

Trabajo fin de grado

Plataforma web para el análisis de mensajería instantánea



Juan Bautista Menchero Amigo

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C/ Francisco Tomás y Valiente nº 11

**UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**Plataforma web para el análisis de mensajería
instantánea**

**Autor: Juan Bautista Menchero Amigo
Tutor: Esther Guerra Sánchez**

junio 2022

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

DERECHOS RESERVADOS

© 24 de Mayo de 2022 por UNIVERSIDAD AUTÓNOMA DE MADRID

Francisco Tomás y Valiente, n.º 1

Madrid, 28049

Spain

Juan Bautista Menchero Amigo

Plataforma web para el análisis de mensajería instantánea

Juan Bautista Menchero Amigo

C\ Francisco Tomás y Valiente N.º 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

*A mi madre.
Este título es tan tuyo como mío.*

*El regalo más grande que le puedes dar a los demás
es el ejemplo de tu propia vida.*

Bertolt Brecht

AGRADECIMIENTOS

Principalmente quiero agradecer a mi madre, *María Jesús Amigo Rodríguez*, la única persona que se ha esforzado en que yo llegara hasta aquí más que yo mismo. Te lo debo todo, gracias por transmitirme los valores del amor, la bondad, el trabajo, la fuerza, la flexibilidad,... Eres la persona que más admiro.

A mi padre, *Juan Bautista Menchero García*, por haberme inculcado la importancia del conocimiento y la constancia. Por despertar mi interés por todas las cosas como fuente inagotable de felicidad.

A mi hermano, *Hugo Jiménez Amigo*, por mostrarme la vida que quería seguir y no para la que estaba predestinado, y haberme limpiado y allanado el camino que era desconocido para ambos. Por animarme y guiarme a ser la mejor versión de mí que puedo ser.

A *Aitana Rickert Llàcer*, por haberme acompañado en cada línea de código y cada página de la carrera con tu luz infinita. Por haberme cuidado y dado todo lo que necesitaba cuando yo no era capaz de darlo de vuelta, pero sobre todo por enseñarme a ser mejor persona, aunque no estés para verlo.

A *Álvaro Martínez Morales* (yo sí que me sé tus apellidos), por ser lo mejor que me ha dado esta etapa, no podía soñar nada que supere haberme llevado alguien más en mi familia.

RESUMEN

En este trabajo se ha diseñado e implementado tanto el software como la infraestructura necesarias para ofrecer una aplicación web accesible por el público para la generación de reportes de interés sobre su uso de aplicaciones de mensajería instantánea.

El objetivo es contribuir a la democratización de los datos, así como concienciar a los usuarios de la información que se puede extraer de ellos.

En el contenido de esta memoria se exploran un conjunto de buenas prácticas de la arquitectura de software llevadas a un entorno real en producción, con enfoque en la alta disponibilidad, la escalabilidad, la integración continua y la inmutabilidad, aprovechando la velocidad de desarrollo y abstracciones que permiten las plataformas en la nube actuales.

PALABRAS CLAVE

Computación en la nube, Análisis de datos, Ingesta de datos, Alta disponibilidad, Escalabilidad, Integración continua, Inmutabilidad, AWS, Amazon Web Services, Terraform, Redshift, Vue

ABSTRACT

This work details the design and implementation of both the software and infrastructure that are necessary to provide public access to a web application for the generation of reports of interest on their use of instant messaging applications.

It aims to contribute to the democratization of the access to data, as well as to raise awareness among users of the information that can be extracted from their private data.

The content of this report explores a set of best practices of software architecture taken into a real production environment, focusing on high availability, scalability, continuous integration and immutability, taking advantage of the speed of development and abstractions allowed by current cloud platforms.

KEYWORDS

Cloud computing, Data analysis, Data ingestion, High availability, Scalability, Continuous integration, Immutability, AWS, Amazon Web Services, Terraform, Redshift, Vue

ÍNDICE

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	1
1.3	Organización de la memoria	2
2	Estado del arte	3
2.1	Derecho de acceso	3
2.2	Información destilada de libre acceso	4
2.2.1	Spotify Wrapped	4
2.2.2	Tinder Insights	5
2.2.3	Chat Visualizer	6
2.3	Computación en la nube	7
2.4	Bases de datos	8
2.4.1	Relacionales vs No relacionales	8
2.4.2	Columnares vs Orientadas a filas	8
2.5	Frameworks de frontend	8
3	Análisis de requisitos	9
3.1	Requisitos no funcionales	9
3.2	Reportes individuales	9
3.3	Reportes comparativos	10
4	Diseno	11
4.1	Experiencia de usuario	11
4.2	Infraestructura	12
4.3	Planificacion	14
5	Implementacion	15
6	Pruebas	17
7	Conclusiones y trabajo futuro	19
7.1	Conclusiones	19
7.2	Trabajo futuro	19
	Bibliografía	21

LISTAS

Lista de algoritmos

Lista de códigos

Lista de cuadros

Lista de ecuaciones

Lista de figuras

2.1	Spotify Wrapped	4
2.2	Tinder Insights	5
2.3	Chat Visualizer	6
2.4	Cuota de mercado	7
2.5	Frontend frameworks	8

Lista de tablas

Lista de cuadros

INTRODUCCIÓN

1.1. Motivación

En la actualidad, como usuarios de un gran número de servicios informáticos gratuitos, generamos una cantidad inmensa de información sobre nosotros mismos todos los días a la que solo tienen acceso las plataformas que nos prestan los servicios que consumimos [1].

La mayoría de la población, en muchos casos hasta ajena de la definición de metainformación, ni siquiera puede plantearse ser conscientes de las conclusiones que son capaces de sacar sobre nosotros mismos a partir de nuestros datos, lo que conlleva un descuido cada vez más asumido sobre nuestra privacidad.

Este trabajo propone las bases para una plataforma de acceso libre y gratuito, que podrá expandirse gracias a la comunidad opensource, a lo que típicamente resultaría un proyecto interno de análisis de datos para cualquier empresa que tuviera acceso a los mismos para atajar esa falta de conocimiento.

1.2. Objetivos

La intención principal de este trabajo es concienciar al público de la información que los actuales propietarios de sus datos pueden obtener sobre su intimidad, para, en última instancia, promover una gestión más consciente de su información.

En segundo plano también se pretende entretener a los visitantes del sitio con los infográficos generados, siendo el boca a boca el único método viable actualmente para publicitar la plataforma y tener un mayor alcance para el mensaje.

En tercer lugar, este proyecto pretende tener recorrido más allá del ámbito académico, sentando las bases para una comunidad de desarrollo de código abierto, con un enfoque en la separación de conceptos [2] para añadir nuevas analíticas de manera independiente para favorecer la contribución de código por parte de la comunidad.

Por último y a título personal, el desarrollo realizado ha servido como oportunidad para extraer conclusiones creativas a partir de lenguaje natural y metadatos, así como explorar tecnologías de cloud provistas por AWS para la ingesta y procesamiento de los datos y aplicar buenas prácticas de la arquitectura de software en un entorno de producción real.

1.3. Organización de la memoria

En este trabajo presento una aplicación web donde los usuarios pueden analizar sus copias de seguridad de aplicaciones de mensajería instantánea (como Telegram o What's App) y visualizar varios reportes con información que se puede destilar en base a esos históricos.

A lo largo de los siguientes capítulos de la memoria detallo:

- 1.— **Estado del arte:** Se profundiza en el marco legal que permite la realización de este proyecto, el estudio de otros proyectos similares así como un pequeño resumen del contexto y alternativas de las principales tecnologías elegidas (AWS, Redshift y Vue).
- 2.— **Análisis de requisitos:** Se recopilan todos los requisitos tanto funcionales como no funcionales identificados, en los que se basan las decisiones tecnológicas detalladas más adelante.
- 3.— **Diseño:** Se realiza una propuesta de experiencia de usuario que satisfaga todos los requisitos funcionales indicados previamente, la infraestructura que de soporte a los requisitos no funcionales y una descripción de la planificación de la ejecución a seguir.
- 4.— **Implementación:** Profundiza en la solución de código adoptada tanto para la infraestructura como para la plataforma.
- 5.— **Pruebas:** Se resumen las pruebas realizadas para garantizar la calidad del proyecto.
- 6.— **Conclusiones y trabajo futuro:** Recopila las conclusiones alcanzadas tras el desarrollo de este trabajo y propone una serie de futuros pasos a seguir tras su defensa.

ESTADO DEL ARTE

En este capítulo se profundiza tanto en el marco legal que permite la realización de este proyecto, así como en el estudio de otros proyectos similares y un pequeño resumen del contexto y alternativas de las principales tecnologías elegidas (AWS, Redshift y Vue).

2.1. Derecho de acceso

El 14 de abril de 2016 se aprobó en el Parlamento Europeo el Reglamento General de Protección de Datos [3], entrando en vigor el 24 de Mayo de 2016 y concediendo un periodo de aplicación de dos años hasta el 24 de Mayo de 2018. A partir del 25 de Mayo de 2018 todas las empresas, organizaciones, organismos o instituciones dentro del marco europeo comenzaron a tener la obligación, bajo multa por incumplimiento de hasta 20 millones de euros, de ofrecer a los usuarios de una manera fácilmente accesible y legible una copia de sus datos almacenados.

Gracias a esta ley podemos solicitar ante cualquier plataforma que almacene o trate nuestros datos una copia de seguridad de toda nuestra información que tengan disponible. Desde Google, Facebook, Twitter, Spotify, Tinder o, en el caso de la información de la que es sujeto el análisis de este trabajo, a aplicaciones de mensajería instantánea como What's App o Telegram.

Esto nos permite tener acceso a una fuente de datos para el proyecto de manera sencilla y fácilmente ingerible por el sistema propuesto, cuando disponer de fuentes reales de datos es usualmente la parte más complicada a la hora de aterrizar una idea conceptual en un entorno de producción.

No obstante esta ley, aunque obliga a las empresas a compartir la propiedad de los datos crudos con el individuo que los genere, no impone bajo ninguna cláusula que se compartan las conclusiones destiladas que generan a partir de esa información cruda.

2.2. Información destilada de libre acceso

2.2.1. Spotify Wrapped

Desde 2016 Spotify lanza anualmente una campaña de marketing que permite a sus usuarios visualizar una compilación de sus datos de uso, comparándolos con el resto de la comunidad que utiliza la aplicación. Desde resúmenes meramente estadísticos, así como análisis de emociones o intereses.

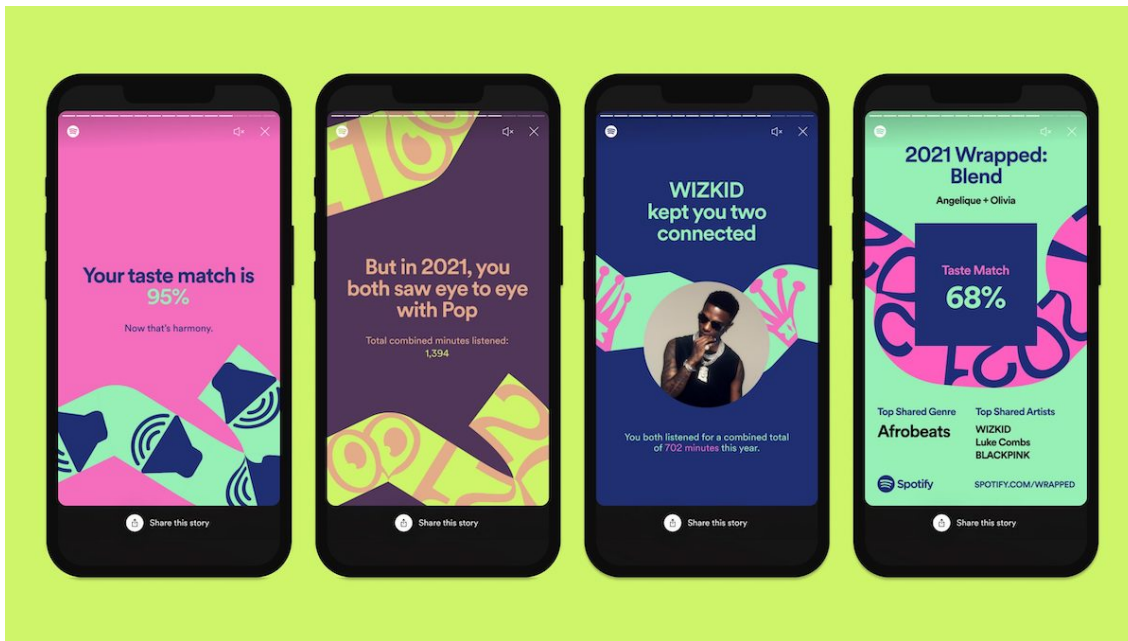


Figura 2.1: Ejemplos de pantallas de reporte de la iniciativa Spotify Wrapped

Esta campaña ha tenido un gran impacto social y cultural, llegando a formar parte de la cultura pop de las nuevas generaciones, y generando en el inconsciente colectivo inquietud e interés por los datos a una población regularmente ajena y desinteresada de su información en la era digital.

2.2.2. Tinder Insights

Creada en 2019 por Dora Szucs (Software Engineer) y Krisztina Szucs (UX Designer) de manera independiente, da acceso a información estadística sobre el uso de la aplicación Tinder, como la cantidad de mensajes enviados y recibidos, tiempo medio de chat, ...

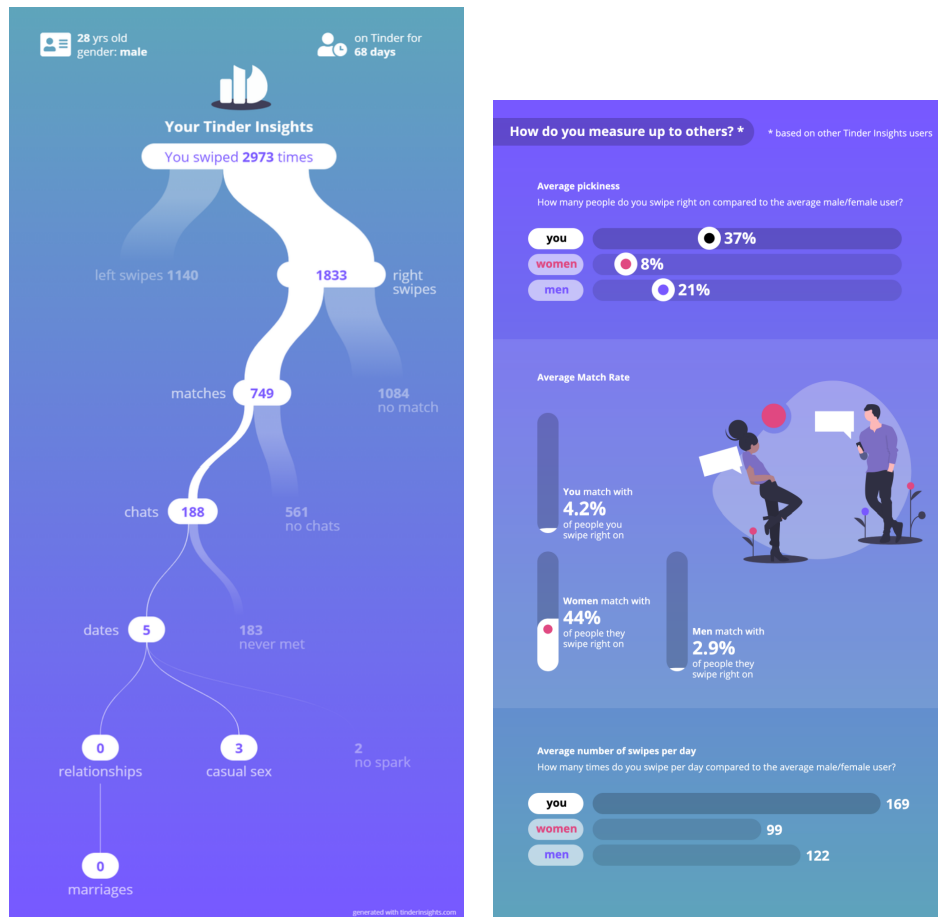


Figura 2.2: Ejemplos de pantallas de reporte de Tinder Insights

Siguiendo el mismo planteamiento que nuestro proyecto, requieren que el usuario solicite paralelamente desde un flujo externo a la aplicación su copia de seguridad, sirviendo como ejemplo práctico de que el interés por la información sobre uno mismo puede ser suficiente para que un gran número de población comparta sus datos privados de uso con una aplicación de terceros para analizarla (incluyendo sus conversaciones a través de la app).

2.2.3. Chat Visualizer

Una herramienta que permite analizar la actividad de un chat de What's App de manera estadística (72K chats analizados en el momento de la redacción de esta memoria).

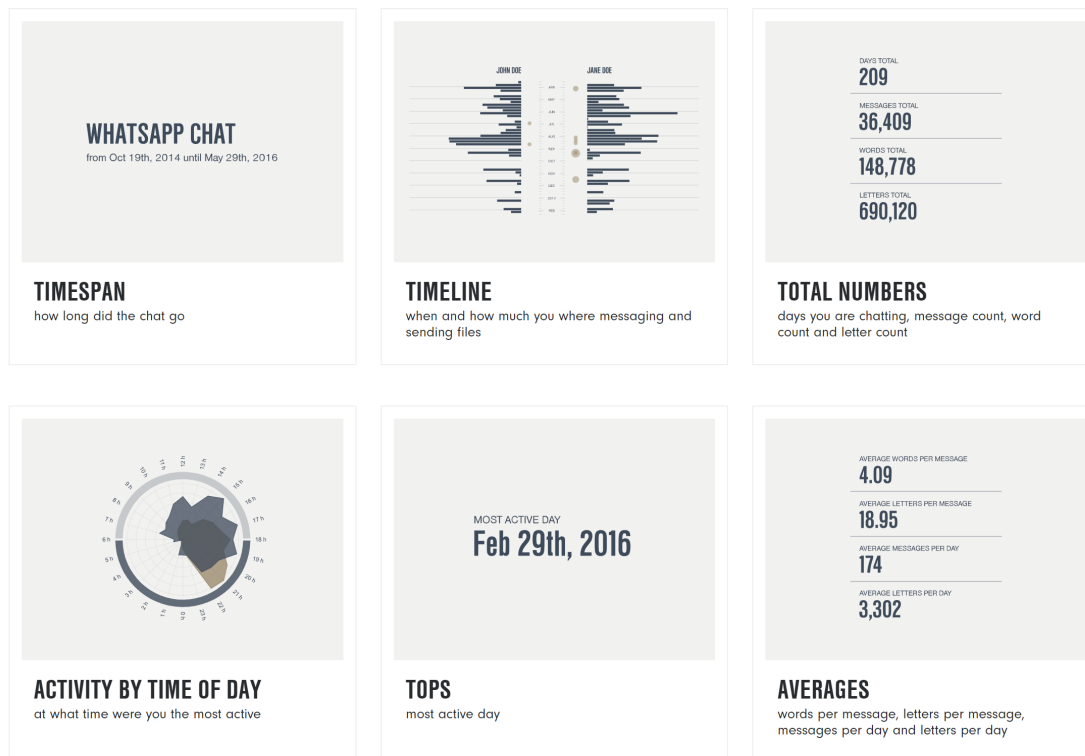


Figura 2.3: Ejemplos de pantallas de reporte de Chat Visualizer

Contras que este trabajo pretende subsanar:

- Solo permite analizar una conversación
- Solo analiza información de What's App
- Procesa los datos en el lado del servidor, sin garantizar la privacidad del usuario
- Solo ofrece un análisis meramente estadístico de la actividad del chat

2.3. Computación en la nube

La computación en la nube es la oferta de servicios de computación tales como servidores, bases de datos, redes privadas, software, herramientas de análisis, ... a través de internet, sin la necesidad de disponer de hardware físico en propiedad y beneficiándose de la economía de escala que aporta el compartir recursos de forma flexible entre un gran número de usuarios.

Actualmente hay numerosos proveedores de cloud, a continuación se esbozan las principales ventajas de aquellos con más cuota de mercado:

- **Amazon Web Services:** Mayor presencia en el mercado, gran número de servicios, disponibilidad global con escalabilidad y replica entre regiones, 99.99 % de disponibilidad garantizada.
- **Microsoft Azure:** Destacan sus servicios de aprendizaje automático e inteligencia artificial.
- **Google Cloud:** Como creadores de Kubernetes, es la solución mejor integrada para la utilización de contenedores de Docker en organizaciones con arquitectura de microservicios.
- **Digital Ocean:** Enfocados en dar servicio a desarrolladores, y no tanto a grandes organizaciones con complejos requisitos de infraestructura.

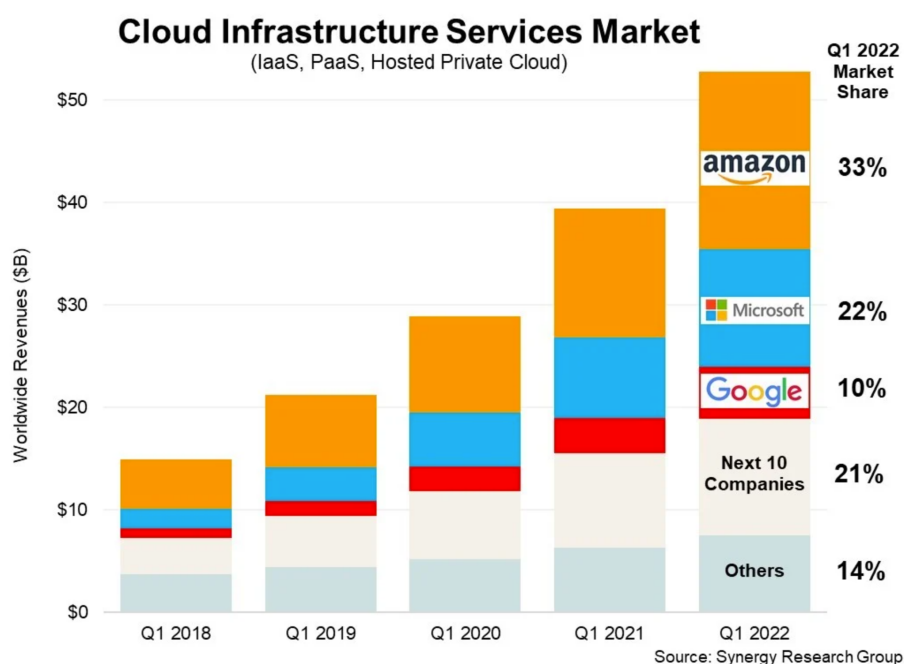


Figura 2.4: Histórico de la progresión de cuota de mercado de proveedores de cloud

2.4. Bases de datos

2.4.1. Relacionales vs No relacionales

- **Relacionales:** Almacenan la información de manera estructurada, con una forma constante y una declaración explícita de las relaciones entre los diversos elementos (MySQL, Oracle, SQL Server, PostgreSQL, Redshift, ...).
- **No relacionales:** Almacenan la información como documentos independientes, que no tienen por qué mantener la misma forma ni tener relaciones explícitas (MongoDB, Redis, Elasticsearch, Cassandra, ...).

2.4.2. Columnares vs Orientadas a filas

- **Columnares:** Almacenan de manera contigua cada columna o atributo, para acelerar el análisis estadístico de toda una tabla (Redshift, BigQuery, ...).
- **Orientadas a filas:** Almacenan en memoria contigua la información de cada entidad, permitiendo un acceso rápido a todos los datos de una misma entidad (SQLServer, PostgreSQL, ...).

2.5. Frameworks de frontend

El espectro del frontend es muy cambiante, en apenas dos o tres años las tecnologías más presentes pueden pasar a ser consideradas obsoletas, pero gracias a iniciativas como StateOfJS, una encuesta global sobre el ecosistema front, se puede consultar de manera muy visual el estado actual de esta rama.

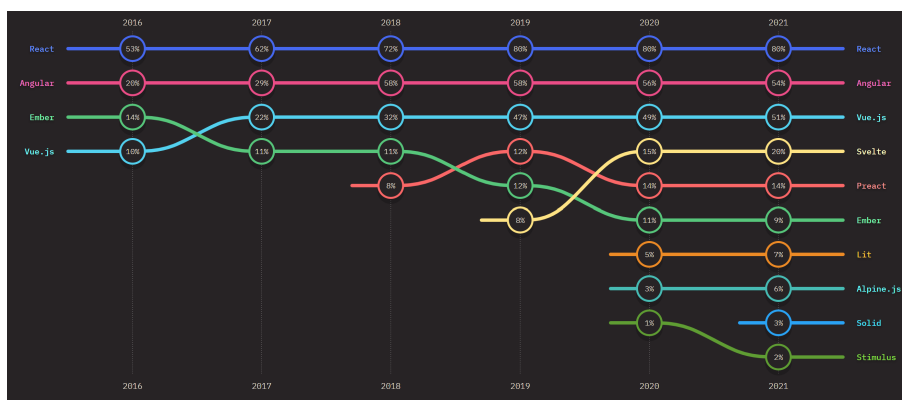


Figura 2.5: Progresión de uso de frameworks de frontend. Fuente: StateOfJS.com

Como podemos observar en el gráfico de uso, Vue.JS parece ser el único framework con un crecimiento constante y consistente en los últimos cinco años, pudiendo significar un declive del resto de competidores en los próximos años en favor de este framework, destacable por su suave curva de aprendizaje y su sintaxis de HTML enriquecida, integrando bucles, condicionales y bindings de datos en las propias etiquetas nativas de HTML.

ANÁLISIS DE REQUISITOS

En esta sección se recopilan todos los requisitos tanto funcionales como no funcionales identificados, en los que se basan las decisiones tecnológicas detalladas más adelante.

3.1. Requisitos no funcionales

- **RNF-01 [Alta disponibilidad]:** Garantizar un tiempo de disponibilidad suficientemente alto como para que no haya usuarios que decidan no usar la aplicación por no estar disponible.
- **RNF-02 [Disponibilidad geográfica]:** Disponer de presencia del sitio web en las caches globales para permitir su acceso regular desde cualquier parte del mundo.
- **RNF-03 [Escalabilidad horizontal]:** Permitir escalar de manera virtualmente infinita tanto en datos como en procesamiento, para cubrir un posible crecimiento de popularidad del sitio.
- **RNF-04 [Intuitividad de uso]:** Ofrecer una interfaz gráfica intuitiva para permitir el acceso a cualquier tipo de usuario.
- **RNF-05 [Tiempos de respuesta]:** Que los tiempos de generación de reportes sean suficientemente rápidos como para no resultar perceptibles por el usuario.
- **RNF-06 [Interés personal]:** Mostrar reportes interesantes para que los usuarios tengan una motivación a la hora de utilizar la plataforma.
- **RNF-07 [Interés social]:** Utilizar infográficos llamativos para favorecer que se compartan por redes sociales y publicitar la herramienta.
- **RNF-08 [Mensaje de concienciación]:** Aportar información que conciencie del compromiso a la intimidad que supone otorgar acceso a los históricos de chat.
- **RNF-09 [Privacidad de los datos]:** Evitar que la información sensible del usuario salga de su ordenador para garantizar la privacidad de sus datos.
- **RNF-10 [Comunidad de desarrollo]:** Simplificar la colaboración con el código fuente para la ampliación de los reportes mostrados.

3.2. Reportes individuales

- **RF-01 [Presentación del proyecto]:** Presentar el proyecto de la forma más rápida posible.
- **RF-02 [Guía de exportación]:** Guiar al usuario durante el proceso de exportación de sus datos.
- **RF-03 [Procesar datos de Telegram]:** Permitir subir el histórico de mensajes de Telegram.

- **RF-04 [Procesar datos de What's App]:** Permitir subir el histórico de mensajes de What's App.
- **RF-05 [Emojis más usados]:** Mostrar una lista con los emojis más usados.
- **RF-06 [Uso diario]:** Desglosar la actividad diaria de la aplicación.
- **RF-07 [Horas de sueño o trabajo]:** Mostrar durante qué horas se puede deducir que el usuario duerme o trabaja.
- **RF-08 [Intereses]:** Resumir las temáticas principales de las conversaciones del usuario.
- **RF-09 [Interacciones]:** Mostrar las personas que han aparecido o desaparecido de la vida del usuario a partir de su frecuencia de mensajes.
- **RF-10 [Estado anímico]:** Simular una progresión anímica basándose en los emojis más utilizados según el periodo.
- **RF-11 [Contactos favoritos]:** Mostrar a qué contactos responde más rápido el usuario.
- **RF-12 [Contactos más sociales]:** Visualizar un ranking con las personas en las que el usuario comparta más grupos.

3.3. Reportes comparativos

- **RF-13 [Compartir estadísticas]:** Permitir seleccionar al usuario si desea compartir sus estadísticas sin comprometer los datos originales para poder generar comparativas.
- **RF-14 [Comparativa de longitud de mensajes]:** Comparar la longitud de mensaje media del usuario con el resto de la población.
- **RF-15 [Comparativa de uso]:** Comparar la cantidad de mensajes enviados por el usuario con el resto de la población.

DISENYO

En esta seccion se presenta el resumen del proceso de disenyo de la aplicacion, desde la definicion de la experiencia de usuario a partir de los requisitos antes expuestos, a los detalles de implementacion, pasando por la arquitectura e infraestructura necesarias para darles soporte.

Aunque el proceso real de disenyo ha sido realizado de manera iterativa, se presenta organizado por categorias para una mayor facilidad de consulta.

4.1. Experiencia de usuario

La plataforma pretende tener el menor numero de pantallas y elementos posibles para facilitar su uso, intentando reducir el numero de clicks necesarios para consultar los reportes.

Segun uno de los principios de disenyo mas citados, "3-Click rule" [4] ninguna pieza de informacion debe estar a mas de 3 clicks de distancia para evitar perder al usuario en el flujo de la aplicacion. Aunque dicho principio ha sido desmitificado por algunos estudios [5], en nuestro caso en particular hay ya una gran barrera de usabilidad al no disponer de ninguna forma inmediata de acceder a los datos y requerir una serie de pasos intermedios por el usuario fuera de la plataforma para obtener el historico de datos. Por ello se ha limitado a las cuatro siguientes secciones.

1. Presentacion

[Screenshot presentacion]

En esta seccion el objetivo principal era comunicar de la manera mas concisa posible el valor que aporta la aplicacion. La eleccion de palabras intenta motivar el interes apelando a la intimidad comprometida de los usuarios. Acompañada de un corto video de ejemplo donde pueden previsualizar un ejemplo del reporte generado por la aplicacion, para mantener el texto lo mas reducido posible.

2. Pantalla de subida

[Screenshot pantalla de subida]

En la pantalla de subida se presentan las instrucciones para generar la copia de seguridad de manera tanto escrita como visual para acompañar lo máximo posible al usuario durante el proceso que debe realizar inevitablemente fuera de la plataforma.

También se avisa antes de que el usuario comience con la copia de seguridad de que ninguno de sus datos saldrá de su ordenador para que la privacidad no sea un impedimento de uso, y que pueden acceder al código fuente disponible en GitHub. Esto ha sido motivado tanto por intentar comenzar una comunidad de desarrollo entorno a la plataforma, como en un ejercicio de transparencia para transmitir más confianza respecto a la forma de procesar sus datos privados.

3. Reportes individuales

- * Emojis wall

[Screenshot emojis wall]

- * Hourly report

[Screenshot hourly report]

4. Reporte comparativo

[Screenshot comparativa]

Durante todo el diseño visual se han utilizado principios de diseño de White Space Is Not Your Enemy [6].

4.2. Infraestructura

[Diagrama arquitectura global]

Para dar soporte a la plataforma web y su necesidad de almacenamiento de datos, así como todos los requisitos no funcionales anteriormente detallados, se han tomado la siguiente serie de decisiones a la hora de plantear la infraestructura.

Almacenamiento del código: GitHub

De todas las opciones alternativas de almacenamiento de código [TODO: Listar alternativas], se ha optado por GitHub al ser gratuito, el estándar en la comunidad open source, y ofrecer servicios de CI/CD como GitHub Actions o herramientas para la gestión de claves como GitHub Secrets.

Proveedor de cloud: AWS

Los principales proveedores de computación en la nube, como Google o Microsoft, ofrecen servicios similares, no obstante he optado por Amazon por tener una mayor cuota del mercado y seguir en crecimiento, así como por un interés personal en perseguir la certificación de Solutions Architect

de AWS, siendo la certificación de nube más valorada a la hora de la redacción de este documento [TODO: Citar fuente].

Por su transparencia a la hora de ofrecer alta disponibilidad y escalabilidad gracias a servicios como S3 (utilizado tanto para el hosting de la aplicación, como por Redshift como sistema de almacenamiento de sus datos), Amazon Web Services es la alternativa elegida para dar soporte a este proyecto.

Infraestructura como código: Terraform

El almacenar la definición de la infraestructura como código permite tanto [7]. Se opta por Terraform por su fácil integración con GitHub Actions y AWS.

[TODO: Como funciona terraform] Terraform funciona comparando el nuevo estado de infraestructura definido contra el anterior, y ejecutando una serie de pasos mediante las APIs a las que tiene acceso para cambiar en el entorno todo aquello que sea necesario para dejarlo como en la nueva definición del estado.

En este caso en particular, al utilizar GitHub Actions para ejecutar esos cambios, se requiere de lo que se denomina un ".estado remoto". Sin un estado remoto, Terraform genera un fichero local que no sería capaz de salvar en GitHub, por ello requiere configurar a mano un almacenamiento para el estado de manera remota. Se considera una práctica estándar en la integración entre Terraform y AWS utilizar un bucket de S3 para ello.

Hosting: S3

Para el hosting del cliente web se utiliza S3 por dar soporte a disponer el mismo contenido replicado en diferentes áreas geográficas tanto por seguridad como acceso y permitir la redirección de dominios a contenido web estático almacenado.

[TODO: Que es S3]

Base de datos: Redshift

Basándonos en las siguientes categorizaciones y teniendo en cuenta tanto la elección de proveedor de cloud anterior, como que la naturaleza de los datos a almacenar es un resumen de estructura fija, anonimizado y ya procesado del usuario, y que el uso que se le va a dar es realizar un resumen de cada uno de los datos de todos los usuarios en conjunto, normalmente promediados, la opción que más se adecúa a las necesidades del proyecto sería Redshift, por ser relacional, columnar y proveer de escalado y replicado automáticos.

Al funcionar encima de S3 garantiza la misma disponibilidad, escalabilidad y fiabilidad que S3, y al ser una base de datos columnar permite realizar análisis sobre conjuntos de atributos (el caso principal de uso de la aplicación) con un menor tiempo de respuesta.

También ofrece una API para ejecutar sentencias de SQL a través del SDK de AWS, evitando así

la necesidad de una capa intermedia para el backend durante las primeras iteraciones del proyecto.

De todos los servidores de Redshift disponibles, selecciono DC2 por estar incluido gratuitamente en el tier básico de AWS [8].

[TODO: Definir SDK, AWS, S3, ...]

Control de acceso: IAM

El servicio de gestión de usuarios y acceso principal de Amazon Web Services. Se ha utilizado para generar los accesos a los servicios de AWS para GitHub Actions y Terraform.

[Diagrama arquitectura AWS]

Framework de frontend: Vue

4.3. Planificación

Sprints

- 1.— Frontend scaffolding with CD/CI and Infrastructure as a code with AWS
- 2.— Upload steps and easy reports entirely in frontend
- 3.— Save RAW data in S3 Warehouse
- 4.— Save reports in DB

IMPLEMENTACION

PRUEBAS

CONCLUSIONES Y TRABAJO FUTURO

7.1. Conclusiones

7.2. Trabajo futuro

- 1.– Mejorar politicas de acceso IAM
- 2.– Migrar claves de acceso a GitHub secrets
- 3.– Contratar a un UX Designer para replantear el aspecto grafico de la plataforma
- 4.– Comprar un dominio y liberar el acceso a la aplicacion

BIBLIOGRAFÍA

- [1] B. Vuleta, "How much data is created every day?," *SeedsScientific.com*, 2021. (Visitar).
- [2] N. Natesan, "How to implement design pattern," *CastsOfSoftware.com*, 2019. (Visitar).
- [3] DOUE, "Reglamento general de protección de datos," *Diario Oficial de la Unión Europea*, 2016. (Visitar).
- [4] B. Vuleta, "Three-click rule," *SeedsScientific.com*, 2021. (Visitar).
- [5] B. Vuleta, "Three-click rule myth," *SeedsScientific.com*, 2021. (Visitar).
- [6] K. Golombisky and R. Hagen, *White Space Is Not Your Enemy*. USA: A. K. Peters, Ltd., 3rd ed., 2016.
- [7] M. Artac, T. Borovssak, E. Di Nitto, M. Guerriero, and D. A. Tamburri, "Devops: Introducing infrastructure-as-code," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pp. 497–498, 2017.
- [8] B. Vuleta, "Redshift pricing," *SeedsScientific.com*, 2021. (Visitar).



Universidad Autónoma
de Madrid