Ejercicio técnico - Proceso de selección Ualá

Este desafío es un punto de partida para crear una plataforma similar a Twitter, pero aún es un proyecto considerable que requerirá tiempo y recursos significativos.

Objetivo

Crear una versión simplificada de una plataforma de microblogging similar a twitter que permita a los usuarios publicar, seguir y ver el timeline de tweets.

Requerimientos

Tweets

- Los usuarios deben poder publicar mensajes cortos (tweets) que no excedan un límite de caracteres (por ejemplo, 280 caracteres).

Follow:

- Los usuarios deben poder seguir a otros usuarios.

Timeline:

- Deben poder ver una línea de tiempo que muestre los tweets de los usuarios a los que siguen.

Assumptions

- Todos los usuarios son válidos, no es necesario crear un módulo de signin ni manejar sesiones. Se puede enviar el identificador de un usuario por header, param, body o por donde crea más conveniente.
- Pensar una solución que pueda escalar a millones de usuarios.
- La aplicación tiene que estar optimizada para lecturas.

Criterios de evaluación

- Armar documentación high level de la arquitectura y los componentes usados en la aplicación.

- La elección del lenguaje es libre. Podes elegir la tecnología que prefieras para implementarlo, ya sea utilizando un lenguaje de programación específico o un framework de desarrollo. A nivel infraestructura y protocolos es lo mismo, solo tiene que estar especificada en el documento high level. Ej: serverless, docker, kubernetes, message brokers, queues, bases de datos, cache, load balancers, gateways, grpc, websockets, etc.
- No hay que desarrollar un front end.
- Se pueden sumar assumption a los definidos y dejarlos plasmado en un archivo business.txt
- Nos interesa mucho la arquitectura interna y como están separados los layers. Clean architecture, DDD, arquitectura hexagonal, port and adapters, onion architecture, mvc, etc. Todas son válidas.
- A fines de simplificar pueden implementar una db in memory pero debería estar especificado en el doc del diseño que motor o tipo de db usarían y porque.
- Valoramos el testing, no es necesario que tenga 100% coverage, priorizamos los casos de usos principales. Test funcionales, de integración o aceptación también son bienvenidos pero no requeridos.

Consideraciones de entrega

- Para la entrega, se debe compartir un repositorio personal sobre el mismo mail donde se envía el ejercicio. Importante dejar el acceso público para su corrección.
- Proporcionar una documentación clara para levantar el proyecto en un readme.md.
- Se puede usar la wiki del repositorio para subir la documentación.
- Se valora hostear/dockerizar la/las aplicaciones.
- Ante cualquier consulta podes escribirnos a backendchallenge@uala.com.ar