

E6 - DOCUMENTO DE PRUEBAS

6.1. Introducción

En nuestro documento de pruebas, vamos a detallar todos los casos de prueba realizados en la implementación de las clases del modelo en las funcionalidades que tiene la aplicación (pruebas unitarias y pruebas funcionales) y en la integración de componentes (pruebas de integración) y fichero pom.xml preparado para el lanzamiento de todas las pruebas.

6.2. Generación de casos de prueba

Los casos de prueba mencionados, son automáticamente generados por el framework “SpringRoo”, pero para que se creen debemos introducir unos comandos, dependiendo del tipo de pruebas que queramos obtener.

6.2.1. Realización de pruebas unitarias

En este tipo de pruebas, como hemos mencionado anteriormente, son casos de prueba aplicados a la implementación de las clases del modelo. Se ejecutan para cada clase, y se realizan con el siguiente comando para todas las entidades:

```
test mock --entity ~.domain.Entidad
```

6.2.2. Realización de pruebas funcionales

Este otro tipo de pruebas, son casos de prueba aplicados a las funcionalidades de la aplicación. Estas también se ejecutan para cada clase, y se realizan con el siguiente comando para todas las entidades:

```
test integration --entity ~.domain.Entidad
```

6.2.3. Realización de pruebas de integración

Las pruebas de integración son test aplicados a la integración de los componentes, por lo que no se ejecutan en nuestro entorno, sino en otro distinto. Nosotros en nuestro caso, vamos a realizar estas pruebas con la herramienta Selenium. Las pruebas de integración se ejecutan para el controlador de cada entidad, y se realizan de la siguiente manera para todos los controladores:

```
selenium test --controller ~.web.EntidadController
```

6.3. Ejecución de los casos de prueba

6.3.1. Pruebas unitarias y funcionales

Ambos tipos de pruebas se ejecutan de la misma manera, escribiendo el siguiente comando:

```
mvn test
```

6.3.2. Pruebas de integración

En cambio, las pruebas de integración, se realizan escribiendo el siguiente comando:

```
mvn selenium:selenese
```

Fichero de registros para comandos usados en las pruebas:

// Spring Roo 1.3.2.RELEASE [rev 8387857] log opened at 2016-08-20 12:40:30
test mock --entity ~.domain.Centro
test mock --entity ~.domain.Demandante
test mock --entity ~.domain.Empresa
test mock --entity ~.domain.Experiencia
test mock --entity ~.domain.Formacion
test mock --entity ~.domain.Formaciondemandante
test mock --entity ~.domain.Inscripcion
test mock --entity ~.domain.Oferta
test mock --entity ~.domain.Puesto
test mock --entity ~.domain.Sede
test mock --entity ~.domain.Tipoformacion
test mock --entity ~.domain.Usuario
test integration --entity ~.domain.Centro
test integration --entity ~.domain.Demandante
test integration --entity ~.domain.Empresa
test integration --entity ~.domain.Experiencia
test integration --entity ~.domain.Formacion
test integration --entity ~.domain.Formaciondemandante
test integration --entity ~.domain.Inscripcion
test integration --entity ~.domain.Oferta
test integration --entity ~.domain.Puesto
test integration --entity ~.domain.Sede
test integration --entity ~.domain.Tipoformacion
test integration --entity ~.domain.Usuario
selenium test --controller ~.web.CentroController
selenium test --controller ~.web.DemandanteController
selenium test --controller ~.web.EmpresaController

```
selenium test --controller ~.web.ExperienciaController
```

```
selenium test --controller ~.web.FormacionController
```

```
selenium test --controller ~.web.FormaciondemandanteController
```

```
selenium test --controller ~.web.InscripcionController
```

```
selenium test --controller ~.web.OfertaController
```

```
selenium test --controller ~.web.PuestoController
```

```
selenium test --controller ~.web.SedeController
```

```
selenium test --controller ~.web.TipoformacionController
```

```
selenium test --controller ~.web.UsuarioController
```

```
// Spring Roo 1.3.2.RELEASE [rev 8387857] log closed at 2016-08-26 22:09:40
```