

# ■ Práctica 1: Aplicaciones basadas en Componentes Distribuidos.

*Victor Rampérez Martín- Isidoro González Irisarri*

*Curso: 2014-2015*

Middleware

ETSIINF-UPM

Madrid

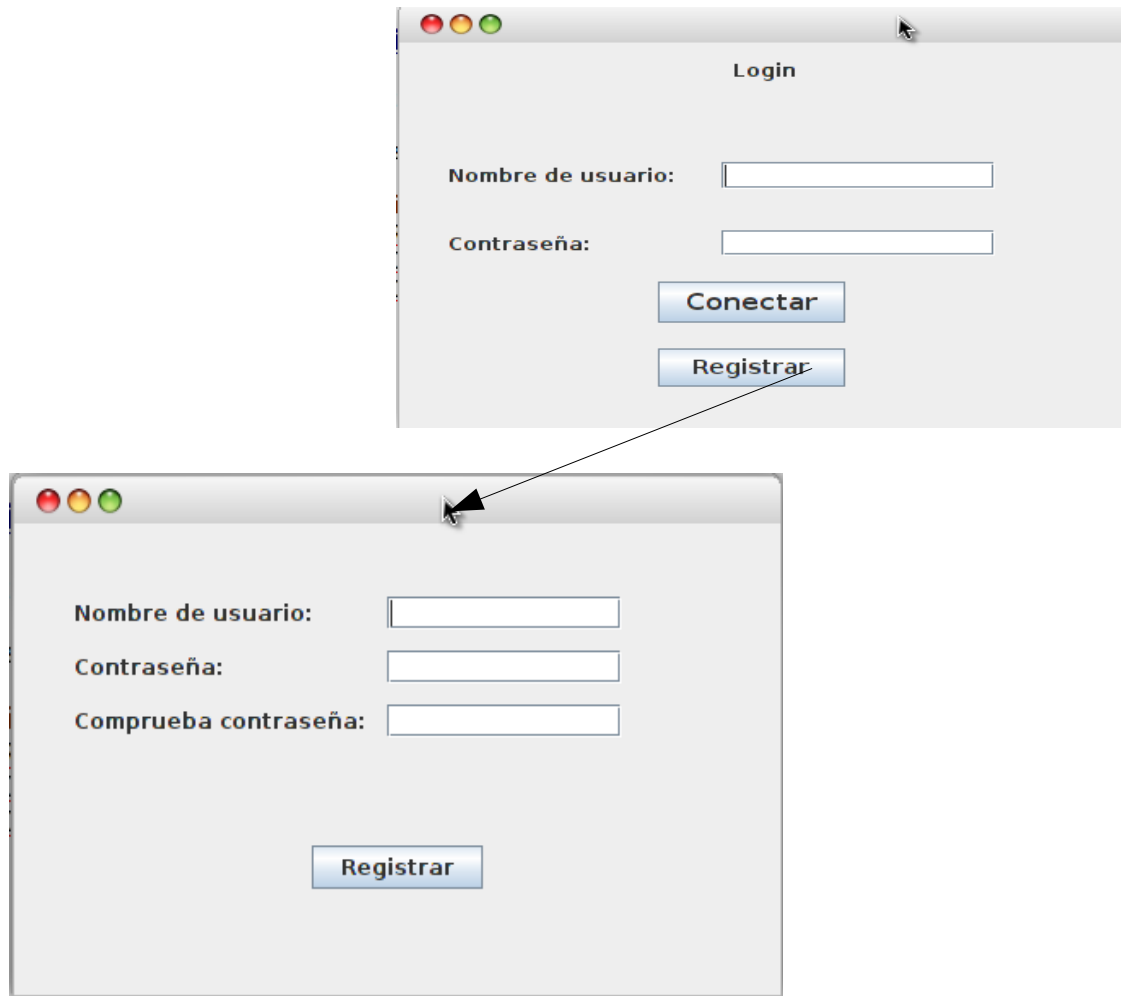
09/11/2014

## 1. Definición de la interfaz del sistema.

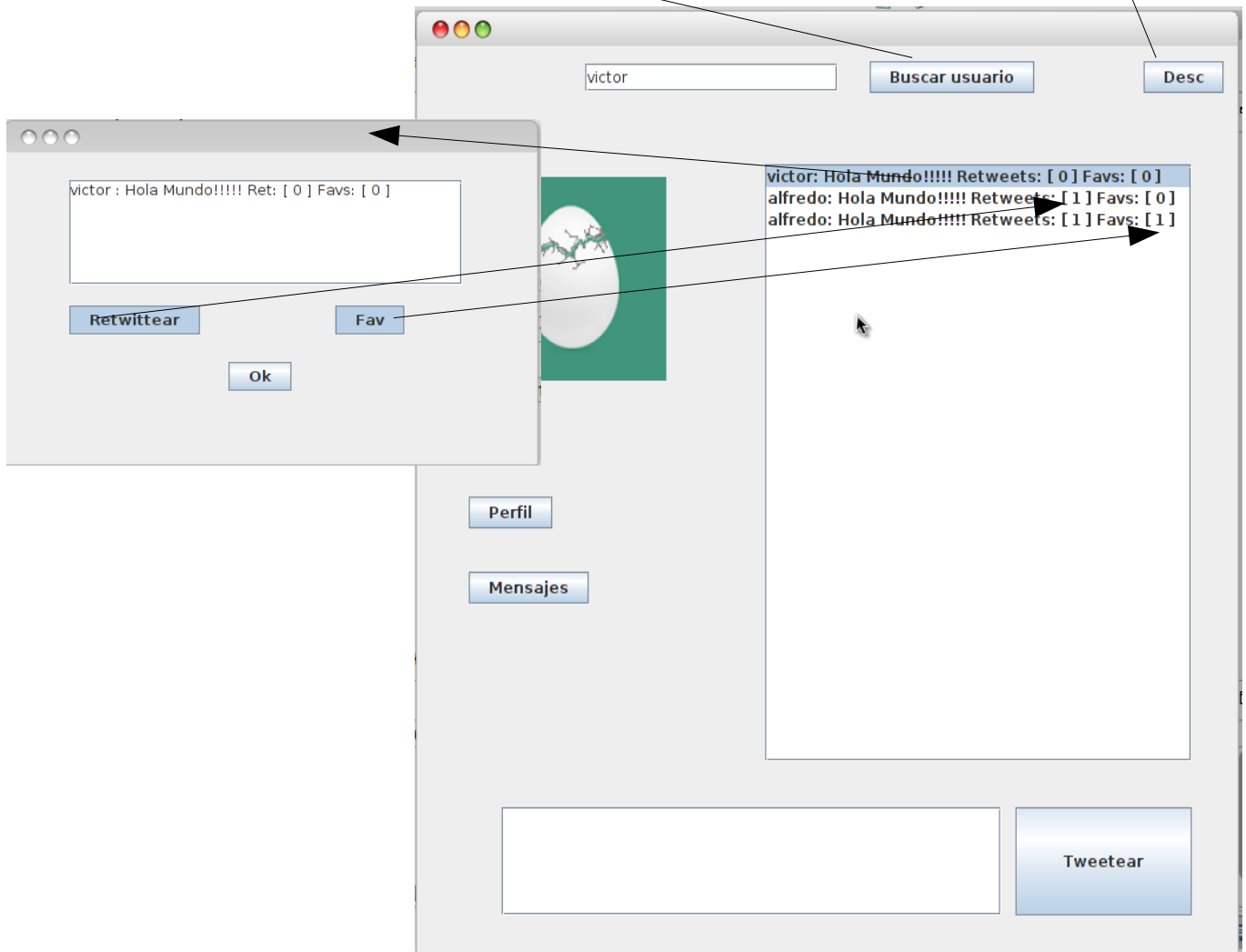
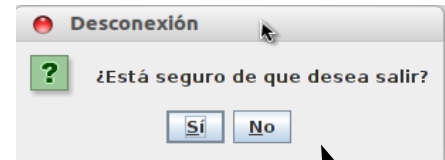
El sistema está dividido en tres paquetes: uno gráfico, uno de servidor y otro de cliente.

Centrándonos en las funcionalidades de la interfaz para el cliente tenemos:

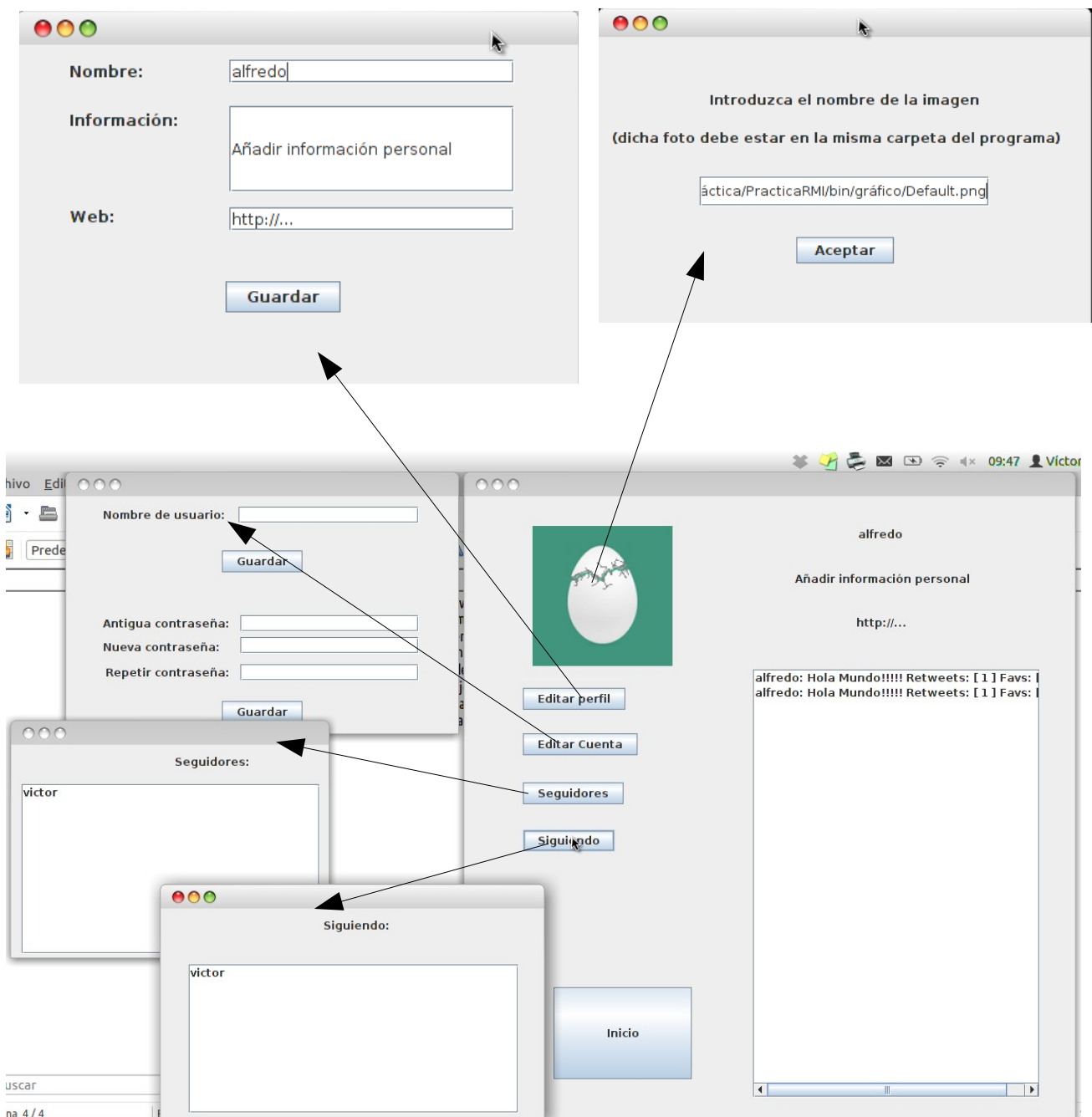
- Funciones de login:
  - Registrar: para registrarnos en el sistema.
  - Conectar: para conectarnos al sistema con nuestro usuario registrado.



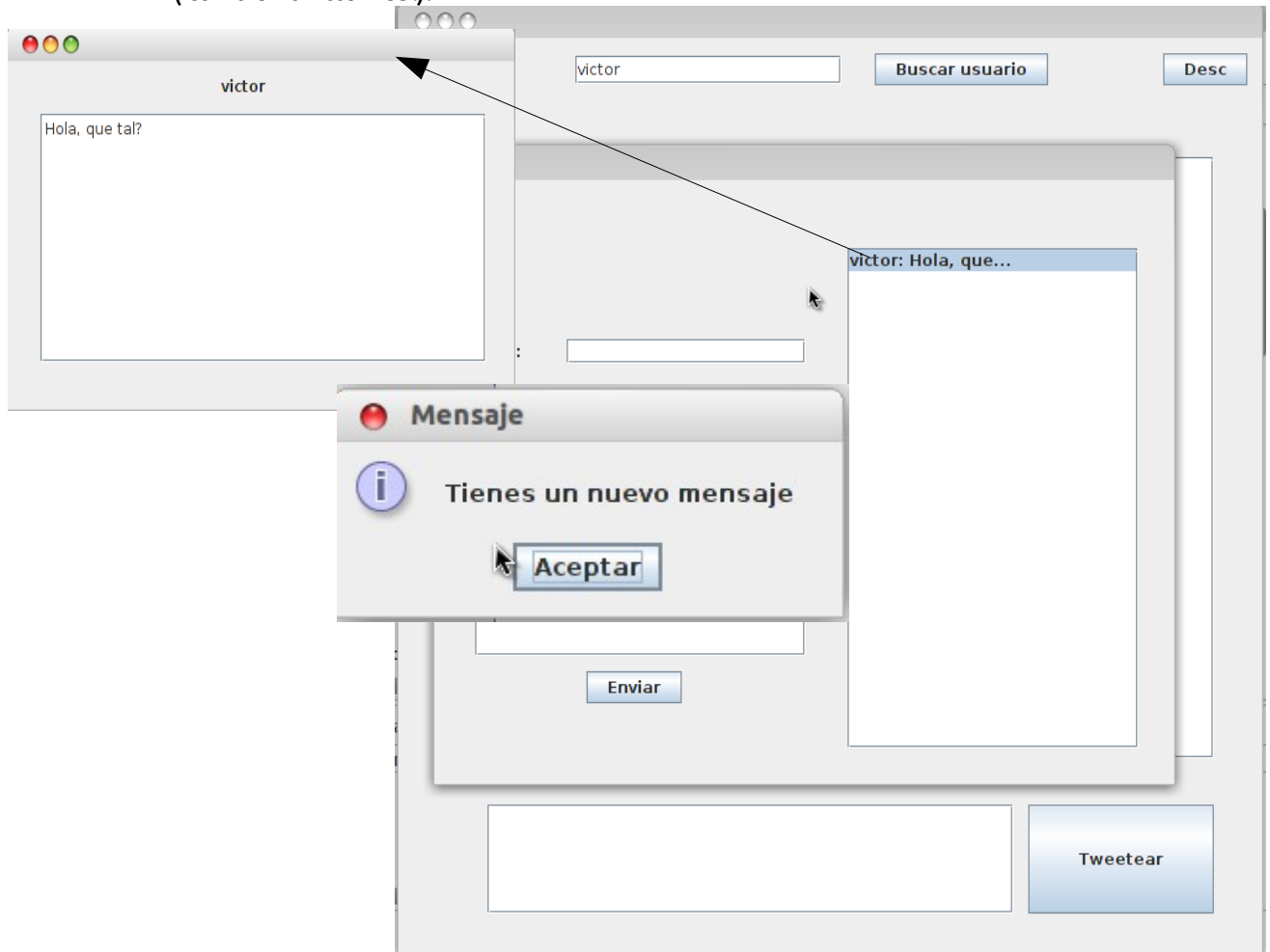
- Funciones dentro del sistema:
  - Desc: para desconectarnos del sistema.
  - Buscar usuario: permite buscar a otros usuarios registrados y poder ver su perfil.
  - Twittear: permite enviar un tweet a todos tus seguidores.
  - Timeline: muestra en tiempo real cada uno de los tweets de los usuarios a los que sigues, pudiendo ver cuando son retwiteados y marcados como fav.



- Perfil: permite acceder a nuestro perfil donde podemos ver:
  - Información personal: nombre real, web y otro información que ugeramos añadir.
  - Editar nuestro perfil: cambiar foto de perfil(haciendo click), información personal, nombre, web....
  - Editar cuenta: cambiar nombre de usuario y contraseña.
  - Seguidores: muestra todos los usuarios que te siguen actualmente.
  - Siguiendo: muestra todos los usuarios a los que sigues actualmente.
  - MisTweets: permite ver todos los tweets que has publicado y el numero de veces que han sido retwitteados y marcados como favoritos.



- Mensajes: permite acceder a tu mailbox. El mailbox ofrece:
  - Ver todos tus mensajes. Haciendo click obtendrás el mensaje completo.
  - Escribir un mensaje a otro usuario. NOTA: el destinatario debe ser una persona que te siga ( como en twitter real).

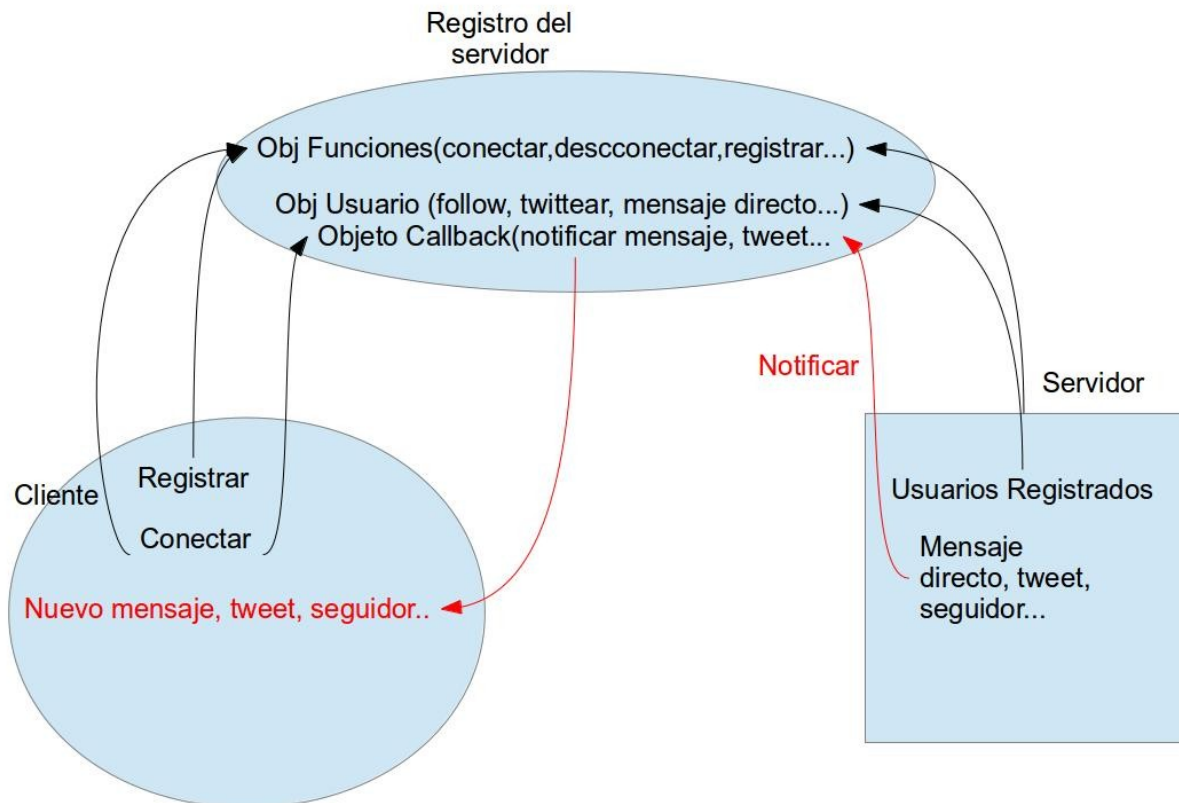


## 2. Explicación de la API RMI diseñada.

En primer lugar al arrancar el servidor este coloca un objeto de la clase Funciones (ver punto 3). Posteriormente el cliente llamará remotamente a dicho objeto para registrarse o conectarse. Al conectarse el servidor coloca en el registro el objeto correspondiente de la clase User del usuario y este coloca un objeto de la clase Callback.

Una vez realizado el login el cliente podrá llamar a métodos remotamente del objeto user (tweeetear, enviar mensajes, seguir usuarios...) y cuando sea necesario el servidor le llamará a los correspondientes métodos de notificación del objeto callback del cliente.

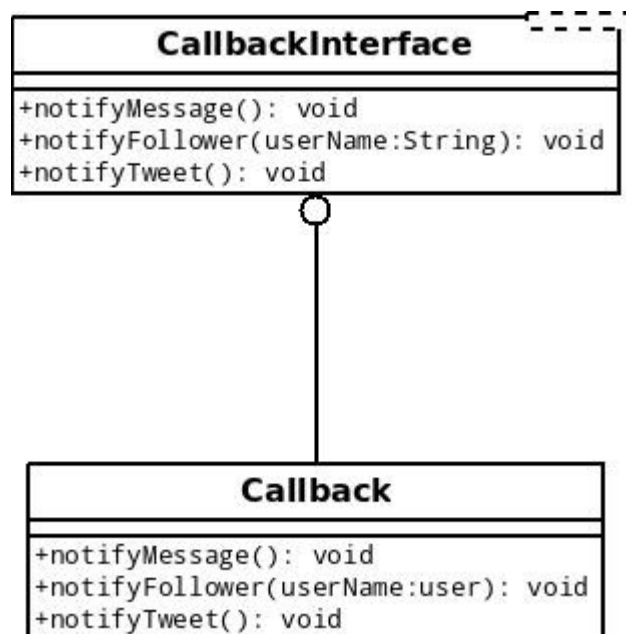
Cuando el cliente se desconecta se quita el objeto callback del registro.



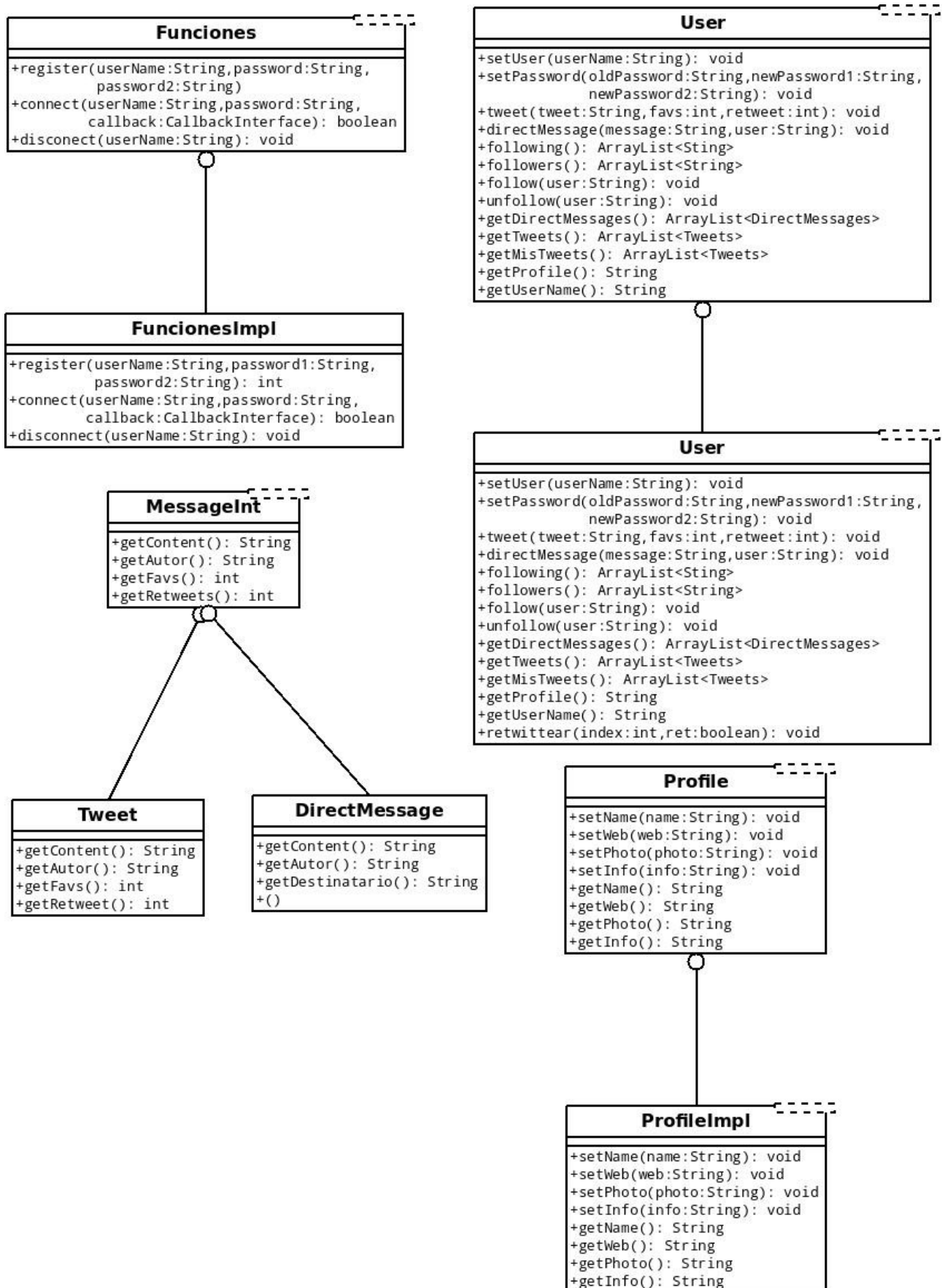
### 3. Diagrama de clases de los componentes distribuidos.

Solo se representan en el el diagrama las clases del servidor y cliente, es decir, todas menos las de la parte gráfica, para no complicar el diagrama.

Cliente:



Servidor:



#### 4. Opinión personal.

Nos ha parecido una práctica muy interesante y sobre todo nos ha gustado el alto grado de libertad de la misma, ya que no se trata de una práctica dirigida sino que podíamos diseñar desde cero el funcionamiento de nuestro sistema. Por otra parte nos ha parecido muy útil lo aprendido de java RMI.

Finalmente nos hubiese gustado algo más de tiempo para poder haber mejorado algunas funcionalidades, principalmente de la interfaz gráfica (pues es la primera vez que hacemos una).