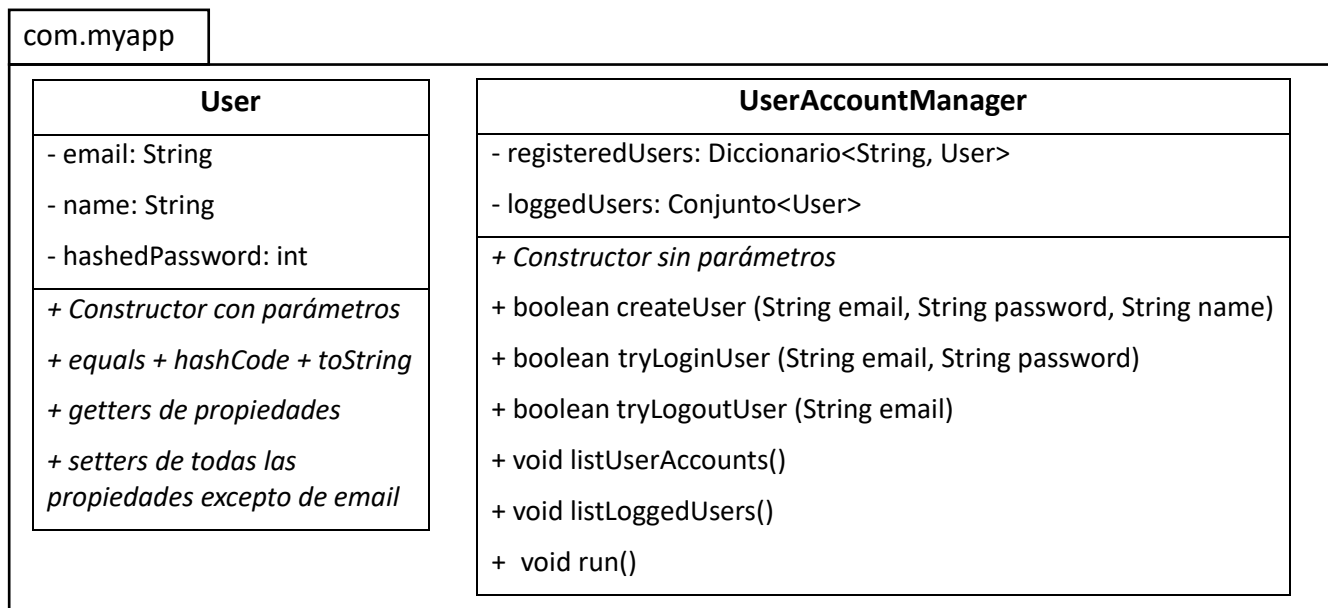


EJERCICIO EVALUABLE UD 4 – 20/02/2023

Debes realizar una aplicación de gestión de cuentas de usuario utilizando la **notación de genéricos** e implementando el siguiente diagrama de clases:



La clase **User** representa a un usuario de la aplicación que podrá iniciar sesión introduciendo su dirección de email y su contraseña. Por motivos de seguridad, esta clase no guarda la password (String) que teclea el usuario, lo que se guarda es el código hash (int) de la password tecleada.

La clase **UserAccountManager** tiene dos colecciones de información: *registeredUsers* almacenará en un diccionario a los usuarios registrados en el sistema utilizando su email como clave. Por otro lado, *loggedUsers* será el conjunto que guarde los usuarios que hayan iniciado sesión en la aplicación y aún no la hayan cerrado.

Los métodos de la clase **UserAccountManager** deben cumplir las siguientes especificaciones:

- El **constructor** debe inicializar las dos colecciones que contiene la clase como propiedades.
- El método **createUser** devolverá *true* si consigue crear y almacenar en el diccionario un nuevo usuario a partir de los parámetros recibidos. Este método debe comprobar si ya existe un usuario registrado con ese email, si la password tiene menos de 8 caracteres, si el nombre es la cadena vacía y si la dirección del email no contiene la @. Si ocurre alguna de estas circunstancias se debe emitir un mensaje de error especificando lo sucedido y devolver *false*.
- El método **tryLoginUser** devolverá *true* si se consigue iniciar sesión con los parámetros recibidos. Para iniciar sesión correctamente debe existir un usuario registrado con la dirección de email recibida, no debe haber iniciado sesión previamente y el código hash de la password introducida como parámetro debe coincidir con el que almacena la propiedad *hashedPassword* del usuario. Solo entonces se actualizará el conjunto *loggedUsers* para reflejar el nuevo inicio de sesión. Si alguna de las condiciones anteriores no se cumple se emitirá un mensaje de error especificando lo sucedido y se devolverá *false*.
- El método **tryLogoutUser** devolverá *true* si se consigue cerrar la sesión asociada al email recibido como parámetro, actualizando las colecciones que correspondan. Si el email no se corresponde con algún usuario registrado o bien sí se corresponde, pero ese usuario no había iniciado sesión previamente se emitirá un mensaje de error especificando lo sucedido y se devolverá *false*.

- El método ***listUserAccounts*** mostrará el encabezado “List of user accounts:” y a continuación se listarán los usuarios registrados en el sistema. En el caso de que no haya ningún usuario registrado se mostrará el mensaje: “Zero user accounts registered at the moment”
- El método ***listLoggedUsers*** mostrará el encabezado “List of logged users:” y a continuación se listarán los usuarios que actualmente tenga una sesión abierta en el sistema. En el caso de que no haya ningún usuario con una sesión abierta se mostrará el mensaje: “Zero logged users at the moment”
- El método ***run*** se encargará de **toda la interacción con el “humano”**, ofreciendo el siguiente menú de opciones:

1 - Create a user account

2 - User login

3 - User logout

4 - List of user accounts

5 - List of logged users

6 – Exit

El menú se repetirá hasta que se seleccione la opción 6 y se emitirá un mensaje de error si se escoge una opción incorrecta. Una vez se elija una opción válida, se pedirán todos los datos necesarios por teclado y se llamará al método que realice la acción correspondiente.

Para aquellos métodos que devuelvan un *boolean* se recogerá el valor devuelto y se mostrará un mensaje que describa lo sucedido. Por ejemplo, para el método *createUser* se mostrará “User account created succesfully” si devolvió *true* o bien “Unable to create the user account” si devolvió *false*.

Por último, debes crear la clase **Application** que lance la ejecución con el siguiente código:

```
public class Application {  
    public static void main(String[] args) {  
        UserManager uam = new UserManager();  
        uam.run();  
    }  
}
```

NOTA: puedes poner todos los mensajes en español o en inglés, pero no mezclar ambos idiomas. El nombre de las clases, propiedades y métodos debe ser exactamente el indicado en el diagrama.