



Parciales 2 y 3. Diseño e implementación de TextME

En este ejercicio deberás desarrollar TextME, una aplicación de mensajería electrónica. Para ello, crearás una base de datos (no relacional, en caso de que tengas que realizar el tercer parcial) que almacenará una entidad **Messages**, con la siguiente información:

Cabecera (Headers)

- **id**. Identificador interno y único del mensaje (por ejemplo, ObjectID de MongoDB).
- **de**. Identificador del usuario redactor del mensaje (por ejemplo, email o Google ID).
- **para**. Identificador del usuario a quien se dirige el mensaje.
- **asunto**. Texto breve (de hasta 80 caracteres) con el título o asunto del mensaje.
- **stamp**. Fecha y hora, asignadas automáticamente al crear el mensaje.

Cuerpo (Body)

- **contenido**. Texto (sin restricciones de tamaño) del mensaje.
- **adjunto**. URL de una imagen (opcional) adjunta al mensaje.

En el campus virtual encontrarás *scripts* SQL y JSON para poblar la base de datos.

Funcionalidad común a los dos parciales

1. El *back-end* de la aplicación ofrecerá la siguiente funcionalidad:

- Obtener los datos de cabecera de los mensajes recibidos y enviados (indistintamente) por un usuario dado, en orden descendente de fecha y hora.
- Obtener los datos completos (cabecera y cuerpo) de un mensaje, a partir de su identificador.
- Crear un mensaje, a partir de los campos **de**, **para**, **asunto** y **contenido**.

Funcionalidad específica del parcial 2 (Servicios Web)

2. La funcionalidad del *back-end* se ofrecerá a través de una API diseñada de acuerdo a las normas de estilo **REST**. Además de lo indicado en el apartado 1, ofrecerá también la siguiente funcionalidad:

- Modificar el asunto de un mensaje a partir de su identificador y un nuevo título.
- Borrar un mensaje a partir de su identificador.
- Obtener los datos de cabecera de la conversación completa (conjunto de mensajes enviados y recibidos) que ha tenido lugar entre dos usuarios, a partir de los identificadores de ambos, en orden ascendente de fecha y hora.
- Obtener los datos de cabecera de los mensajes recibidos por un usuario y que están pendientes de respuesta, es decir, aquellos que no tienen un mensaje posterior al remitente del mensaje.
- Obtener los contactos de un usuario, esto es, la lista de usuarios con los que ha intercambiado mensajes.

3. La API desarrollada debe estar documentada mediante OpenAPI o mediante un conjunto de pruebas Postman, de manera que todas las operaciones puedan probarse de forma independiente. Aparte de esto, no es necesario desarrollar un cliente para la aplicación.



Funcionalidad específica del parcial 3 (Computación en la nube)

4. El *back-end* se desplegará en un proveedor *cloud PaaS público* (como por ejemplo, Heroku o Firebase). La base de datos donde se almacenen los mensajes será **no relacional**. En la entrega del examen se indicará la URL donde está desplegada la aplicación, así como las credenciales de la base de datos.
5. Se desarrollará también un *front-end* (local o alojado en un proveedor *cloud*) con la siguiente funcionalidad:
- Al conectarse a la aplicación, el usuario se identificará mediante OAuth 2.0. La aplicación permitirá a los usuarios hacer *login* y *logout*, todas las comunicaciones con el *back-end* serán autenticadas mediante el ID *token*, de forma que solo se podrá acceder a los mensajes enviados o recibidos por el usuario identificado.
 - No obstante, si debes realizar también el Parcial 2, deberás prever alguna forma de probar el servidor de forma independiente (tal como se ha indicado antes), por ejemplo, mostrando en la GUI o en la consola del cliente el ID *token* a utilizar en la cabecera *Authorization* de las operaciones. Recuerda explicar este aspecto en la memoria del examen.
 - Una vez identificado el usuario, se mostrarán los datos de cabecera de los mensajes recibidos o enviados (indistintamente), en orden descendente de fecha y hora. Seleccionando uno de estos mensajes se mostrará además su cuerpo, es decir, el contenido del mensaje y la imagen adjunta (en caso de que exista).
 - El *front-end* permitirá también crear un nuevo mensaje o para responder a un mensaje recibido. Para ello, se mostrará un formulario para introducir los datos del mensaje, con los campos **de**, **para** y **asunto** rellenados convenientemente según el caso.
 - Al redactar un mensaje se podrá adjuntar una imagen al mismo, seleccionándola de un archivo en el ordenador del usuario. La imagen se subirá al *back-end* o a un servicio externo (por ejemplo, Imgur) y la URL resultante se almacenará en la base de datos.

Entrega del ejercicio

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

- El **código fuente** completo que hayas desarrollado.
- El **script** o un *back-up* de la base de datos, si has modificado su diseño, y las **credenciales** para acceder a ella, en caso de estar alojada en la nube.
- Las **pruebas** Postman o la especificación OpenAPI del *back-end* (solo para el Parcial 2).
- Una pequeña **memoria** que indique las tecnologías utilizadas (lenguaje de programación, *frameworks*, base de datos, proveedor *cloud*, etc.), y las instrucciones de instalación (si son necesarias), y que describa además la funcionalidad que se ha implementado y las limitaciones de la solución propuesta o problemas encontrados durante su desarrollo.
- En la memoria recuerda indicar la **URL** donde está desplegada la aplicación (solo para el Parcial 3)