Vamos a realizar una aplicación distribuida para encontrar pareja. Será una aplicación con pocas funcionalidades, más bien un sistema de mensajería aunque sí que pedirá preferencias y mostrará personas por afinidad. También deberemos idear algún sistema en el que si dos personas muestran interés se establezcan como amigos y puedan verse el uno al otro cuando ingresen en el sistema.

La aplicación constará de:

* Un formulario de inicio que permitirá entrar en el sistema. En esta página de inicio se permitirá también enlazar a la ventana de registro.
* El registro permitirá dar los datos personales (incluyendo *email* que servirá como usuario de inicio de sesión). La clave del usuario irá resumida en la base de datos. Los datos de registro irán cifrados y firmados.
* Una vez dentro del sistema como usuario estándar tendremos las siguientes funcionalidades:
  + Ver usuarios de mi preferencia lo más afines posible a mis gustos.
  + Permitir establecer qué usuarios me gustan (o dejan de gustar) con un simple click. En caso de que sea recíproco, nos convertiremos automáticamente en amigos.
  + Deberíamos poder listar los amigos que tenemos. El formato será libre.
  + Enviar mensajes a otros usuarios (amigos o no).
  + La primera vez que entramos al sistema nos pedirá que rellenemos un formulario con nuestras preferencias. Estas preferencias se podrán editar siempre, así como nuestro perfil (foto, nick, clave de acceso…).
* Como administrador podemos, con un CRUD, gestionar a los usuarios: altas, bajas, modificaciones, activaciones… los usuarios estarán desactivados por defecto hasta que los active el administrador. También se podrán crear a otros administradores (o quitarles los privilegios a los actuales).
* Los usuarios podrán enviar mensajes a otros usuarios y ver los mensajes que otros usuarios les han enviado. A estos mensajes se les podrán adjuntar archivos, lo obligatorio es que se pueda adjuntar al menos uno; como optativo, pero muy valorado, es que permita más de un archivo adjunto.
* Las preferencias que un usuario podrá poner son las siguientes:
  + Relación seria 🡪 indicarán si su finalidad es una relación seria o esporádica.
  + Deportivos 🡪 valor numérico que indica, de 0 a 100, su gusto por los deportes.
  + Artísticos 🡪 valor que indica su inquietud artística.
  + Políticos 🡪 interés por la política, valorando de 0 a 100.
  + Tiene/Quiere hijos o no 🡪 Indicando si tiene y/o si quiere hijos.
  + Interés en: hombres, mujeres o ambos.

Todos estos valores se tendrán en cuenta a la hora de mostrar a posibles parejas candidatas en la pantalla principal del usuario.

* Los usuarios verán, por lo tanto, a los candidatos más afines a sus propios valores. La forma de hacer esta parte es libre, se valora el algoritmo ideado.

(Si el valor es menor de 25, no le gusta, si es mayor que 25 tiene poco interés, si esta entre 25 y 50 le gusta, si esta entre 50 y 75 le gusta mucho, si es mayor que 75, es su tema favorito).

* La aplicación permitirá un número indefinido de conexione múltiples. Será controlado el acceso a la base de datos de forma exclusiva por cada usuario, cuando sea necesario.
* Todas las comunicaciones irán cifradas.

Se tendrá en cuenta:

* Programación clara y ajustada al MVC.
* Diseño de la BD que recoja toda la información de forma correcta.
* Que lo programado se ajuste a lo pedido.
* Una planificación de tareas coherente.
* El proyecto deberá estar sincronizado con un repositorio Git y tener una rama, mínimo, de desarrollo. Después de cada tarea planificada se debería hacer un merge a main (estaría bien que se usaran los board de GitKraken investigando su sincronización con GitHub).