

Desligamiento de los identificadores numéricos en el marco del testing

En este proyecto, como ampliación respecto del A+, se ha decidido además implementar en el proyecto un método para evitar depender del uso de los identificadores numéricos a la hora de realizar tests. Esto se debe a que un mínimo cambio en la población de la base de datos puede hacer que la testing suite al completo deje de funcionar.

Para ello, se ha decidido realizar pequeños cambios en la utilidad PopulateDatabase.java así como en AbstractTest.java

```
protected static void persist(final DatabaseUtil databaseUtil, final List<Entry<String,
String name;
DomainEntity entity;
//Properties object definition
final Properties properties = new Properties();
OutputStream output = null;

try {
    // Output file definition
    output = new FileOutputStream("src\\test\\resources\\populate.properties");

    System.out.println();
    databaseUtil.openTransaction();

    for (final Entry<String, Object> entry : sortedEntities) {
        name = entry.getKey();
        entity = (DomainEntity) entry.getValue();

        System.out.printf("> %s", name);
        databaseUtil.persist(entity);
        System.out.printf(": %s\n", entity.toString());
        //Register of the bean id against the identifier in the properties
        properties.setProperty(name, String.valueOf(entity.getId()));
    }

    //Storage of the file.
    properties.store(output, null);
    databaseUtil.closeTransaction();
    System.out.println();

} catch (final IOException e) {
    e.printStackTrace();
} finally { // Closing the file.
    if (output != null)
        try {
            output.close();
        } catch (final IOException e) {
            e.printStackTrace();
        }
}
```

En cuanto a las modificaciones en PopulateDatabase.java, en el método de persist ahora se crea un archivo .properties donde se guardará la relación entre la bean id del populate.xml con su identificador único. Este archivo se guardará en src/test/resources.

```

application4=140
application3=139
application2=138
application1=137
post8=128
post7=127
post6=126
post5=125
post4=124
post3=123
post2=122
post1=121
systemConfiguration=144
customer6=88
customer5=87
customer4=86
customer3=85
customer2=84
customer1=83
admin2=82
admin1=81

```

Este sería el formato final de este archivo una vez poblada la base de datos, no se imprimen en orden, pero como veremos ahora, eso no es ningún problema.

```

private static Properties prop;

// Set up and tear down -----

@BeforeClass
public static void setUpClass() {
    AbstractTest.prop = new Properties();
    InputStream input = null;

    try {

        input = new FileInputStream("src/test/resources/populate.properties");

        // load a properties file
        AbstractTest.prop.load(input);

    } catch (final IOException ex) {
        ex.printStackTrace();
    } finally {
        if (input != null)
            try {
                input.close();
            } catch (final IOException e) {
                e.printStackTrace();
            }
    }
}

```

Por último, esta sería la modificación principal de la clase AbstractTest, en la cual se carga, cada vez que se ejecuta la clase de test correspondiente, el archivo de properties, del cual se extraerá, con otro método la id del bean correspondiente.

```

public int extract(final String beanName) {
    int result;

    result = Integer.valueOf(AbstractTest.prop.getProperty(beanName));

    return result;
}

```

Siendo este el método que se usa en los tests.

```

// templateCreation
protected void templateCreation(final String username, final String title, final String
    Class<?> caught;
    caught = null;
    try {
        this.authenticate(username);
        final Customer cus = this.customerService.findOne(this.extract("customer2"));
        final Commentable com = cus;
        final Comment c = this.commentService.create(com);
        c.setBanned(banned);
        c.setActor(cus);
        c.setMoment(moment);
        c.setStars(stars);
        c.setText(text);
        c.setTitle(title);

        this.commentService.save(c);
        this.commentService.flush();
        this.unauthenticate();
    }
}

```

Uso de la extracción en la plantilla de creación de comentarios.