

Práctica 7: Pruebas de integración (top-down)

Ángel Joaquín Rodríguez Mercado
Miguel Oña Guerrero

27 de mayo de 2022

1. Casos de prueba:

En las pruebas de integración top-down propuestas, se va añadiendo de forma procedimental un módulo distinto del programa a probar. Empezando por una versión reducida, en la que solo se tiene en cuenta la fecha. A continuación se añade la validación de la contraseña y por último además se comprueba si la credencial ya se encuentra registrada. Para sustituir a los módulos no probados en cada prueba, se emplea un objeto stub creado con Mockito.

Para terminar se realiza una prueba en la que todas las condiciones se cumplen y por lo tanto el usuario es registrado.

Las pruebas implementadas son las siguientes:

1. **Given an incorrect date, When register(), Then return BIRTHDAY_INVALID**

Se comprueba que dada una fecha incorrecta, el usuario no se registra y se devuelve el error de fecha inválida. La contraseña y almacén de credenciales son objetos stub.

2. **Given a correct date but an incorrect password, When register(), Then return PASSWORD_INVALID**

Se comprueba que dada una fecha correcta pero una contraseña incorrecta, el usuario no se registra y se devuelve un error de contraseña inválida. El almacén de credenciales es un objeto stub.

3. **Given a correct date and a correct password but credential exists, When register(), Then return EXISTING_CREDENTIAL**

Se comprueba que dada una fecha correcta, y una contraseña correcta pero las credenciales ya se encuentran en el almacén de credenciales, el usuario no se registra y se devuelve el error de credencial ya existente. En esta prueba no se emplean objetos stub.

4. **Given a correct date and a correct password and a not existing credential, When register(), Then return VALIDATION_OK**

Se comprueba que cuando se cumplen las tres condiciones (fecha, contraseña y credenciales no existentes), el usuario se registra correctamente y se devuelve el código de validación satisfactoria.

2. Modificación del código:

Se ha modificado el método `register()` de la clase `UserRegistration`, para que este devuelva por el `return` el valor del enumerado `status`. De esta forma se puede comprobar de forma más fácil cuál ha sido la condición que no se ha cumplido, o si se ha realizado con éxito el registro.

Además en la clase `CredentialValidator`, en el método `validate()` se ha modificado el `else if` que comprueba si las credenciales ya se encuentran en el almacén. Se ha quitado la negación de la condición, para que efectivamente se devuelva el valor de error `EXISTING_CREDENTIAL` en caso de que esas credenciales existan ya en el almacén.