Parcial 2. Servicios Web

En este ejercicio desarrollarás servicios REST para *Instant*, una red social de mensajería instantánea similar a *WhatsApp* o *Telegram*, en la que los usuarios intercambian mensajes breves de texto a través de sus teléfonos móviles.

1. Diseño de la base de datos (2 puntos)

En *Instant* distinguiremos dos tipos de entidades:

- ***Usuarios.*** Representa los usuarios de la red social, con los siguientes atributos:
  - ***teléfono.*** Número de teléfono del usuario. Funcionará como identificador de la entidad.
  - ***alias.*** Nombre del usuario, tal como se presenta a otros usuarios de la red.
  - ***contactos.*** Lista de contactos del usuario, representado cada uno por su número de teléfono y su alias (nombre con el que lo designa este usuario).
- ***Mensajes.*** Representa los mensajes intercambiados, con los siguientes atributos:
  - ***identificador.*** Clave o identificador del mensaje, propia de la base de datos elegida.
  - ***timestamp.*** Fecha y hora en la que se envió el mensaje.
  - ***origen.*** Número de teléfono del usuario que envía el mensaje.
  - ***destino.*** Número de teléfono del usuario al que se envía el mensaje.
  - ***texto.*** Texto del mensaje (hasta 400 caracteres).

Estas entidades del modelo conceptual pueden dar lugar a un número mayor o menor de entidades de base de datos, con más o menos atributos, dependiendo de la base de datos elegida y el diseño creado para representarlos.

2. Servicios REST básicos (2 puntos)

Una vez diseñada la estructura de la base de datos, deberás desarrollar un proyecto que despliegue localmente un servidor REST cuyas operaciones devolverán sus resultados en JSON (opcionalmente, también en XML). Se valorará especialmente que los *endpoints* del servidor y sus posibles parámetros estén definidos de acuerdo a las normas de estilo propias de los servicios REST.

El servidor ofrecerá las siguientes operaciones básicas:

- CRUD de usuarios (GET ALL, GET, POST, PUT, DELETE).
- CRUD de contactos de un usuario (GET ALL, GET, POST, PUT, DELETE).
- CRUD de mensajes (GET ALL, GET, POST, PUT, DELETE).

3. Servicios REST adicionales (3 puntos)

Ampliarás la funcionalidad del servidor con las siguientes operaciones:

- Buscar un usuario de la red social a partir de su alias.
- Buscar entre los contactos de un usuario a partir de una cadena con parte de su alias, devolviendo una lista de contactos.
- Enviar un mensaje a un contacto de un usuario, a partir de su alias. El *timestamp* del mensaje se establecerá a la fecha y hora actuales.

- Obtener las conversaciones de un usuario, representadas como una lista de alias de contactos ordenada (descendente) por el *timestamp* del último mensaje enviado o recibido, de forma similar a como se muestra en la interfaz de WhatsApp o Telegram.
- Obtener la conversación de un usuario con uno de sus contactos, representada como una lista de mensajes ordenada (descendente) por *timestamp*.
- Buscar entre los mensajes enviados o recibidos por un usuario a partir de una cadena con parte de su texto, devolviendo una lista de mensajes.

4. Descripción y pruebas del servicio (3 puntos)

Todas las operaciones del servidor deberán poder probarse de forma independiente. Las pruebas pueden definirse a partir de una especificación *OpenAPI* o un conjunto de pruebas *Postman*, que acompañarán a la entrega del examen. Tanto en un caso como en otro deben de ser lo suficientemente descriptivas (incluyendo los comentarios y ejemplos que consideres necesarios) como para permitir la prueba del servidor de forma sencilla.

Entrega

Este ejercicio se entregará a través del campus virtual, mediante un archivo comprimido que contenga:

- Una memoria en la que describas:
  - el diseño de la base de datos (apartado 1) describiendo su estructura (entidades y atributos) y su URI o credenciales de acceso (por ejemplo, para MongoDB Atlas o Firestore). Si la base de datos es local, deberás adjuntar en la entrega del examen una descarga o *backup* de la misma.
  - las tecnologías utilizadas (lenguaje de programación, frameworks, etc.), y los *scripts* e instrucciones de instalación y despliegue (si son necesarias), así como el puerto de despliegue del servidor.
  - una relación de las limitaciones de la solución propuesta y los problemas encontrados durante su desarrollo (si procede).
- Las fuentes completas (no basta con una referencia a un repositorio GitHub) del servidor REST con las operaciones desarrolladas (apartados 2 y 3).
- La especificación OpenAPI del servidor, o un conjunto de pruebas Postman del servicio (apartado 4).