Universidad Autónoma de Madrid

Escuela Politécnica Superior

21|22

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

Campus Internacional excelencia UAM CSIC+

www.uam.es

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informatica

TRABAJO FIN DE GRADO

Plataforma web para el análisis de mensajería instantánea

Autor: Juan Bautista Menchero Amigo
Tutor: Esther Guerra Sánchez

junio 2022

Juan Bautista Menchero Amigo
Plataforma web para el análisis de mensajería instantánea

Juan Bautista Menchero Amigo
C\ Francisco Tomás y Valiente Nº 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

Listas

1

Introduccion

1.1. Motivacion

Actualmente 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, sin ni siquiera ser conscientes de las conclusiones que son capaces de sacar sobre nuestros datos.

1.2. Objetivos

La intencion es concienciar (o por lo menos entretener) a la gente con la información que se puede obtener de ellos, y tener la oportunidad de extraer conclusiones creativas a partir de lenguaje natural y metadatos, así como explorar tecnologías de cloud provistas por AWS para la ingesta y procesado de los datos.

1.3. Organizacion de la memoria

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

2

Estado del arte

2.1. Derecho de acceso

El 14 de abril de 2016 se aprobo en el Parlamento Europeo el Reglamento General de Proteccion de Datos, entrando en vigor el 24 de Mayo de 2016 y concediendo un periodo de aplicacion de dos anyos hasta el 24 de Mayo. A partir del 25 de Mayo de 2018 todas las empresas, organizaciones, organismos o instituciones dentro del marco europeo comenzaron a tener la obligacion, bajo multa por incumplimiento de hasta 20 millones de euros, de ofrecer a los usuarios de una manera facilmente 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 informacion que tienen disponible, desde Google, Facebook, Twitter, Spotify, Tinder, o, en el caso de la informacion a analizar en este trabajo, a aplicaciones de mensajeria instantanea como What's App o Telegram. Permitiendonos tener acceso a una fuente de datos de manera sencilla y facilmente ingerible por el sistema propuesto.

2.2. Informacion destilada de libre acceso

Spotify Wrapped:

Desde 2016 Spotify lanza anualmente una campana de marketing que permite a sus usuarios visualizar una compilacion de sus datos de uso, comprandolos con el resto de la comunidad que utiliza la aplicacion. Desde resumenes meramente estadisticos, asi como analisis de emociones o intereses.

Esta camapana ha tenido un gran impacto social y cultural, llegando a formar parte de la cultura pop de las nuevas generaciones, y generando de una forma inconsciente inquietud e interes por los datos a la poblacion general, regularmente ajena y desinteresada de sus datos en la era digital.

Tinder Insights:

Creada en 2019 por Dora Szucs (Software Engineer) y Krisztina Szucs (UX Designer), da acceso

a informacion estadistica sobre el uso de la aplicacion Tinder, como la cantidad de mensajes enviados y recibidos, tiempo medio de chat, ...

Ejemplo practico de que el interes por la informacion sobre uno mismo puede ser suficiente para que un gran numero de poblacion comparta sus datos de uso con una aplicacion de terceros para analizarla.

https://tinderinsights.com/contact https://who.is/whois/tinderinsights.com

Chat Visualizer:

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

Contras que este trabajo pretende subsanar:

- Solo permite analizar una conversacion
- Solo analiza informacion de What's App
- Procesa los datos en el lado del servidor, no garantizando la privacidad de esa informacion
- Solo ofrece un analisis estadistico de actividad

2.3. Computacion en la nube

2.4. Frameworks frontend

2.5. Bases de datos

Relacionales vs No Relacionales

Relacionales: Almacenan la informacion de manera estructurada, con una forma constante y una declaracion explicita de las relaciones entre los diversos elementos (MySQL, Oracle, SQL Server, PostgreSQL, ...) No Relacionales: Almacenan la informacion como documentos independientes, que no tienen por que mantener la misma forma ni tener relaciones explicitas (MongoDB, Redis, Elasticsearch, Cassandra, ...)

Columnares vs Orientadas a filas

Orientadas a filas: Almacenan en memoria contigua la informacion de cada entidad, permitiendo un acceso rapido a todos los datos de una misma entidad (SQLServer, PostgreSQL, ...) Columnares: Almacenan de manera contigua cada columna o atributo, para acelerar el analisis estadistico de toda una tabla (Redshift, BigQuery, ...)

3

Analisis de requisitos

Requisitos no funcionales de la aplicacion:

- Alta disponibilidad
- Disponibilidad geografica
- Crecimiento horizontal

3.1. Reportes individuales

- Horas a las que sueles responder - Contactos favoritos a los que respondes mas rapido - Tendencias de tu estado de humor segun los emoticonos que usas mas frecuentemente - Personas que han aparecido o desaparecido de tu vida - Tus temas e intereses favoritos - Horas de suenyo o trabajo. - Personas con las que estas en mas grupos

1.– Ofrecer una interfaz grafica intuitiva para permitir el acceso a cualquier tipo de usuario

2.– Tiempos de respuesta rapidos para evitar que el tiempo de procesamiento sea un impedimento a la hora de utilizar la aplicacion

3.– Mostrar reportes interesantes para que los usuarios tengan una motivacion a la hora de utilizar la plataforma

4.– Aportar informacion que conciencie del compromiso a la intimidad que supone otorgar acceso a los historicos de chat

5.– Utilizar infograficos llamativos para favorecer que se compartan por redes sociales y publicitar la herramienta

6.– Evitar que la informacion del usuario salga de su ordenador para garantizar la privacidad total de los datos

3.2. Reportes comparativos

- Top emojis - Tiempo de uso - Longitud del mensaje medio

1.– Almacenar solo los minimos datos necesarios para los reportes comparativos, sin informacion privada y completamente anonimizados

4

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" [**?**] 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 [**?**], 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. Acompanyada 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 acompanyar lo maximo posible al usuario durante el proceso que debe realizar inevitablemente fuera de la plataforma.

Tambien se avisa antes de que el usuario comience con la copia de seguridad de que ninguno de sus datos saldra de su ordenador para que la privacidad no sea un impedimento de uso, y que pueden acceder al codigo 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 mas 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 disenyo visual se han utilizado principios de disenyo de White Space Is Not Your Enemy [**?**].

4.2. Infraestructura

[Diagrama arquitectura global]

Para dar soporte a la plataforma web y su necesidad de almacenamiento de datos, asi 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 codigo: GitHub

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

Proveedor de cloud: AWS

Los principales proveedores de computacion 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, asi como por un interes personal en perseguir la certificacion de Solutions Architect

de AWS, siendo la certificacion de nube mas valorada a la hora de la redaccion de este documento [TODO: Citar fuente].

Infraestructura como codigo: Terraform

El almacenar la definicion de la infraestructura como codigo permite tanto [**?**]. Se opta por Terraform por su facil integracion 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 definicion 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 seria capaz de salvar en GitHub, por ello requiere configurar a mano un almacenamiento para el estado de manera remota. Se considera una practica estandar en la integracion 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 areas geograficas tanto por seguridad como acceso y permitir la redireccion de dominios a contenido web estatico almacenado.

[TODO: Que es S3]

Base de datos: Redshift

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

Tambien ofrece una API para ejecutar sentencias de SQL a traves del SDK de AWS, evitando asi 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 basico de AWS [**?**].

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

Control de acceso: IAM

El servicio de gestion 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]

4.3. Planificacion

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

5

Implementacion

6

Pruebas

7

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] “Three-click rule,” (Visitar).

[2] “Three-click rule myth,” (Visitar).

[3] K. Golombisky and R. Hagen, *White Space Is Not Your Enemy: A Beginner’s Guide to Communicating Visually Through Graphic, Web amp; Multimedia Design*. USA: A. K. Peters, Ltd., 3rd ed., 2016.

[4] 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.

[5] “Redshift pricing,” (Visitar).


UÁM
Universidad Autónoma
de Madrid