

**Instrucciones.** Lee cuidadosamente el problema que se presenta a continuación. Determina cuales son los requerimientos que se solicitan y desarrolla la solución apropiada.

“El multimillonario **Mark Zuckerberg** lo contrata para hacer su siguiente idea de negocio innovadora. Lleva por nombre **Juatsapp** y pretende que sea considerada la app de chats definitiva. La idea es muy sencilla, y permite que los usuarios se registren a través de una **GUI** proporcionando su teléfono, contraseña, fecha de nacimiento, una imagen de perfil, dirección completa y sexo (masculino, femenino, robot, ninja, otro). Dados las recomendaciones de seguridad es necesario que la **contraseña** se almacena de forma **encriptada** con algún mecanismo.

Una vez que un usuario ha sido creado, este podrá iniciar sesión con su cuenta y solamente podrá consultar y editar sus datos, sin opción de darse de baja. El **core** de la app es que los usuarios que inicien sesión puedan crear chats para comunicarse con otros usuarios (distintos a ellos mismos), y que una vez creado el chat, se puedan enviar mensajes tanto de emisor como de receptor con la opción de editar o eliminar tanto chats como los mensajes enviados en los mismos. Cabe mencionar, que un usuario debe poder visualizar la lista de chats en los que está participando, independientemente de si fue creado por él, o si fue invitado al mismo. Además el usuario debe poder ingresar a la lista de mensajes para consulta, y realizar el envío de nuevos mensajes si así lo desea. Los mensajes deberán contener el texto del mensaje, una imagen opcional y el usuario que lo envió, así como la fecha y hora de registro. Los chats deben incluir una imagen que se debe mostrar como miniatura junto a su nombre para facilitar la navegación de los usuarios dentro de la aplicación. Todas las imágenes deberán almacenarse dentro de la base de datos para mejorar la privacidad de la información.

Por último y como **requerimiento obligatorio**, la GUI deberá ser **similar a la de cualquier cliente** de mensajería del mercado, de otro modo se corre el riesgo de que el proyecto no sea aceptado por el cliente y la solución fracase.

Siendo el Sr. Zuckerberg un conocedor de tecnologías y partidario del movimiento Open Source, propone el uso de **Java** como lenguaje de programación, además de **MongoDB** como motor de base de datos, siendo este último moderno y con posibilidad de almacenar la información de diversas formas”.

## Consideraciones generales

1. El sistema tiene que ser una aplicación sencilla Java SE que utiliza Maven como gestor de dependencias, con interfaces gráficas de usuario, utilizar motor MongoDB con la librería MongoClient para el manejo de las operaciones de persistencia.
2. Para la selección de fechas en el sistema es necesario que se utilice un componente selector gráfico. Se recomienda utilizar LGoodDatePicker, pero la decisión final es del equipo diseñador del sistema.
3. Las librerías externas deben incluirse como dependencias en el archivo POM.
4. No es obligatorio que el sistema funcione en tiempo real actualizando los mensajes o chats creados nuevamente, pero se considerarán puntos extras en caso de lograrlo.
5. Incluir todas las validaciones correspondientes para cada interfaz gráfica como pueden ser: validación de datos de entrada (por ejemplo formato y campos vacíos), que no se permita guardar información repetida (por ejemplo que existan 2 usuarios repetidos), validaciones lógicas.
6. La usabilidad del sistema es de suma importancia, por lo tanto la información debe presentarse de forma legible incluyendo formatos de fechas. El sistema debería informar siempre de los resultados de las operaciones realizadas a los usuarios con mensajes claros y concisos.

## Entregables del proyecto obligatorios

1. Prototipos de interfaz gráfica coherentes que incluyan información de muestra.
2. Diagrama de clases de dominio.
3. Modelo entidad relación.
4. Mapeo de prototipos de interfaz gráfica con clases de dominio.
5. Proyecto Java SE de Netbeans con Maven como gestor de dependencias
6. URL del repositorio público de Github con el código fuente actualizado.
7. Documentación JavaDoc en formato HTML.

**NOTA:** En caso de no incluir algún entregable el proyecto no se calificará.

**ACLARACIÓN:** no se trata de que hagan el proyecto como tradicionalmente los hacen (cada quien en su pc, se pasan los cambios por drive o USB, luego los juntan) y suban todo al final al Git como si fuera Google Drive. Se trata de que vayan subiendo los cambios en Git gradualmente, de tal forma que la otra persona los descargue, agilice el desarrollo y mejore la mantenibilidad del software.