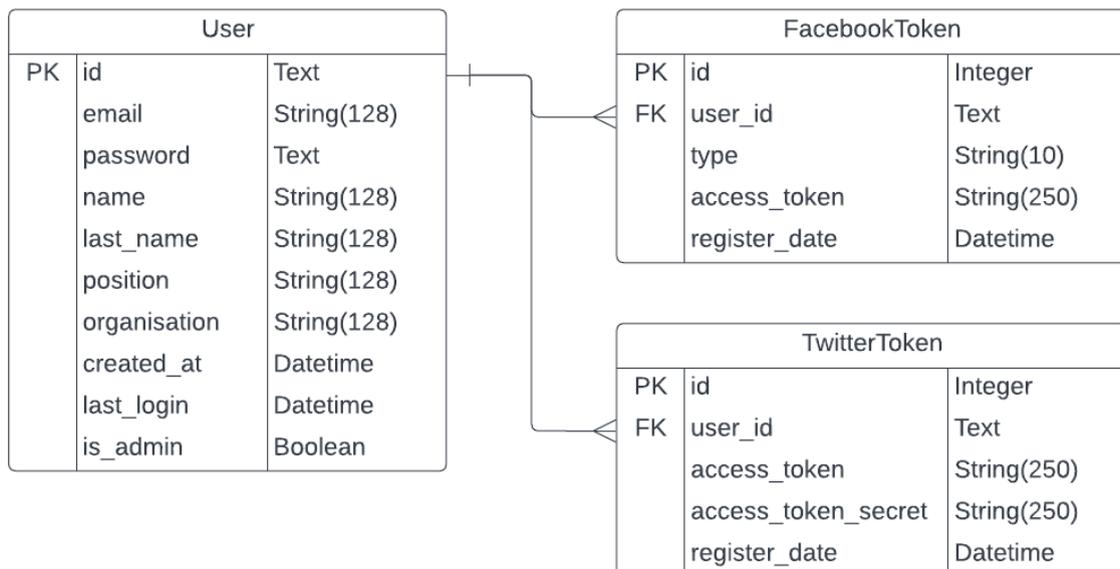


Figura 3: Modelo de datos relacional de la aplicación.

4.2.2. Archivos estáticos

Los usuarios también cuentan con información adicional relativa a los servicios; dado que esta información es altamente variable, se almacena en archivos en formato JSON. Cada usuario está asociado a tres archivos:

- *authentication.json*: Contiene información acerca de todas las cuentas autenticadas del usuario, para cada servicio: etiquetas, IDs y mapeo a los tokens almacenados en la base de datos relacional.

- *jobs.json*: Contiene información sobre los servicios solicitados por el usuario: historial, procesos en ejecución y procesos programados.
- *preferences.json*: Registra las preferencias del usuario sobre cada servicio: favoritos, métricas a mostrar, tipo de gráfico, colores, etc. También contiene el estado de las restricciones de cada usuario, como el límite de tweets.

Además, se cuenta con un archivo *globals.json* que contiene información de carácter global, que afecta a todos los usuarios. Dicho archivo mantiene, entre otras cosas, el estado de las restricciones globales de las APIs, o los códigos válidos de reinicio de contraseña.

4.2.3. Caché

Cada usuario cuenta con archivos JSON de caché para cada servicio. En estos archivos se almacena el resultado de la última consulta realizada, junto con metadatos como la fecha y hora. De esta forma, cuando el usuario acceda a un servicio, se mostrarán los últimos resultados obtenidos sin necesidad de consultarlos de nuevo. Las consultas programadas no se mantienen en caché.

4.2.4. Últimos informes

Cada usuario está asociado a un directorio en el que se almacenan los últimos 100 informes solicitados por éste, dentro de los últimos 60 días, con un tope máximo de 1 GB. Estos quedan disponibles para descarga, y se despliegan junto a los metadatos almacenados en *jobs.json*.

4.2.5. Archivos temporales

Finalmente, cada usuario cuenta con un directorio de archivos temporales, en el que se almacenan las imágenes relativas a las consultas realizadas en los distintos servicios. Esto se hace para poder desplegarlas en cualquier navegador, sin necesidad de enlaces externos. Además, permite un procesamiento más rápido de éstas al momento de generar los informes en PDF. Los archivos temporales de cada servicio, al estar asociados directamente a un resultado almacenado en caché, son borrados y reemplazados con cada consulta.

4.3. Diseño de interfaces de usuario

Previo a comenzar el desarrollo de la aplicación cliente, se realizó una sesión de análisis de mockups para delinear los principales componentes de las interfaces. Se definió que las principales interfaces serían las siguientes:

- *Landing*: Portada de la aplicación, con enlaces al Dashboard, a los documentos de Política de Privacidad y Términos y Condiciones, y al sitio web de LIG Tech. Además, ésta debe contar con un formulario de contacto.

- **Documentos:** Interfaz minimalista en que se muestra el PDF de cada documento. Incluye:
 - *Política de Privacidad.*
 - *Términos y Condiciones.*
- **Iniciar sesión:** Formulario de inicio de sesión, que cuenta con un botón para solicitar la recuperación de la contraseña.
- **Servicios:** Este es un conjunto de las principales interfaces de cada servicio y tienen un diseño ad hoc a cada uno de ellos. En caso de acceder a alguna de estas interfaces sin haberse autenticado, la plataforma redirigirá al usuario al formulario de inicio de sesión. Estas interfaces cuentan con una barra de navegación al lado izquierdo, y el contenido principal al lado derecho. Los servicios a acceder a través de estas interfaces son los siguientes:
 - **Dashboard:** servicio principal, en que se pueden consultar las métricas que el usuario ha marcado como favoritas a lo largo de todos los demás servicios.
 - **Informes:** servicio que permite ver el historial de informes solicitados, permitiendo su descarga. También muestra informes pendientes y programados.
 - **Análisis por usuario de Twitter:** servicio que permite consultar métricas de una cuenta de Twitter.
 - **Análisis por usuario de Instagram:** servicio que permite consultar métricas de una cuenta de Instagram.
 - **Análisis por página de Facebook:** servicio que permite consultar métricas de una página de Facebook.
 - **Análisis por propiedad de Google Analytics 4:** servicio que permite consultar métricas de una propiedad de Google Analytics 4.
 - **Configuración de la cuenta:** formularios que permiten modificar los atributos de cada usuario, y cambiar la contraseña.
- **Administración:** Interfaz exclusiva para el usuario administrador, en donde puede crear y eliminar usuarios. También cuenta con una opción para enviar correos electrónicos a los usuarios seleccionados.

En la Figura 4 se observa un diagrama que detalla cómo las interfaces se relacionan entre sí, y los caminos que podría seguir el usuario.

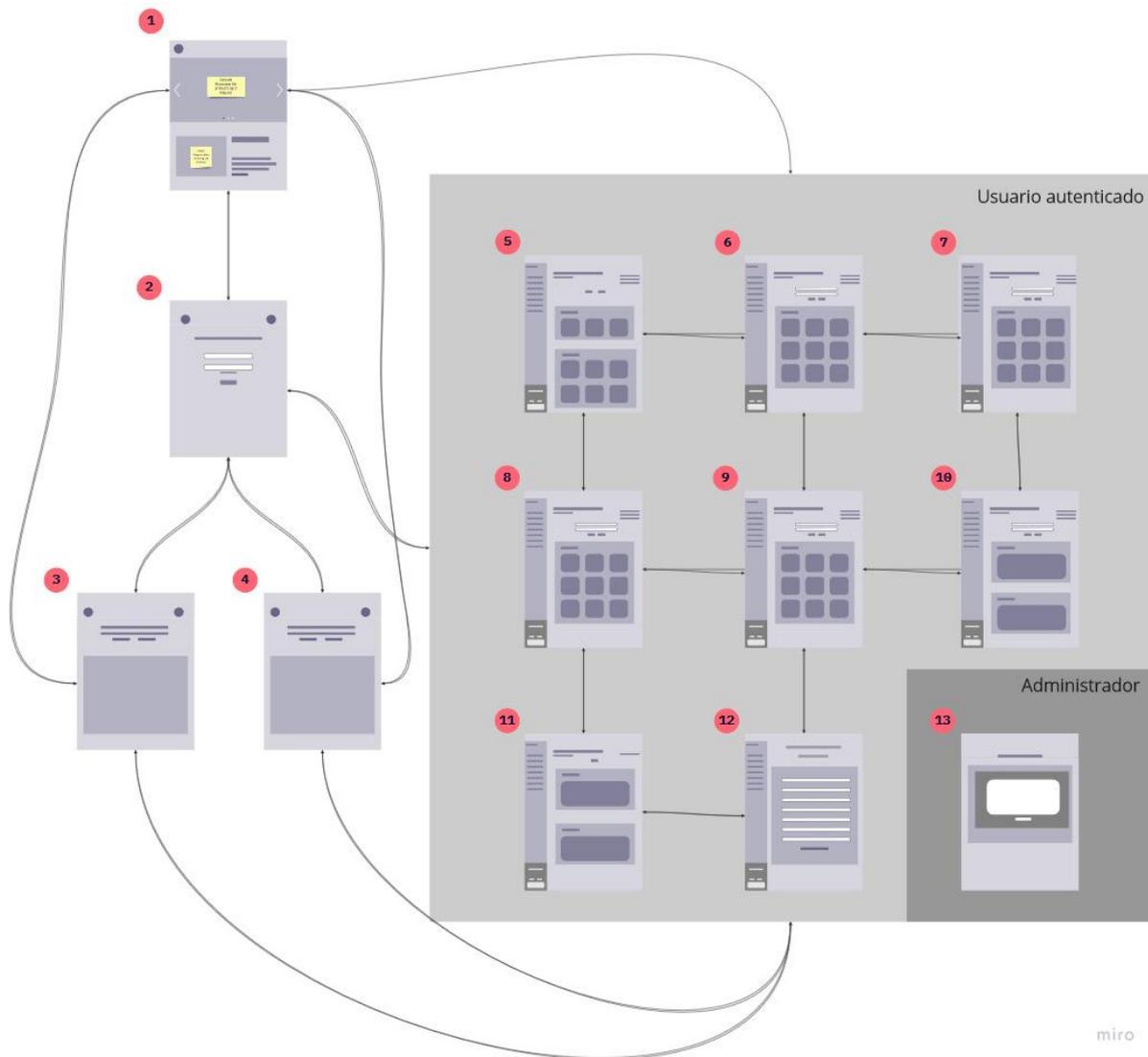


Figura 4: Conexiones entre las interfaces.

Al ingresar a la URL, el usuario llega al landing (rótulo 1, Fig. 4, desde el cual puede acceder a los servicios (rótulos 5-12) mediante el dashboard (5). En caso de no encontrarse autenticado, éste será redirigido al formulario para iniciar sesión (2) donde, tras autenticarse, será redirigido al dashboard.

El acceso a los documentos de Política de Privacidad (3) y Términos y Condiciones (4) no requiere autenticación, y por lo tanto estos pueden ser accedidos desde cualquier vista. La interfaz de administración (13) sólo puede ser accedida desde la URL, contando con autenticación del usuario administrador. De caso contrario, se mostrará un mensaje denegando el acceso, junto con enlaces al landing y al dashboard. Desde un servicio se puede acceder a cualquier otro mediante la barra lateral de navegación. La tonalidad del fondo gris ilustra los tipos de usuarios que tienen acceso a cada interfaz.

4.4. Sistema de manejo de usuarios

Dado que la aplicación requiere almacenar información sensible de los usuarios (relativa a la autenticación de cuentas), y que el acceso a ésta es restringido, se requiere de un sistema de usuarios. Éste funciona a través del módulo Flask-Session, el cual almacena una cookie del lado del cliente al momento de iniciar sesión.

Estos usuarios se identifican con un correo electrónico y una contraseña. Además, cuentan con atributos, tales como nombre, apellido, organización, posición, logotipo, etc. Todos estos atributos son modificables por los usuarios, con la excepción del correo electrónico. En caso de que el usuario no recuerde su contraseña, puede proporcionar su correo electrónico y, en caso de existir el usuario, se le enviará un enlace para restablecer la contraseña.

Al momento de iniciar la aplicación, se crea un usuario administrador. Éste tiene acceso a una interfaz exclusiva, en la cual puede crear y borrar otros usuarios (exceptuando a sí mismo). Queda exclusivamente en manos del administrador la creación de usuarios, por lo que la aplicación no cuenta con formulario de registro. El usuario administrador se diferencia de un usuario normal únicamente en el acceso a dicha interfaz de administración, como se indica en la Figura 4.

5. Implementación de la Plataforma

El proceso de implementación de la plataforma involucró las macro-actividades definidas en el plan de trabajo; es decir, el desarrollo base, el desarrollo de los servicios, puesta en producción del producto y la entrega del producto. A continuación se describen cada una de estas macro-actividades.

5.1. Desarrollo de base

Una vez definidas las tecnologías a utilizar, la arquitectura de la plataforma, el sistema de usuarios, el modelo de datos y de interfaces, corresponde comenzar con el desarrollo base del producto. A continuación se explican las actividades realizadas.

5.1.1. Creación del proyecto

En primer lugar, debe crearse el proyecto. Para esto, se crea un repositorio con *Git*, junto con su versión remota en *GitHub*. Se incorpora una licencia de tipo MIT, y otros archivos tales como un *README.md* que contendrá documentación básica, y un *.gitignore*, creado con la plantilla de *Python*.

Posteriormente, se crean dos maquetas vacías: un proyecto de *Flask*, y un proyecto de *React.js*, que corresponden al back-end y al front-end respectivamente. La aplicación en *Flask* se construyó archivo por archivo, mientras que en el caso del front-end, se hizo uso de la herramienta *create-react-app*, que crea todos los archivos necesarios para iniciar una aplicación de *React.js*.

5.1.2. Armonización de tecnologías

Posteriormente, es necesario asegurar que todos los componentes y tecnologías se puedan comunicar entre sí de la manera deseada. Es aquí donde se realiza la *dockerización* del proyecto, incorporando tres archivos: *Dockerfile.back*, *Dockerfile.front* y *docker-compose.yml*.

El punto de entrada de la aplicación es *docker-compose.yml*, en el cual se definen los contenedores y el volumen a utilizar. *Dockerfile.back* se encarga de construir la aplicación de Flask con Gunicorn, haciendo uso de la imagen de Python. Por otro lado, *Dockerfile.front* cumple dos funciones: en primer lugar, construye la aplicación de React.js mediante la imagen de Node.js, y luego, usando la imagen de Nginx, enlaza la aplicación de Flask con la aplicación de React.js, haciendo uso de un proxy reverso.

En primer lugar, es necesario construir cada componente; para ello, el archivo *docker-compose.yml* realiza las siguientes acciones:

1. Usando la imagen de *Redis*, monta un contenedor para este servicio.
2. Usando las configuraciones de *Dockerfile.back*, se construye la API interna de *Flask*. Esta se asocia a *Redis* para el manejo del sistema de usuarios.
3. Posteriormente, usando el mismo archivo, se construye el *worker* de *Celery* asociado a la API de *Flask* y a *Redis*.
4. Finalmente, usando *Dockerfile.front*, se construye un *bundle* de producción para la aplicación de *React.js*, y se enlaza a la API de *Flask* mediante *Nginx*.

Una vez construido todo, se inicializa cada contenedor. Tras finalizar este proceso, Laik queda expuesta en <http://localhost:3000>. Es importante mencionar que, tal como se mencionó antes, la base de datos no forma parte de la *dockerización*, a pesar de habitar el mismo servidor. Por lo tanto, la aplicación funciona independiente de ésta. Sin embargo, no será posible iniciar sesión en caso de que la base de datos no esté conectada.

5.1.3. Navegación

Inicialmente, la aplicación muestra una única vista por defecto. Sin embargo, se definen las vistas y se crean páginas en blanco para cada una de las siguientes interfaces:

- /
- /login
- /admin
- /privacy
- /terms
- /reset_password
- /account_settings
- /dashboard
- /reports
- /services/twitteruser
- /services/facebookpage
- /services/instagramuser
- /services/ga4

Luego, siguiendo el diagrama en la Figura 4, se agregaron enlaces de navegación entre las distintas interfaces. Al mismo tiempo, se incorporó la barra de navegación de los servicios. Además, se condicionan todas las URL de la forma `'/api/*'`, para que en vez de ser respondidas por el cliente de *React.js*, sean procesadas por la API interna de *Flask*. Tal como se mencionó antes, esto se hace a través del proxy reverso.

5.1.4. Implementación del sistema de manejo de usuarios

Una vez implementada la base de las interfaces, es momento de incluir el sistema de usuarios, para que de esa manera se pueda bloquear las vistas que son exclusivas para usuarios registrados. Para ello, se hace uso del módulo *Flask-Session*. Éste almacena las sesiones de los usuarios usando *Redis* por parte del servidor, y una cookie de lado del navegador.

Haciendo uso de este mecanismo, cada vez que se quiera acceder a una interfaz exclusiva, se hace una llamada a un endpoint de la API interna que verifica la existencia de la sesión. De comprobarse, se permite el acceso, de caso contrario, se redirecciona al usuario a la página para iniciar sesión. Las interfaces restringidas son las siguientes:

- /admin
- /account_settings
- /dashboard
- /reports
- /services/twitteruser
- /services/facebookpage
- /services/instagramuser
- /services/ga4

En caso de acceder a /login estando ya autenticado, se redirecciona a /dashboard. Un principio análogo se usa para la interfaz de administración, la cual verifica que el usuario se encuentre autenticado y sea el administrador, y solo en dicho caso la plataforma permite el acceso.

Debido a que la Laik no cuenta con formulario público de registro, los usuarios deben ser creados a través de la interfaz de administración. La aplicación cuenta con una rutina para configurar la base de datos, de forma tal que se crean las tablas en caso de no existir, y genera un usuario administrador. Asimismo, desde esta interfaz el administrador podrá modificar y eliminar usuarios, y enviarles mensajes vía correo electrónico, entre otras cosas.

5.2. Tareas livianas

Con las interfaces conectadas entre sí, se procede a implementar algunas funcionalidades internas de la aplicación, previo a comenzar a desarrollar los servicios propiamente tal. Estas funcionalidades, en su gran mayoría, consisten en un endpoint de la API de Flask, pues requieren recurrir a información hospedada del lado del servidor, mediante consultas a la base de datos o a los archivos JSON.

A continuación, en la Tabla 1 se describen algunos de los endpoints más importantes de la API interna de Flask. Todos estos procesos son considerados livianos, pues tardan menos de 1 segundo en ejecutarse. Estos son manejados por Gunicorn, asegurando que la concurrencia de solicitudes de múltiples usuarios sea manejada exitosamente.

Tabla 1: Algunos endpoints de la API interna de Flask.

Endpoint	Tipo	Permisos necesarios	Función	Cuándo se llama
Iniciar sesión	POST	Ninguno	Verificar que el email y la contraseña proporcionados sean válidos. De ser así, se crea una sesión.	Tras clickear el botón de iniciar sesión.

Cerrar sesión	POST	Usuario	Elimina la sesión del usuario.	Tras clicar el botón de cerrar sesión.
Obtener perfil	GET	Usuario	Obtener los campos del usuario autenticado.	Tras cargar una interfaz de algún servicio.
Actualizar perfil	POST	Usuario	Actualizar los campos del usuario autenticado con los datos proporcionados.	Tras clicar el botón para actualizar perfil en la página de configuración de cuenta.
Cambiar contraseña	POST	Usuario	Cambiar la contraseña del usuario, si la contraseña original proporcionada es correcta.	Tras clicar en el botón para cambiar contraseña en la página de configuración de cuenta.
Obtener preferencias	GET	Usuario	Obtener el archivo JSON de preferencias.	Tras cargar la interfaz de algún servicio.
Actualizar preferencias	POST	Usuario	Actualizar el archivo JSON de preferencias.	Tras clicar en 'Actualizar' en cualquier formulario de algún modal de preferencias.
Obtener cuentas	GET	Usuario	Obtener el archivo JSON de autenticación.	Tras cargar la interfaz de algún servicio.
Actualizar cuentas	POST	Usuario	Actualizar el archivo JSON de autenticación.	Tras completar el proceso de autenticación de cuenta en algún servicio, o eliminar alguna cuenta.
Obtener usuarios	GET	Admin	Obtener todos los usuarios con todos sus campos.	Tras cargar el panel de administración.
Crear usuario	POST	Admin	Crear un usuario nuevo a partir de los datos proporcionados.	Tras clicar el botón 'Crear' en el formulario de crear usuario, en el panel de administración.
Eliminar usuarios	POST	Admin	Eliminar usuarios.	Tras clicar en 'Eliminar' en la tabla de usuarios del panel de administración, habiendo seleccionado al menos un usuario.

Adicionalmente, se cuenta con endpoints adicionales para servir información estática, como imágenes o documentos en PDF. De igual forma, se cuenta con endpoints necesarios para realizar los procesos de autenticación en redes sociales.

Cada servicio cuenta con endpoints específicos que reciben cada tipo de consulta. Estos se encargan de procesar y pre-configurar la solicitud, para luego ponerla en cola para ser ejecutada por el worker de Celery. Esto se debe a que la ejecución de los servicios toma tiempos mayores a un segundo. Los detalles de las tareas pesadas se describen en la sección siguiente.

5.3. Desarrollo de servicios

Luego de hacer todo esto, se cuenta con las interfaces minimalistas conectadas entre sí, y con el sistema de usuarios implementado. También se tienen funcionalidades internas, relativas al manejo de los datos y al panel de administración. Por lo tanto, corresponde

comenzar con el desarrollo de los servicios que debe ofrecer la aplicación. A continuación se explica el desarrollo de los mismos.

5.3.1. Generalidades

Los servicios consisten en consultas que realiza el usuario, las cuales son procesadas por endpoints de la API interna, y luego enviadas al worker de Celery para que éste las encole y ejecute. Cada servicio proporciona resultados distintos, los cuales pueden desplegarse en tiempo real, o exportarse a través de un documento PDF.

Los resultados en tiempo real se organizan en tarjetas, cada una de las cuales representa una métrica o resultado. Estas tarjetas pueden ser marcadas como favoritas, tras lo cual esa misma métrica (para la misma consulta) quedará disponible en el dashboard. El usuario puede agregar y eliminar la cantidad de métricas y consultas favoritas que desee.

Cada servicio cuenta con un modal de preferencias, en donde el usuario puede definir las métricas que desea obtener en cada tipo de consulta. También, puede personalizar características de las métricas, como el tipo de gráfico o el color de estos, y características del servicio en general, como el número máximo de tweets.

Todos los informes en PDF que se soliciten pueden ser descargados o enviados por email, según decida el usuario. Además, los últimos 100 informes quedan disponibles para descarga, en un servicio dedicado en /reports.

5.3.2. Autenticación de usuarios

Debido a que en la gran mayoría de los servicios se requiere autenticación en cuentas, por ejemplo, de redes sociales, estos cuentan con un modal de autenticación. En este modal, se muestran los detalles de las cuentas con autorización, pudiendo agregarlas o eliminarlas según corresponda.

Para el caso de Facebook e Instagram, se permite al usuario acceder a Facebook y autorizar un perfil, junto con las páginas asociadas a este, y las cuentas de Instagram de tipo Business o Creator. Para el caso de Twitter, el usuario puede agregar cuentas siguiendo el flujo de autenticación oficial. En ambos casos, se sigue el protocolo OAuth 1.0. Los flujos de autenticación se observan en las Figuras 5 y 6 respectivamente.

En el caso de Google Analytics 4, para autorizar un sitio web es necesario agregar un correo electrónico generado por la aplicación de Laik en dicha plataforma. Una vez que el usuario haya hecho esto para su sitio web, podrá solicitar validarlo, y de ser exitosa dicha validación, quedará autenticado dicho sitio web, y se podrá acceder a las métricas.

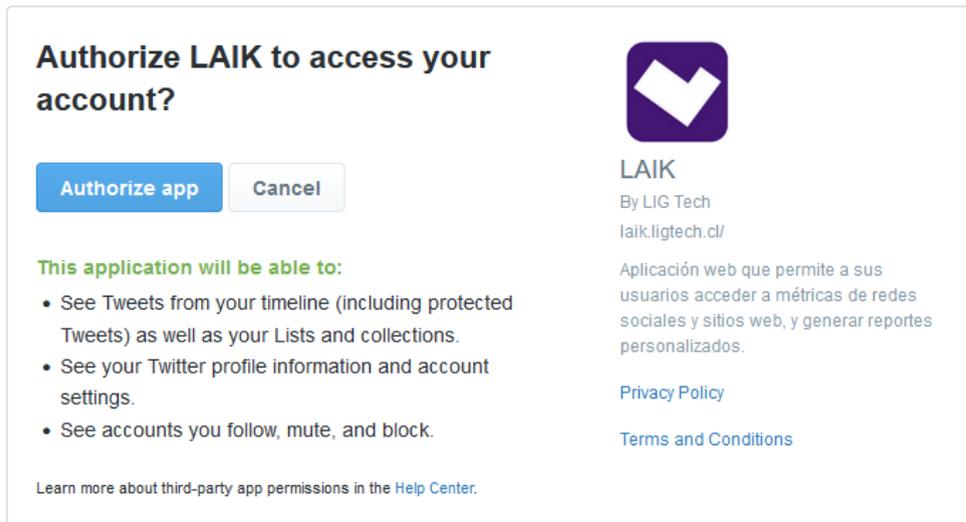


Figura 5: Flujo de autenticación en Twitter.

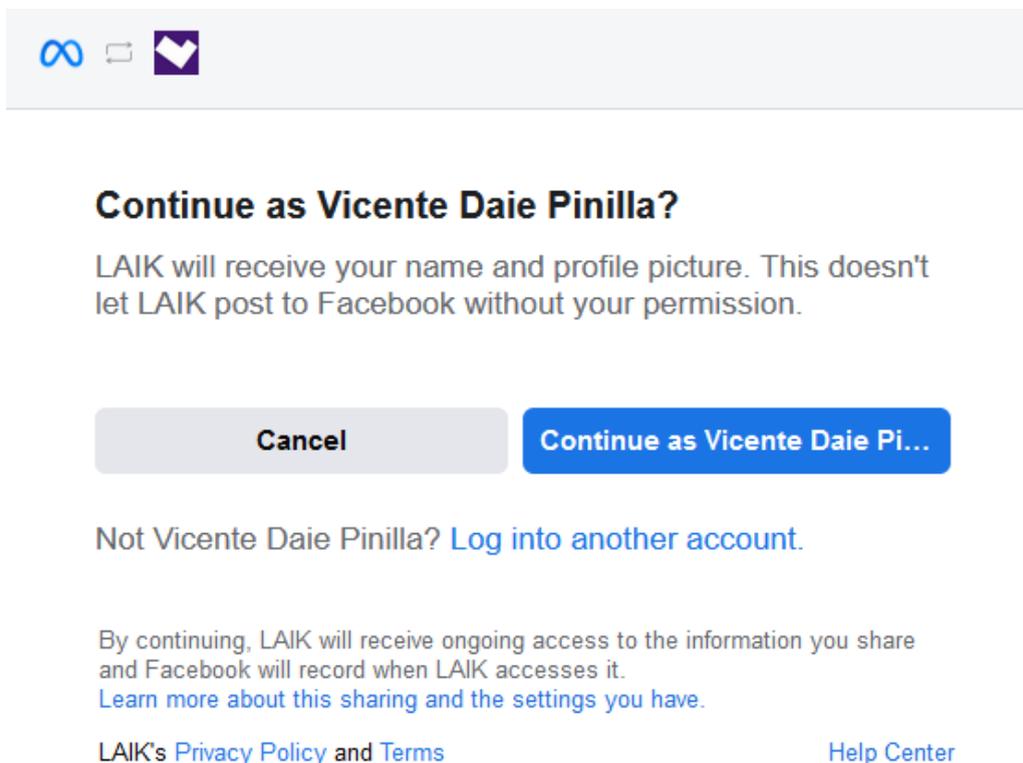


Figura 6: Flujo de autenticación en Facebook e Instagram.

5.3.3. Administración de procesos pesados

Las consultas que se ofrecen en cada servicio consisten en procesos pesados, por lo que deben manejarse de forma tal de evitar tiempos de espera excesivos, distribuyendo los recursos del servidor de manera óptima. La API de Flask se encarga de encolar estas solicitudes al worker de Celery, las cuales se ejecutan una vez que se tenga recursos para

ello, priorizando las solicitudes con mayor tiempo de espera. La API de Flask se asegura de que el usuario no pueda solicitar algún servicio del que ya se tenga una solicitud en cola.

Las tareas pesadas consisten en conectarse a las APIs respectivas de cada red social (o de Google Analytics 4, en el caso de sitios web), y solicitar la información necesaria, proporcionando la autenticación necesaria. Esta información se ordena y se filtra, según la información en los archivos JSON de preferencias de cada usuario, y, una vez que se encuentre lista, se almacena el resultado para que pueda ser solicitado por la API de Flask. En caso de haberse solicitado un informe en PDF, éste es creado y guardado, para ser descargado por el usuario cuando éste lo requiera.

Cada vez que se solicita desplegar los resultados de un servicio en tiempo real, se registra que este está en proceso, y, mientras siga así, se gatillan consultas periódicas cada un segundo. Si una de estas consultas observa que la solicitud está lista, entonces se reciben los resultados y se despliegan al usuario. Los resultados de la última consulta quedan en caché, el cual se muestra al usuario cada vez que se carga un servicio.

5.3.4. Tareas del lado del cliente

Algunas tareas, debido a su simplicidad, naturaleza, o que no requieren acceso directo a información privada que esté en el servidor, se ejecutan directamente a través de funciones de JavaScript. Algunas de éstas son las siguientes:

- Llamadas a la API interna de Flask, a través de axios³³.
- Construcción de gráficos, a través de react-chartjs-2³⁴.
- Construcción de tablas dinámicas, a través de MUI³⁵.
- Validación de formularios.

5.4. Línea gráfica

A este punto, se cuenta con una aplicación funcional, la cual cuenta con interfaces interconectadas, un sistema de usuarios funcional, funcionalidades internas y servicios que se encuentran operativos. Desde el punto de vista funcional, ésta se encontraría completa, sin embargo, para tener valor real para los usuarios, debe contar con interfaces amigables e intuitivas, y la información debe estar organizada de manera ordenada y coherente. Además, ésta debe contar con una “identidad propia”, para lo cual se requiere un logotipo y línea gráfica. En primer lugar, se define el logotipo de la aplicación, quedando el de la Figura 7.

³³ <https://axios-http.com/docs/intro>

³⁴ <https://react-chartjs-2.js.org/>

³⁵ <https://mui.com/>



Figura 7: Logotipo de Laik.

Posteriormente, siguiendo dicha imagen, se procede a definir el resto de las vistas, desde el punto de vista gráfico. Se define que el color morado del logotipo será el color principal de Laik, el cual se manifestará principalmente en el landing, en la barra de navegación, y los botones a lo largo de los servicios. Los estilos son construidos a partir de Bootstrap v5.2³⁶, mediante CSS y Sass³⁷.

La aplicación se diseña para que sea responsiva, esto es, que la información se despliegue de manera ordenada independiente del tamaño de la pantalla. Esto se logra gracias a las funciones de Bootstrap, las cuales permiten definir límites de tamaños de pantalla, y comportamientos dinámicos en función de estos límites.

Para el despliegue de la información proporcionada por los servicios en tiempo real, se define que cada métrica se mostrará en una “tarjeta”, las cuales estarán agrupadas según su tipo. Éstas contarán con un título, un botón para marcarla como favorita, y el contenido, el cual puede ser un número, texto, una imagen, una tabla o un gráfico. A continuación se muestra un ejemplo de tarjeta en la Figura 8; en este caso, corresponde al servicio de análisis de usuario por Twitter.

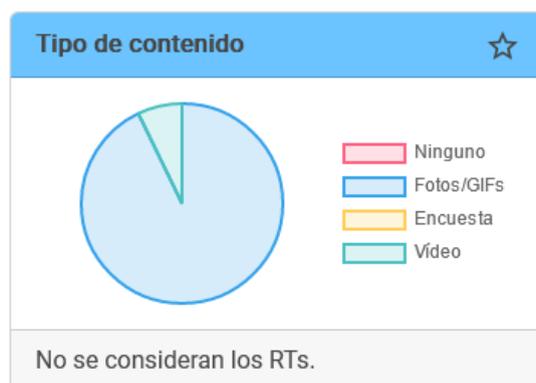


Figura 8: Tarjeta conteniendo una métrica.

5.5. Puesta en producción

Una vez que se tiene un producto mínimo viable (con las funcionalidades y servicios listos, pero no toda la línea gráfica implementada), se procede a poner en producción una primera

³⁶ <https://getbootstrap.com/>

³⁷ <https://sass-lang.com/>

versión de la aplicación, para que pueda ser sujeta al proceso de evaluación mediante pruebas de usuario.

Se cuenta con un servidor de LIG Tech hospedado en Linode³⁸, el cual opera bajo el sistema operativo Ubuntu 22.04 LTS. Se clona el repositorio desde este servidor, y, mediante Docker y docker-compose, se levanta la plataforma. Luego, esta pasa por un proxy reverso de Nginx, la cual la deja expuesta en la URL asociada al servidor: <https://laik.ligtech.cl>. Es necesario montar también la base de datos, puesto que ésta no forma parte de la dockerización.

Una vez que se realiza la primera ronda de pruebas de usuario, se procede a recoger los comentarios obtenidos, y usarlos como retroalimentación durante las últimas semanas del desarrollo. La gran mayoría de los comentarios de la primera ronda se basaron en la falta de una línea gráfica completada, lo que es natural puesto que esta parte del desarrollo quedó para el final. Posteriormente se realiza la segunda ronda de pruebas de usuario, cuyos comentarios se usan principalmente como insumos para los próximos pasos a seguir de este proyecto. El proceso completo de evaluación mediante pruebas de usuario se describe en el capítulo 7.

5.6. Entrega del producto

Finalmente, tras la segunda ronda de pruebas de usuario, teniendo la plataforma implementada y desplegada en el servidor, corresponde hacer la entrega del producto a LIG Tech. Esto se hizo formalmente a través de una reunión durante la semana del lunes 5 de diciembre, posterior a la segunda ronda de pruebas de usuario.

Debido a los comentarios obtenidos por los usuarios de prueba, se concluyó que el cumplimiento de los objetivos planteados es efectivo (lo que se discutirá a cabalidad en los capítulos 7 y 8). Sin embargo, múltiples ideas han surgido durante el desarrollo y las pruebas de usuario, las cuales son compatibles con la extensibilidad de la plataforma. Por lo tanto, se optó por continuar con el desarrollo del producto, incorporando servicios y funcionalidades nuevas que agregarían valor a la plataforma. Estas extensiones también se discuten en el capítulo 8.

³⁸ <https://www.linode.com/>

6. Interfaces del Sistema

En esta sección se presentarán los resultados del desarrollo, recorriendo cada una de las interfaces de la aplicación.

6.1. Landing page del sitio

Al ingresar a la URL <https://laik.ligtech.cl>, el usuario se encontrará con la landing page del sitio (Figura 9). En ésta se describe de manera general el funcionamiento de la aplicación. Se cuenta con enlaces a la página web de LIG Tech, y un botón para acceder al Dashboard (o a la pantalla de inicio de sesión, según corresponda).



ANALIZA LA SITUACIÓN COMPLETA

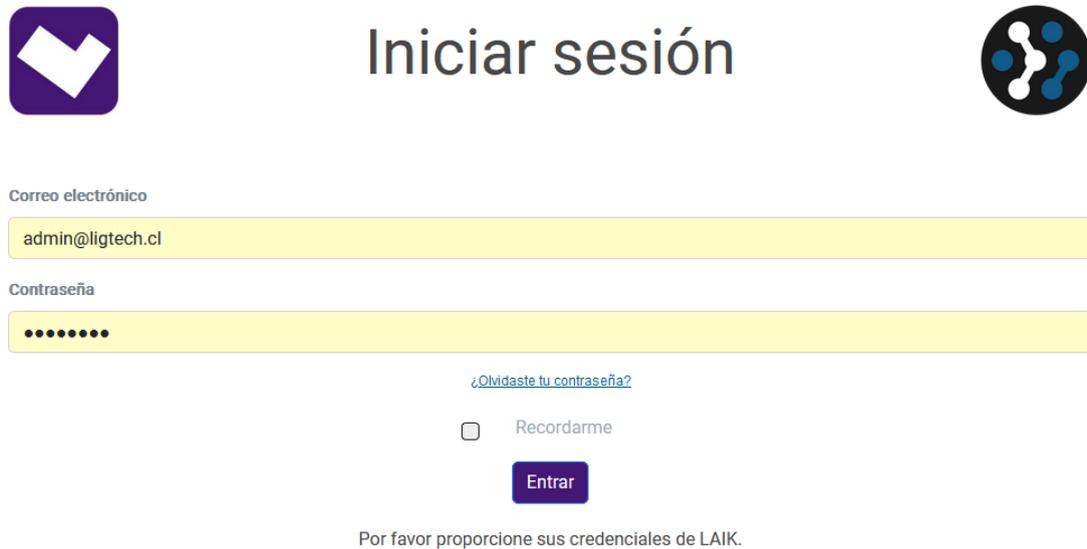
Laik cuenta con variados servicios que permiten obtener información tanto de tu marca como de la competencia

- Cuentas de Twitter**
Analiza cuentas de Twitter en profundidad, junto con estadísticas de sus tweets. Proporciona autenticación y accede a métricas exclusivas.
- Cuentas de Instagram**
Estudia cuentas de Instagram en profundidad. Proporciona autenticación y accede a estadísticas de su audiencia y sus publicaciones.
- Páginas de Facebook**
Recolecta información sobre páginas de Facebook, junto con las de sus publicaciones.
- Google Analytics 4**
Accede a las dimensiones y métricas de tus sitios web.
- Clipping de prensa**
Recoge las menciones de tu marca (o cualquier consulta) en medios de prensa nacionales.
- Clipping en redes sociales**
Realiza búsquedas en redes sociales, filtrando por palabras clave y universos de interés.

Figura 9: Landing page de Laik.

6.2. Autenticación de usuarios

Al intentar acceder a una interfaz restringida de las descritas anteriormente, el usuario es redireccionado a la página de inicio de sesión (Figura 10). En ésta, el usuario puede además recurrir a un modal para recuperar su contraseña, como se observa en la Figura 11.



Correo electrónico

admin@ligtech.cl

Contraseña

••••••••

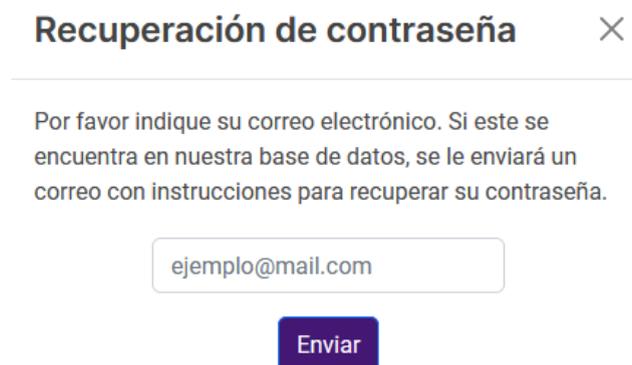
[¿Olvidaste tu contraseña?](#)

Recordarme

Entrar

Por favor proporcione sus credenciales de LAIK.

Figura 10: Formulario de inicio de sesión.



Recuperación de contraseña ×

Por favor indique su correo electrónico. Si este se encuentra en nuestra base de datos, se le enviará un correo con instrucciones para recuperar su contraseña.

ejemplo@mail.com

Enviar

Figura 11: Formulario para solicitar recuperación de contraseña.

6.3. Vista de administración

La vista de administrador de la Figura 12, exclusiva para un único usuario, cuenta con una tabla que describe el resto de los usuarios registrados en la plataforma. Aquí, el administrador puede agregar usuarios (mediante el formulario de la Figura 13), y eliminar o enviar correos electrónicos a los usuarios seleccionados.

Panel de administración

Usuarios

Usuarios activos							
<input type="checkbox"/>	Email ↑	Nombre	Apellido	Fecha de creación	Última conexión	Organización	Posición
<input type="checkbox"/>	vdaiep@gmail.com	Vicente		Fri, 09 Dec 2022 13:39:57 GMT			

Filas por página: 5 ▾ 1-1 of 1 < >

[Agregar un usuario](#)

[Ir al dashboard](#)

Figura 12: Panel de administración.

Crear un usuario



Los campos con (*) son obligatorios.

Email (*)

Contraseña (*)

Confirmar contraseña (*)

Nombre (*)

Apellido

Organización

Cargo

Crear usuario

Figura 13: Formulario para crear un nuevo usuario.

6.4. Interfaces de servicios

Estas interfaces se encuentran conectadas entre sí mediante una barra de navegación al costado izquierdo. En cada una (excepto la de Informes), se pueden desplegar resultados en tiempo real, o solicitar informes en PDF. En caso de solicitar resultados en tiempo real, estos quedarán guardados en caché, y se desplegarán la próxima vez que se visite dicha interfaz.

6.4.1. Dashboard

En el Dashboard, el usuario puede consultar las métricas que ha marcado como favoritas a lo largo de los demás servicios. A continuación, en las Figuras 14 y 15, se muestra el dashboard con los resultados de una consulta en tiempo real. Este ejemplo cuenta con métricas marcadas como favoritas provenientes de Google Analytics 4 (Laik), Instagram (@ligtech_test), Twitter (@uchile) y Facebook (Universidad de Chile Ingeniería).

★ Dashboard

Métricas favoritas

⚙️ Preferencias

👤 Autenticación

🔍 Seguimiento

Aquí encontrará sus métricas favoritas de todos los servicios disponibles en LAIK. En Preferencias podrá gestionar su listado de favoritos. Además, puede agregar (o quitar) métricas, (des)marcando la estrella en la esquina superior derecha de cada una.

Resultados en tiempo real

Generar PDF

Mostrando favoritos, actualizados al 19/12/2022 05:44:24.

📊 Google Analytics 4: Laik



📷 Instagram: @ligtech_test

Cuenta autenticada

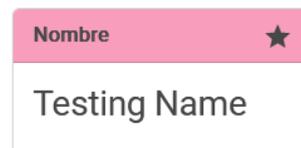
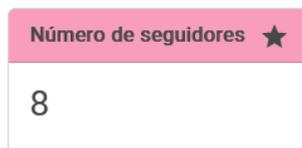
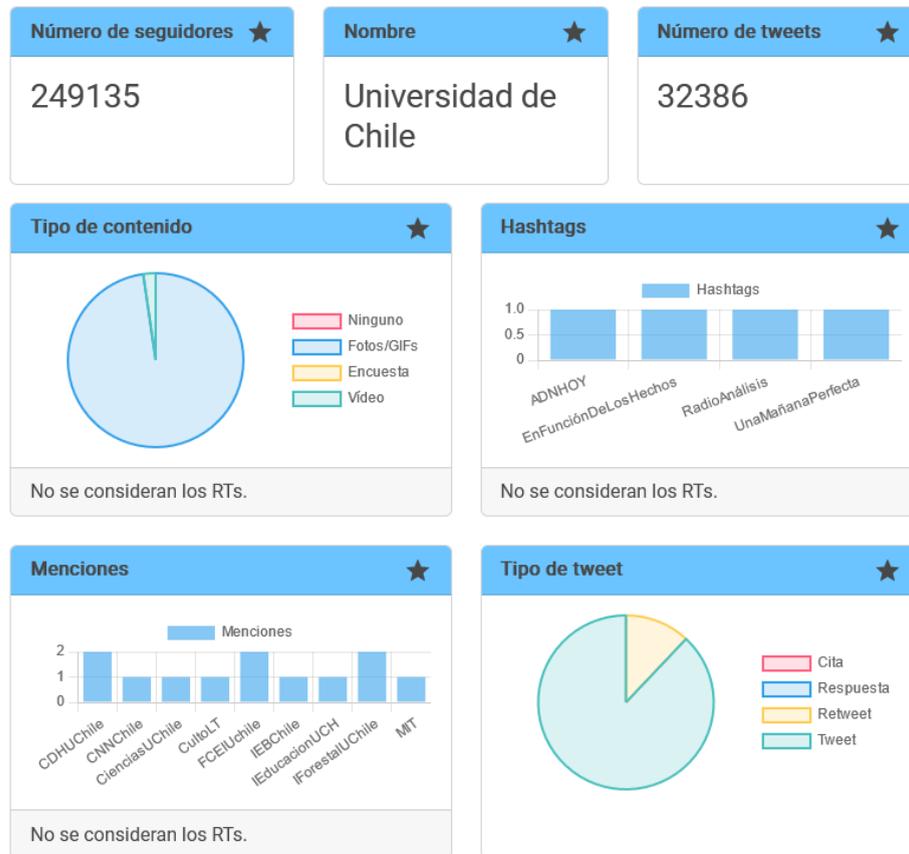


Figura 14: Ejemplo de consulta en el dashboard, parte 1.

Twitter: @uchile
Cuenta sin autenticar



Facebook: Universidad de Chile Ingeniería
Página sin autenticar



Figura 15: Ejemplo de consulta en el dashboard, parte 2.

6.4.2. Twitter

En este servicio, se despliegan las métricas solicitadas de un usuario de Twitter. Éstas se organizan en secciones, mostrando sólo aquellas que el usuario ha solicitado (configurables en el modal de Preferencias). A continuación, en la Figura 16 se observa este servicio, junto con resultados en tiempo real para la consulta @uchile.

Nombre ☆	Descripción ☆	Fecha de creación ☆
Universidad de Chile	Somos la principal y más antigua institución de educación superior del	23/07/2007 16:26:01

Figura 16: Análisis por usuario de Twitter.

6.4.3. Facebook

Para Facebook, se cuenta con un servicio para obtener métricas a partir de una página, estando esta autenticada o no. A continuación se muestra este servicio en la Figura 17, con resultados para la consulta 'Universidad de Chile', página sin autenticar.

Laik
INTELIGENCIA DE MEDIOS

Dashboard
Informes
Facebook
Análisis por perfil
Análisis por página
Instagram
Twitter
Sitios web

Administrador
Cuenta Salir

LIG Tech 2022 © – Versión 0.4.1
Política de Privacidad | Términos y Condiciones

Facebook
Análisis por página

Preferencias
Autenticación

Tipo de página
Sin autenticar

Página
Universidad de Chile

Resultados en tiempo real
Generar PDF

Mostrando resultados de caché para Universidad de Chile, con fecha 16/12/2022 12:31:17.

Facebook: Universidad de Chile
Página autenticada

Portada

Nombre ☆	Descripción ☆	Sitio web ☆
Universidad de Chile	Con 178 años de tradición, somos la principal y más antigua institución de educación superior del Estado, nacional y pública. http://www.uchile.cl	http://www.uchile.cl

Figura 17: Análisis por página de Facebook.

6.4.4. Instagram

Este servicio permite extraer métricas de cuentas de Instagram de tipo Business o Creator. En la Figura 18 se muestra un ejemplo de los resultados de este servicio desplegados en tiempo real, para la consulta @ligtech_test, cuenta autenticada.

Laik
INTELIGENCIA DE MEDIOS

Dashboard
Informes
Facebook
Instagram
Twitter
Sitios web

Administrador
Cuenta Salir

LIG Tech 2022 © – Versión 0.4.1
Política de Privacidad | Términos y Condiciones

Instagram
Análisis por usuario

Preferencias
Autenticación

Tipo de usuario
Autenticado

Nombre de usuario
@ligtech_test

Resultados en tiempo real
Generar PDF

Mostrando resultados para @ligtech_test actualizados al 18/12/2022 02:33:21.

Instagram: @ligtech_test
Cuenta autenticada

Portada

Nombre ☆	Biografía ☆	Sitio web ☆
Testing Name	This is the biography	https://play1.vdaiep.cl/

Figura 18: Análisis por usuario de Instagram.

6.4.5. Google Analytics 4

Este servicio permite extraer métricas personalizables de páginas autenticadas que usen el servicio de Google Analytics 4, como se muestra en la Figura 19.

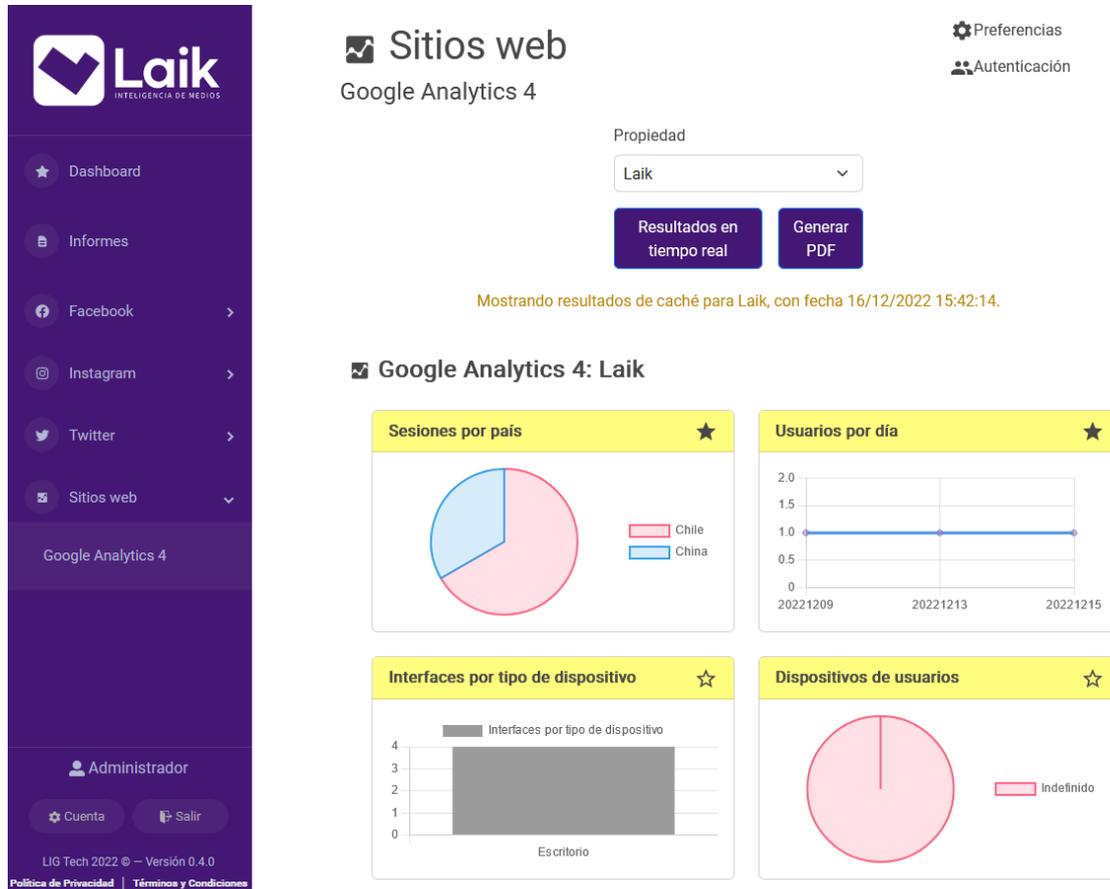


Figura 19: Análisis de sitio web en Google Analytics 4.

6.4.7. Acceso a los informes

En esta interfaz, el usuario puede acceder a su historial de informes en PDF, pudiendo descargar los 100 más recientes. Además, se muestran los informes pendientes y programados, en caso de haber, como se observa en la Figura 20.

The screenshot shows the 'Informes' section of the Laik application. On the left is a dark purple sidebar with navigation options: Dashboard, Informes, Facebook, Instagram, Twitter, Sitios web, and a user profile section for 'Administrador' with 'Cuenta' and 'Salir' buttons. The main content area has a header with 'Informes' and 'Preferencias'. Below the header is a text block explaining that the section shows a history of PDF reports requested from LAIK, including those being generated or scheduled. A 'Actualizar tablas' button is present. The 'Informes listos' section contains a table with one row for 'Favoritos' and a 'DESCARGAR' button. The 'Informes pendientes' section contains a table with one row for a Twitter account '@vdaiep'.

Informes Preferencias

En esta sección encontrará un historial de todos los informes en PDF que ha solicitado a LAIK, en cualquiera de sus servicios. También se indican aquellos informes que se encuentran generándose en este momento, y aquellos programados mediante seguimientos. Los 100 documentos más recientes estarán disponibles para ser descargados durante 90 días, tras lo cual serán borrados de nuestro servidor.

Actualizar tablas

Informes listos

Servicio	Consulta	Detalle	Fecha	Tamaño [kB]	Descarga
★ Favoritos	—	—	04/12/2022 22:19:26	8.12	DESCARGAR

Filas por página: 10 1-1 de 1

Informes pendientes

Servicio	Consulta	Detalle	Fecha
Twitter	@vdaiep	Cuenta autenticada	16/12/2022 19:50:19

Filas por página: 10 1-1 de 1

LIG Tech 2022 © — Versión 0.4.0
Política de Privacidad | Términos y Condiciones

Figura 20: Interfaz en que se detalla el estado de los informes.

6.4.8. Modales de preferencias

Cada servicio puede ser personalizado por el usuario, lo cual es posible a través del respectivo modal de preferencias. A continuación se muestran dos ejemplos, uno para el servicio de análisis de usuario de Twitter (Figura 21), y otro para el de análisis de usuario de Instagram (Figura 22).

Preferencias ×

General Métricas Informes

Consumo de tweets

Cada usuario de Laik cuenta con un máximo de 100.000 tweets mensuales para consultar.

Tweets extraídos: 2481
Tweets programados: 0

2% 0%

Generalidades

Límite de tweets:

50

Incluir RTs en muestra: Métricas no públicas: Tweets referenciados:

Actualizar preferencias

Figura 21: Modal de preferencias de Twitter.

Preferencias



Métricas

Informes

Portada

- | | | | |
|-----------|---|-------------------------|---|
| Nombre | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes | Número de seguidores | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes |
| Biografía | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes | Número de seguidos | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes |
| Sitio web | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes | Número de publicaciones | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes |

Inspección

- | | | | |
|-------------------|--|-------------------|--|
| Impresiones | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes
Color <input type="text" value=""/> | Nuevos seguidores | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes
Color <input type="text" value=""/> |
| Alcance | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes
Color <input type="text" value=""/> | Clicks en URL | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes
Color <input type="text" value=""/> |
| Visitas al perfil | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes
Color <input type="text" value=""/> | Clicks en Email | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes
Color <input type="text" value=""/> |

Audiencia

- | | | | |
|--------|---|--------|---|
| País | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes | Género | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes |
| Ciudad | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes | Edad | <input checked="" type="checkbox"/> Mostrar en tiempo real
<input checked="" type="checkbox"/> Mostrar en informes |

Actualizar preferencias

Figura 22: Modal de preferencias de Instagram.

6.4.9. Modales de autenticación

Debido a que algunos servicios pueden (o deben) contar con autenticación, se tiene un modal para realizar los flujos de autenticación descritos anteriormente. Desde cada servicio se puede acceder al modal correspondiente, mientras que desde el dashboard se tiene acceso a todos. Las Figuras 23, 24 y 25 muestran los modales para autenticación en Facebook/Instagram, Twitter y Google Analytics 4 respectivamente.

Autenticación



Facebook | Instagram

Twitter

Google Analytics 4

Mis cuentas

No tienes usuarios autenticados.

Gestionar permisos

Para acceder a información de cuentas de Facebook y/o Instagram, es necesario proporcionar autorización a LAIK. Haciendo click a continuación, podrá autenticar un usuario de Facebook, y autorizar las páginas de Facebook y cuentas de Instagram asociadas a este. También, podrá modificar los permisos ya otorgados.

Facebook

Puede autenticar todos los usuarios que desee, cada uno con sus respectivas páginas y cuentas asociadas. Puede quitar los permisos en cualquier momento.

Figura 23: Modal de autenticación para Facebook e Instagram.

Autenticación



Facebook | Instagram

Twitter

Google Analytics 4

Mis cuentas

Cuenta	Eliminar
@vdaiep	X

Agregar cuentas

Para acceder a información privada de cuentas de Twitter, es necesario proporcionar autorización a LAIK. Haciendo click en el siguiente botón, podrá autenticar una cuenta de Twitter.

Twitter

Puede autenticar todas las cuentas que desee, y quitar los permisos en cualquier momento.

Figura 24: Modal de autenticación para Twitter.

Propiedades autenticadas

ID	Nombre	Eliminar
336984814	Laik	✕

Agregar propiedades

Para acceder a métricas de Google Analytics 4, es necesario proporcionar autorización a LAIK. Para ello, por favor, [agregue el siguiente usuario a la propiedad que desea autorizar](#) (incluya permisos de lectura y análisis):

```
starting-account-xl5puh9woi6h@laik-1667333241639.iam.gserviceaccount.com
```

Una vez realizado, rellene el formulario y haga click en el siguiente botón para verificar.

ID de propiedad

Nombre

Puede autenticar todas las propiedades que desee, y quitar los permisos en cualquier momento.

Figura 25: Modal de autenticación para Google Analytics 4.

6.5. Configuración de cuenta de usuario

Esta interfaz, accesible como servicio, permite realizar cambios a la cuenta del usuario. En particular, puede editar los datos de su perfil, y cambiar la contraseña, como se observa en las Figuras 26 y 27 respectivamente.

Configuración de cuenta

Actualizar datos Cambiar contraseña

Correo electrónico
admin@ligtech.cl

Nombre
Administrador

Apellido
Laik

Organización
LIG Tech

Cargo
CTO

Actualizar datos

Figura 26: Formulario para actualizar datos.

Configuración de cuenta

Actualizar datos Cambiar contraseña

Contraseña
●●●●●●

Contraseña nueva
Ingrese una contraseña nueva

Confirmar contraseña nueva
Repita su contraseña nueva

Cambiar contraseña

Figura 27: Formulario para cambiar contraseña.

6.6. Documentos legales

Finalmente, se cuenta con una interfaz (Figura 28) para la descarga de cada documento: la Política de Privacidad, y los Términos y Condiciones. Estos documentos no requieren autenticación de usuario para poder ser accedidos.

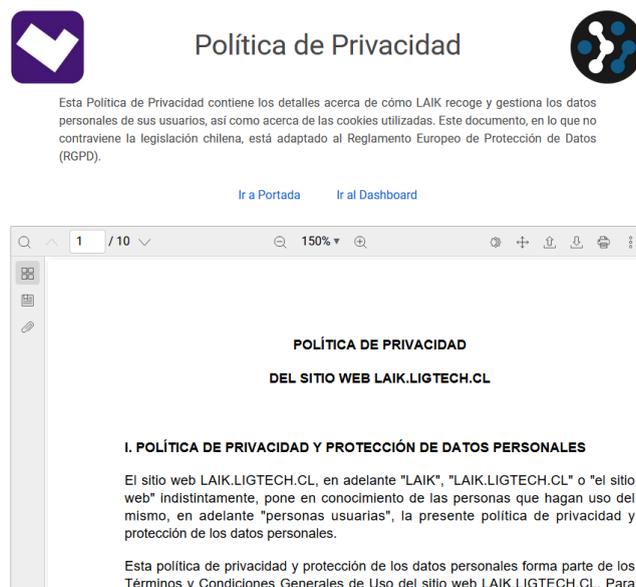


Figura 28: Política de Privacidad del sitio. La interfaz para los Términos y Condiciones es análoga.

7. Evaluación de la Solución

A continuación, se describe el proceso de pruebas de usuario con que se validó el cumplimiento de los objetivos planteados.

7.1. Proceso de evaluación realizado

Para validar los resultados del desarrollo, se realizaron dos rondas de pruebas de usuario, en que se solicitó retroalimentación sobre el cumplimiento de los objetivos, mientras se realizaban las correcciones finales al código. Se esperaba la participación de al menos dos usuarios en cada ronda, los cuales debían cumplir con el perfil de los usuarios finales del producto, esto es, profesionales en áreas de marketing o prensa de sus respectivas empresas.

Una vez terminados la base y los servicios, se cuenta con una primera versión funcional de la aplicación. Por lo tanto, lo siguiente es pasarla a un servidor para que se realicen las pruebas de usuario. Se cuenta con un servidor privado en Linode en que se puede montar la aplicación, para que esta quede accesible desde la red.

Una vez se encuentre funcionando desde la dirección IP del servidor, se invita a los usuarios de prueba a usar la aplicación, probar sus servicios y funcionalidades, y proveer feedback de su experiencia. Estas pruebas de usuario se realizan en dos rondas, con 2 semanas de separación entre sí.

Durante este tiempo, se sigue trabajando en el desarrollo, resolviendo errores menores y detalles de diseño. Además, se refuerza la robustez del código, y se incorporan posibles mejoras que puedan provenir de los comentarios de los usuarios de prueba. Esta fase de detalles y robustez se extiende hasta la última semana, durante la cual se hace la entrega del software a LIG Tech.

El proceso consiste en una visita guiada al sitio, en donde el usuario inicia sesión y realiza consultas que le sean de interés. Se recogen los comentarios que realizan a medida que van probando las distintas funcionalidades, y una serie de comentarios finales.

7.2. Resultados obtenidos

En las tablas 2 y 3 se presenta un resumen de los resultados obtenidos en cada una de las rondas de pruebas de usuario. Los usuarios fueron reclutados directamente por LIG Tech.

Tabla 2: Resultados de la primera ronda de pruebas de usuario, realizada durante la semana del lunes 21 de noviembre de 2022.

Usuario	Perfil	Resumen de comentarios
1	Administradora de tienda online que cuenta con página web y cuenta de Instagram.	<ul style="list-style-type: none"> • La plataforma es útil. • Falta estilo y orden en el dashboard. • El servicio de Google Analytics 4, en particular la forma en que definen las métricas, es "genial". • Puede resultar confusa la forma en que se organizan los favoritos.
2	Encargado de marketing de empresa con sitio web y cuentas de Twitter e Instagram.	<ul style="list-style-type: none"> • Podrían mostrarse métricas calculadas a partir de otras métricas. • La plataforma tiene potencial, pero hay mucho por refinar aún. • Falta reforzar la apariencia de la aplicación, no se siente homogéneo.

Tras la primera ronda de pruebas de usuario, se observa que la principal prioridad debiese ser reforzar la estética de la plataforma. Esto resulta natural pues, a dicha fecha, aún no se aplicaba completamente la línea gráfica definida, puesto que faltaban funcionalidades por desarrollar. Se le dio un poco más de urgencia, logrando homogeneizar gran parte de los estilos de la plataforma, junto con integrar el logotipo y el landing. De esta forma, los próximos comentarios se concentrarán en las funcionalidades.

Tabla 3: Resultados de la segunda ronda de pruebas de usuario, realizada durante la semana del lunes 5 de diciembre de 2022.

Usuario	Perfil	Resumen de comentarios
1	Administradora de tienda online que cuenta con página web y cuenta de Instagram.	<ul style="list-style-type: none"> • Notable mejora visual, con el logo y el esquema morado, la aplicación cobra vida. • Ver solo lo que le interesa en el dashboard es “lo mejor”. • Podría haber más de un dashboard, con listas de métricas distintas. • Interesada en usar la plataforma para aumentar sus ventas.
3	Publicista, con acceso a múltiples cuentas corporativas en redes sociales.	<ul style="list-style-type: none"> • La aplicación puede ser muy útil en las manos correctas. • Se podrían incluir servicios de análisis de cuentas públicas de medios de prensa en redes sociales, para ver qué dice la prensa de su marca en redes sociales. • A medida que se agreguen más servicios, más aumenta el valor de este producto. • “Ilusionado” de lo que podría llegar a hacerse con esto.
4	Periodista, experto en redes sociales	<ul style="list-style-type: none"> • El producto es una “buena idea”. • Manifiesta que a los usuarios les gustará tener un dashboard con solo las métricas de interés. • Algunas métricas deben resaltarse más que otras, según su importancia. Podrían mejorarse las métricas seleccionadas por default para ahorrar pasos al usuario.

De la segunda ronda de pruebas de usuario, se puede concluir que, de manera general, se cumplen los objetivos planteados para esta memoria. Se observa un resultado positivo, en que los usuarios se manifiestan interesados en el producto, además de haber cumplido con el objetivo que a cada uno se le planteó. También, se reciben bastantes ideas que podrían ser incorporadas a la plataforma como servicios adicionales o mejoras a los ya existentes. Estas posibles extensiones se discuten en la sección siguiente.

8. Conclusiones y Trabajo a Futuro

En este trabajo de memoria, se aborda el problema de la dispersión de los datos provenientes de redes sociales y sitios web, el cual complica el trabajo de personas encargadas del área de marketing de las marcas comerciales, puesto que supone grandes curvas de aprendizaje y la inversión de una gran cantidad de tiempo.

8.1. Cumplimiento de los objetivos

La solución propuesta, una aplicación web llamada Laik, soluciona el problema proporcionando una plataforma que permite al usuario acceder a métricas de distintas fuentes de manera sencilla y rápida. Además, permite organizar estas métricas a través del dashboard, pudiendo tener acceso a aquellas que son de interés con un solo click. Los resultados pueden desplegarse tanto en tiempo real como mediante informes en PDF.

Se cuenta con servicios para tres redes sociales: Twitter (mediante análisis de usuarios), Instagram (mediante análisis de usuarios) y Facebook (mediante análisis de páginas). Además, se cuenta con un servicio para Google Analytics 4, que permite extraer información relativa a sitios web.

El cumplimiento de los objetivos se validó a través de dos rondas de pruebas de usuarios, en que potenciales usuarios de la plataforma realizaron visitas guiadas a la aplicación, proporcionando comentarios y retroalimentación. Estos comentarios (en particular los de la segunda ronda), confirman el cumplimiento de los objetivos, pues los usuarios se manifiestan “interesados en usar la plataforma”, “ilusionados” y que “en las manos correctas podría ser muy útil”. Además, todos cumplieron con el objetivo para el que se les planteó ser usuarios de prueba, dado que cumplían con el perfil de los potenciales usuarios finales.

8.2. Futuro del proyecto

La aplicación se desarrolló de forma tal que sea extensible, permitiendo incorporar nuevos servicios a futuro. Una vez que se hizo entrega del producto, se decidió continuar trabajando en éste para incorporar nuevos servicios y funciones, algunos de los cuales fueron propuestos por los usuarios durante las pruebas. A continuación, se describen algunas futuras extensiones:

- *Clipping de prensa*: Servicio que permita realizar consultas sobre los medios de prensa digitales del país, permitiendo ver las menciones de una marca.
- *Análisis de perfil de Facebook*: Al igual que las páginas de Facebook, servicio que extraiga métricas sobre los perfiles.
- *Búsqueda en Twitter*: Servicio que permita realizar búsquedas en Twitter, filtrando por listas de usuarios emisores (universos). Además de los predefinidos (entre ellos uno relacionado a medios de prensa), el usuario podrá definir sus propios universos.

- *Seguimientos programados*: Modal que permita programar consultas a cada servicio, de manera puntual o periódica.
- *Integración con APIs premium de Twitter*: Twitter cuenta con APIs exclusivas para acceder a ciertos tipos de información, como el historial completo de un usuario. Estas APIs podrían integrarse para realizar servicios adicionales relativos a Twitter, pero requieren adicionalmente de una contratación.

8.3. Reflexión

El desarrollo de esta memoria contempló conocimientos y técnicas adquiridas a lo largo de todos los años de la carrera de ingeniería civil en computación. Estos van desde cursos elementales, como Bases de Datos, hasta aquellos más relacionados al desarrollo de software como Desarrollo de Aplicaciones Web o Proyecto de Software, que permitieron contar con todas las herramientas necesarias para desarrollar un proyecto de este tipo.

Junto con los cursos propiamente tal, el enfoque educativo proporcionado por la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile, y, en particular, del Departamento de Ciencias de la Computación, hacen la diferencia a la hora de abordar trabajos de este estilo. La capacidad de análisis y los métodos de resolución de problemas, que lentamente se van adquiriendo a medida que se avanza en la carrera, son también en gran medida responsables de que este proyecto haya podido llevarse a cabo.

Finalmente, es necesario destacar que un proyecto de esta magnitud también involucra una gran cantidad de aprendizaje. Éste se manifiesta principalmente en:

- la organización e integración de proyectos que involucren múltiples tecnologías.
- el uso avanzado de versionamiento con git para mantener al mismo tiempo una instancia de desarrollo y otra de producción.
- el uso del framework Flask como solución para aplicaciones de tipo API.
- el uso del framework React.js, como solución para interfaces de usuario dinámicas.
- el uso de Celery como solución para la administración de procesos pesados.

Todas estas lecciones tendrán, sin duda, su efecto positivo en proyectos futuros, y en la vida profesional en general.

9. Bibliografía

- Clark, D.J. (2014). Evaluating information seeking and use in the changing virtual world: the emerging role of Google Analytics. *Learned Publishing*, 27(3), 163-239.
- Custy, J. (2007). Knowledge centred support & ITIL.
- Smith, M. (2012). Big data privacy issues in public social media. *6th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*.