Challenge Xcale

Gorno, Hector Guillermo

WhatsApp Messages – Notifications

Descripciones Generales

- La aplicacion que gestiona mensajes a usuarios que pertenencen a la misma

- Se plantean los siguientes tipos de usuarios:

  - usuario common : es una persona.
  - usuario group : es un grupo de usuario.

- En la base de datos embebida se configuraron los siguientes usuarios:

  - user1 (del tipo common) el cual tiene como contactos a: user2, user3 y user4
  - user2 (del tipo common) el cual tiene como contactos a: user4
  - user3 (del tipo common) el cual tiene como contactos a: user1 y user4
  - user4 (del tipo group) el cual tiene como contactos a: user1, user2 y user3

- Cuando los usuarios del tipo common envia un mensaje a otro del tipo common, el sistema persiste ese mensaje en la base de datos y marca el estado del mensaje con un valor inicial igual a "NOT_RECEIVED".

- El estado inicial del mensaje (NOT_RECEIVED) hace de bandera para poner al usuario la vez que consulte al servicio por mensajes que este tiene mensajes nuevos en el outbox para el por lo que el sistema se los devuelve.

- Cuando un usuario consulta en el servidor este chequea en la base de datos por aquellos mensajes cuyo estado es "NOT_RECEIVED" y su destinatario coincide con el id del usuario que realiza la consulta una vez encontrados se los devuelve y cambia el estado del mensaje a "RECEIVED"

- Cuando un usuario del tipo common envia un mensaje a otro usuario del tipo group, el sistema consulta por todos los N contactos del usuario group y persiste ese mismo mensaje enviado por el usuario common al usuario group N veces, con las mismas caracteristicas que el mensaje que envia un usuario del tipo common.

Casos de uso (algunos ejemplos happy path)

Caso 1

User1 envia un mensaje al User2

Request:

- Endpoint: http://localhost:8080/message/inbox/
- Metodo: POST
- Body:

```
{
  "destinationId" : 2,
  "originId" : 1,
  "messageContent" : "Hola como andas"
}
```

Response: No content

Caso 2

User2 checkea en el server por nuevos mensajes

Request:

- Endpoint: http://localhost:8080/message/outbox/2
- Metodo: GET
- Body: vacio

Response:

```
[
  {
    "id": 1,
    "destinationId": "USER2",
    "originId": "USER1",
    "referencedMessage": null,
    "messageContent": "Hola como andas"
  }
]
```

Caso 3

User1 envia un mensaje al grupo "Grupo de Trabajo"

Request:

- Endpoint: http://localhost:8080/message/inbox/
- Metodo: POST
- Body:

```
{
  "destinationId" : 4,
  "originId" : 1,
  "messageContent" : "Hola como andan!!!"
}
```

Response: vacio

Caso 4

User1, User2 y User3 checkean los nuevos mensajes

Request:

- Endpoint user1: http://localhost:8080/message/outbox/1,
- Endpoint user1: http://localhost:8080/message/outbox/2,
- Endpoint user1: http://localhost:8080/message/outbox/3,
- Metodo: GET
- Body: vacio

Response:

Body user1:

```
[
  {
    "id": 1,
    "destinationId": "USER1",
    "originId": "Grupo del Trabajo",
    "referencedMessage": null,
    "messageContent": "Hola como andan!!!"
  }
]
```

Body user2:


```
[
  {
    "id": 2,
    "destinationId": "USER2",
    "originId": "Grupo del Trabajo",
    "referencedMessage": null,
    "messageContent": "Hola como andan!!!"
  }
]
```


Body user3:


```
[
  {
    "id": 3,
    "destinationId": "USER3",
    "originId": "Grupo del Trabajo",
    "referencedMessage": null,
    "messageContent": "Hola como andan!!!"
  }
]
```