



Nombre	Usuario	Hora de entrega

Diseñar una aplicación java que permita simular el funcionamiento de la red profesional linkedin. Para ello hay que tener en cuenta las siguientes consideraciones.

- Los Usuarios que se registran en la red pueden ser personas físicas o empresas.
- Los datos que se necesitan tener registrados por usuario, independientemente de su tipo, serán: identificador del usuario (cadena con el email) y nombre. Además se almacenarán sus contactos (usuarios con los que está en contacto) y las solicitudes de contacto que tiene pendientes, serán los usuarios que le han solicitado estar en contacto con ellos y para los que todavía no ha aceptado su solicitud.
- No podrán registrarse dos usuarios con el mismo identificador.
- Si el usuario es una empresa se almacenará las solicitudes de empleo que oferta la empresa. Para cada solicitud se almacena el perfil que se solicita(cadena) y los años de experiencia necesarios para cubrir el puesto
- Por otro lado, las personas físicas pueden ser de dos tipos: estudiantes o profesionales.
- Para los estudiantes se almacenará también la titulación en la que están matriculados y el curso en el que se encuentran
- Para los profesionales se almacenará adicionalmente la empresa en la que trabajan y el año en el que entraron a trabajar en ella.
- Para crear un usuario Empresa será obligatorio especificar su identificador, nombre; para crear un Estudiante habrá que indicar obligatoriamente su identificador, nombre, titulación y curso; por último, para crear un usuario Profesional habrá que especificar su identificador, nombre, empresa en la que trabaja y año en el que entró en la empresa.

Las funcionalidades que linkedin debe permitir son las siguientes:

- addUsuario: esta funcionalidad permite añadir un usuario. Recibirá como parámetro el usuario a añadir (en realidad este objeto será una Empresa, un Estudiante o un Profesional)

Comprueba las siguientes condiciones de error notificándolas mediante excepciones:

UsuarioRepetidoException. se propaga esta excepción cuando en el linkedin ya existe otro usuario con ese identificador.

- buscarUsuario: esta funcionalidad buscará y devolverá el usuario identificado a través del identificador que se reciba como parámetro.

Comprueba las siguientes condiciones de error notificándolas mediante excepciones:

· *UsuarioNoEncontradoException*: se propaga esta excepción cuando no encuentre el usuario. Esta excepción tendrá además una propiedad de lectura que representa el identificador del usuario no localizado.

- solicitarContacto: esta funcionalidad recibirá el identificador del usuario que quiere solicitar un contacto y el identificador del usuario con el que quiere contactar. Si no hay ningún error se añadirá como solicitud a las solicitudes de contacto del usuario con el que se quiere conectar.

Comprueba las siguientes condiciones de error notificándolas mediante excepciones:

· *UsuarioNoEncontradoException*: se propaga esta excepción cuando no encuentre el usuario. Esta excepción tendrá además una propiedad de lectura que representa el identificador del usuario no localizado. (Para cualquiera de los dos usuarios que se pasan como parámetro).

ContactoYaExistenteException: se propaga esta excepción cuando los usuarios que se pasa como parámetro ya están en contacto.

- aceptarContacto: esta funcionalidad recibirá como parámetro el identificador del usuario que quiere aceptar un contacto y el del usuario al que se la acepta. Si no hay ningún error se añadirá un usuario a los contactos del otro y viceversa, además se eliminará ese usuario de las solicitudes de contacto.

Comprueba las siguientes condiciones de error notificándolas mediante excepciones:

· *UsuarioNoEncontradoException*: se propaga esta excepción cuando no encuentre el usuario. Esta excepción tendrá además una propiedad de lectura que representa el identificador del usuario no localizado. (Para cualquiera de los dos usuarios que se pasan como parámetro).

PeticionDeContactoNoExistenteException: se propaga esta excepción cuando el usuario que se pasa como parámetro no tenga pendiente un contacto con el otro usuario que se pasa como parámetro.

- *addOfertaEmpleo*: esta funcionalidad recibe como parámetro el identificador de la empresa que realiza la oferta y la Oferta. Si no hay ningún error añade esa oferta a las ofertas de la empresa.

Comprueba las siguientes condiciones de error notificándolas mediante excepciones:

· *UsuarioNoEncontradoException*: se propaga esta excepción cuando no encuentre el usuario. Esta excepción tendrá además una propiedad de lectura que representa el identificador del usuario no localizado.

· *UsuarioNoEsEmpresaException*: se propaga esta excepción cuando el usuario referenciado no es una Empresa.

- *mostrarOfertasEmpleo*: esta funcionalidad recibe como parámetro el identificador de un usuario y muestra por pantalla las ofertas de empleo que tienen las empresas con las que está en contacto.

Comprueba las siguientes condiciones de error notificándolas mediante excepciones:

· *UsuarioNoEncontradoException*: se propaga esta excepción cuando no encuentre el usuario. Esta excepción tendrá además una propiedad de lectura que representa el identificador del usuario no localizado.

- *listarUsuarios*: esta funcionalidad listará los datos de cada uno de los usuarios registrados: tipo de usuario (Estudiante, Profesional o Empresa), identificador, nombre y contactos. Este listado se mostrará ordenado por el nombre del usuario (hay que tener en cuenta que puede haber varios usuarios con el mismo nombre). Además si es un estudiante se mostrará la titulación en la que está matriculado y el curso; si es un profesional la empresa en la que trabaja y el año en el que entró.

Crear un programa Java en el que compruebes el correcto funcionamiento de tu linkedin

Para entregar la aplicación:

Debes comprimir tu aplicación NetBeans que obligatoriamente tendrá como nombre tus apellidos y nombre (si te llamas Juan Carlos García Vaquero, tu proyecto debe llamarse GarciaVaqueroJuanCarlos). El fichero ZIP que subirás al Moodle también tendrá el mismo nombre que tu proyecto.

Calificación:

Aprobado: debes implementar correctamente las funcionalidades: *addUsuario*, *buscarUsuario*, *solicitarContacto*, *aceptarContacto*

Notable: además de lo anterior, debes implementar correctamente las funcionalidad *addOfertaEmpleo*, *mostrarOfertasEmpleo* y *listarUsuarios*

Sobresaliente: Por defecto todas las funcionalidades devolverán verdadero o falso (a excepción de *buscarUsuario* que devolverá null o un objeto Usuario). Se obtendrá esta calificación si se implementan todas las funcionalidades propagando las excepciones descritas.