Autor	**Mauro Crosignani**
Fecha	**23/05/2025**

Ejercicio Técnico - Mini Tweeter

Descripción

Se busca diseñar una plataforma de gestión de mensajes, emulando tweeter. A continuación se detallan los componentes, junto con las tecnologías elegidas.

Aclaraciones

El ejercicio no llegó a completarse al 100%. Se deja la estructura de las aplicaciones pero faltan tests.

Así mismo hay algunos componentes repetidos que hubiese sido preferible abstraer a una lib.

Tecnologías

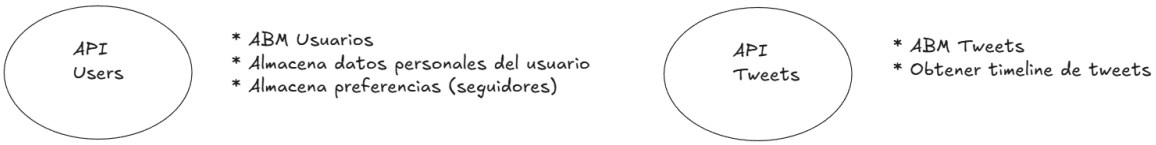
Para los microservicios se elige GO, ya que es un lenguaje de fácil aprendizaje para el desarrollo, simple, con GC automático, y el cual es liviano a la hora de dockerizar y desplegar en la nube.
Las aplicaciones están divididas por scope (Read, Write, Worker) para garantizar escalabilidad.
Para este ejercicio se decide crear 2 microservicios: API Users y API Tweets.

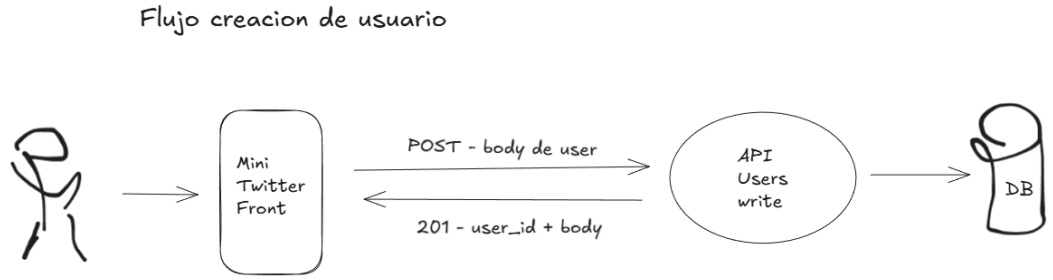
Las bases de datos seleccionadas son relacional (PostgresDB). Por una cuestión de simplicidad en el desarrollo para ambas aplicaciones se usa la misma tecnología de base de datos, pero en un caso real se hubiese optado por una base de datos no relacional para mantener los tweets, y quizás un KVS para las listas de seguidores.

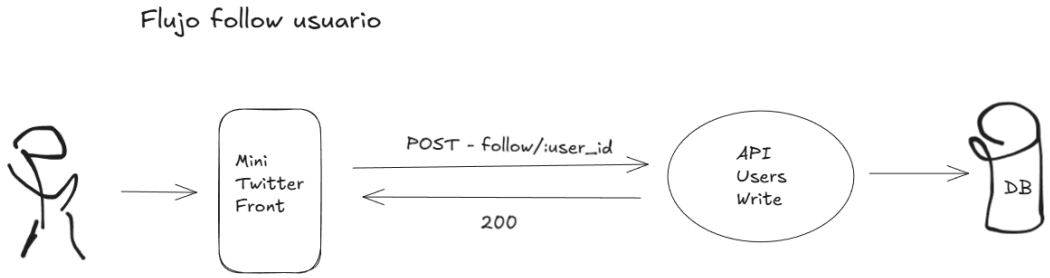
Para colas de trabajo se utiliza NATS, también por ser fácil para el desarrollo.

Para monitoreo las aplicaciones deben reportar métricas y logs al servicio utilizado por la compañía (datadog, grafana). De esta forma se pueden configurar alertas para dar rápida solución ante inconvenientes.

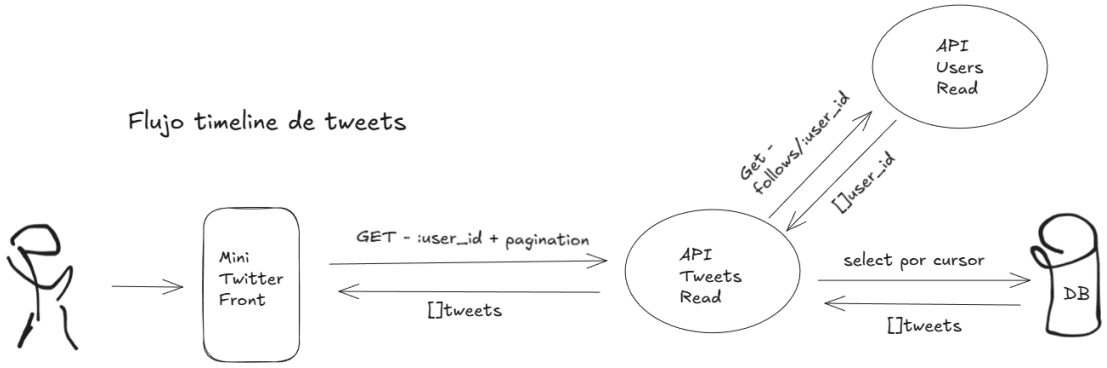
Diseño







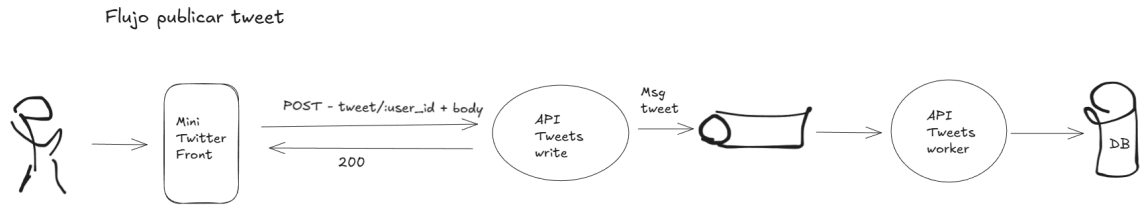
Se crean los ABMs para mantenimiento de usuarios y sus listas de seguidos. Se omiten temas de logueo y seguridad.



Para el timeline de tweets, el endpoint recibe el user_id que está en la aplicación. Con eso el proceso obtiene la lista de usuarios seguidos y hace el select a base de datos.

Para la consulta a base de datos se utiliza la metodología de cursor con la fecha de creación del tweet.

En este punto se podría pensar en implementar caches para lograr una mejor escalabilidad, tanto para la obtención de follows como para tweets más recientes.



Para el flujo de publicación de tweets la instancia HTTP recibe el POST (sería la instancia de write, pero en el ejemplo solo se levanta una general) y se genera un mensaje para el guardado del tweet async.

Un worker recibe el mensaje, hace las validaciones necesarias, y almacena el tweet en base de datos. Ante cualquier falla el mensaje se reintenta.

En este punto deben pensarse mecanismos de contingencia y alertas para cuando los mensajes no puedan ser procesados, y las colas comiencen a generar lag.

Terminado el guardado del tweet podría notificarse a otro tópico para dar aviso a otros usuarios, u otras aplicaciones que accionen ante esto.