**Alin Giurca, Gonzalo Baz, Roberto García-Gasco e Ignacio Llorca**

GRUPO T1

TG2. WEB TESTING TOOLS: FUNCTIONAL TESTING

Desarrollo de tecnologías emergentes

Índice

[1. Autores del trabajo, planificación y entrega 2](#_Toc478570676)

[1.1 Autores 2](#_Toc478570677)

[1.2 Planificación 3](#_Toc478570678)

[1.3 Entrega 4](#_Toc478570679)

[2. Descripción de las tecnologías 4](#_Toc478570680)

[2.1 Descripción de la tecnología 1 4](#_Toc478570681)

[2.2 Descripción de la tecnología 2 5](#_Toc478570682)

[3. Criterios de comparación 6](#_Toc478570683)

[3.1 Categoría A: Nombre 6](#_Toc478570684)

[3.1.1 Criterio A.1: Nombre 6](#_Toc478570685)

[3.1.2 Criterio A.2: Nombre 7](#_Toc478570686)

[3.1.n Criterio A.n: Nombre 7](#_Toc478570687)

[3.2 Categoría B: Nombre 7](#_Toc478570688)

[3.2.1 Criterio B.1: Nombre 7](#_Toc478570689)

[3.2.2 Criterio B.2: Nombre 7](#_Toc478570690)

[3.2.n Criterio B.n: Nombre 7](#_Toc478570691)

[3.3 Categoría Z: Nombre 7](#_Toc478570692)

[3.3.1 Criterio Z.1: Nombre 7](#_Toc478570693)

[3.3.2 Criterio Z.2: Nombre 7](#_Toc478570694)

[3.3.n Criterio Z.n: Nombre 7](#_Toc478570695)

[4. Evaluación de los criterios por tecnología 8](#_Toc478570696)

[4.1 Evaluación de los criterios para la tecnología 1 8](#_Toc478570697)

[4.2 Evaluación de los criterios para la tecnología 2 8](#_Toc478570698)

[5. Comparación de las tecnologías 9](#_Toc478570699)

[6. Recomendaciones 11](#_Toc478570700)

[6.1 Situación 1 11](#_Toc478570701)

[6.1.1 Descripción de la situación 11](#_Toc478570702)

[6.1.2 Recomendación de tecnología a utilizar 11](#_Toc478570703)

[6.2 Situación 2 11](#_Toc478570704)

[6.2.1 Descripción de la situación 11](#_Toc478570705)

[6.2.2 Recomendación de tecnología a utilizar 11](#_Toc478570706)

[7. Encuesta 12](#_Toc478570707)

[8. Conclusiones 12](#_Toc478570708)

# 1. Autores del trabajo, planificación y entrega

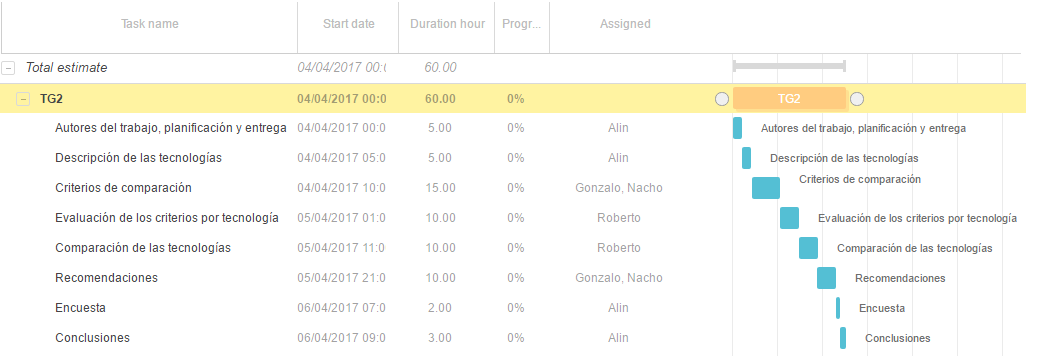
## 1.1 Autores

El trabajo lo ha realizado el grupo T1, formado por:

* Alin Nicolae Giurca
* Roberto García-Gasco Aparicio
* Gonzalo Baz Cifrián
* Ignacio Llorca Rodríguez

## 1.2 Planificación

A continuación, se adjunta una imagen1 del diagrama de Gantt diseñado para planificar el trabajo y la consecución del proyecto:



1: se adjunta una captura de pantalla y no un link al proyecto compartido porque la opción de compartir/exportar ha dejado de estar disponible en la herramienta con la que se ha realizado el diseño, GanttPro.

## 1.3 Entrega

El proyecto al completo con todos sus archivos y versiones se encuentra en el siguiente repositorio de GitHub:

<https://github.com/AlinGiurca/TG2>

# 2. Descripción de las tecnologías

En los siguientes apartados se debe describir brevemente cada tecnología a comparar.

## 2.1 Descripción de la tecnología 1

**Telerik Test Studio.**



Es una aplicación visual que no requiere una experiencia ni conocimiento alto sobre ella. Esta aplicación puede realizar más de un tipo de testing, entre los que se encuentra el testing funcional. El funcionamiento básico consiste en introducir la URL de la aplicación o página web que se va a testear y grabar todos los movimientos que el usuario realiza o el test en cuestión pide. Cuando ha grabado tales movimientos, los almacena para su posterior utilización o modificación.

Una vez que tenemos almacenados y modificados los movimientos que queremos que realice, el programa realiza los mismos movimientos 1 o n veces para verificar que realmente funciona ese test.

Por último, la aplicación guarda toda la información resultante del test en un archivo, ya sea válido o no dicho test.

## 2.2 Descripción de la tecnología 2

**Selenium**



Es una librería del lenguaje de programación Python. Con Selenium se puede realizar functional testing en modo desarrollador, es decir, el propio desarrollador debe programar los movimientos que se van a realizar en la aplicación o página web. Esta librería proporciona numerosos métodos y funciones que permiten al usuario obtener la máxima información sobre la página/aplicación, y con dicha información, se pueden crear los test funcionales.

El funcionamiento no varía demasiado del programa Telerik Test Studio que se ha nombrado antes, sin embargo, proporciona muchísimas más opciones de testing al ser un método a más bajo nivel.

Permite realizar el test de todas las opciones que la aplicación/página dispone, obtener la información, usarla, modificarla y almacenar los resultados en documentos del tipo CSV, por ejemplo.

# 3. Criterios de comparación

En los sub-apartados de este apartado se deben indicar cada uno de los criterios (también llamados factores, propiedades, características, indicadores, etc.), que se usarán en la comparación.

Los criterios deben organizarse en categorías (al menos 3 categorías). El número de criterios totales no puede ser inferior a 20.

Las categorías dependerán del tipo de tecnología, pueden ser categorías como “General”, “Utilidades”, “Rendimiento”, etc.

Los criterios a definir en cada categoría también dependerán del tipo de tecnologías a comparar. En el siguiente apartado hay algunos ejemplos.

## 3.1 Categoría A: Nombre

### 3.1.1 Criterio A.1: Nombre

Por cada criterio hay que indicar el nombre, una breve descripción, y el tipo de valor a asignar al criterio. El número de criterios no puede ser inferior a 20.

Por ejemplo, si se comparan dos herramientas CASE, un criterio podría ser el autor, entonces en este apartado se podría incluir este texto:

*3.1.1 Criterio A.1: Autor de la herramienta*

*Nombre del criterio: Autor.*

*Descripción: Nombre de la persona, institución o empresa que ha creado la herramienta.*

*Tipo de valor: Texto libre.*

Otro ejemplo de criterio con valor Si/No:

*3.1.1 Criterio A.1: Generación de código Java*

*Nombre del criterio: Generación código Java.*

*Descripción: Indica si la herramienta CASE incluye funcionalidad para generar archivos fuente .java a partir de diagramas de clase.*

*Tipo de valor: Booleano (Si/No).*

Otro ejemplo de criterio con valor numérico:

*3.1.1 Criterio A.1: Número de proyectos simultáneos*

*Nombre del criterio: Proyectos simultáneos.*

*Descripción: Indica el número máximo de proyectos de modelado que deja tener abiertos la herramienta de forma simultánea.*

*Tipo de valor: Numérico.*

### 3.1.2 Criterio A.2: Nombre

### 3.1.n Criterio A.n: Nombre

## 3.2 Categoría B: Nombre

### 3.2.1 Criterio B.1: Nombre

### 3.2.2 Criterio B.2: Nombre

### 3.2.n Criterio B.n: Nombre

## 3.3 Categoría Z: Nombre

### 3.3.1 Criterio Z.1: Nombre

### 3.3.2 Criterio Z.2: Nombre

### 3.3.n Criterio Z.n: Nombre

# 4. Evaluación de los criterios por tecnología

## 4.1 Evaluación de los criterios para la tecnología 1

Debe incluir al menos una tabla con la siguiente estructura.

|  |  |
| --- | --- |
| CRITERIOS | EVALUACIÓN |
| Criterio A.1: Nombre |  |
| Criterio A.2: Nombre |  |
| Criterio A.n: Nombre |  |
| … |  |
| Criterio B.1: Nombre |  |
| Criterio B.2: Nombre |  |
| … |  |

Y algunos comentarios aclaratorios sobre aquellos criterios cuyo valor indicado en la tabla no sea suficiente para entenderlo.

La tabla anterior es obligatoria y deben completarla los autores del trabajo, aunque se pueden incluir otros gráficos o tablas complementarias copiadas y pegadas desde diversas fuentes de información, siempre que debajo de cada uno se indique la fuente (al menos la URL).

## 4.2 Evaluación de los criterios para la tecnología 2

# 5. Comparación de las tecnologías

Debe incluir al menos una tabla resumen, en sección de página horizontal, cruzando los criterios y los valores de cada tecnología. Con una columna de comentarios sobre la comparación

Esta tabla anterior es obligatoria y deben completarla los autores del trabajo, aunque se pueden incluir otros gráficos o tablas complementarias copiadas y pegadas desde diversas fuentes de información, siempre que debajo de cada uno se indique la fuente (al menos la URL).

|  |  |  |  |
| --- | --- | --- | --- |
| CRITERIOS | TECNOLOGÍA A | TECNOLOGÍA B | COMENTARIOS |
| A.1 |  |  |  |
| A.2 |  |  |  |
| … |  |  |  |
| B.1 |  |  |  |
| B.2 |  |  |  |
| … |  |  |  |

# 6. Recomendaciones

Deben platearse posibles situaciones de uso, y recomendar justificadamente una u otra tecnología en función de la situación. Al menos 2 situaciones diferentes.

## 6.1 Situación 1

### 6.1.1 Descripción de la situación

*Una posible situación en el caso de comparar dos herramientas CASE, podría ser el caso de una empresa de desarrollo muy interesada en tecnologías open source, que programa sólo en Java, con equipos de desarrollo pequeños, que utiliza UML como notación, etc, etc*…

### 6.1.2 Recomendación de tecnología a utilizar

Debe indicarse la tecnología propuesta para esa situación.

Debe incluirse una tabla como la siguiente, mostrando las ventajas, respecto a los criterios, que ofrece cada tecnología en esa situación concreta.

Incluir sólo los criterios sobre los que se aprecien ventajas de una de las tecnologías frente a otra. No incluir criterios que no sean relevantes para la decisión (por ejemplo, el criterio “autor” seguramente no será relevante).

|  |  |  |
| --- | --- | --- |
| Criterios relevantes para la decisión | Ventajas tecnología 1 | Ventajas tecnología 2 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 6.2 Situación 2

### 6.2.1 Descripción de la situación

### 6.2.2 Recomendación de tecnología a utilizar

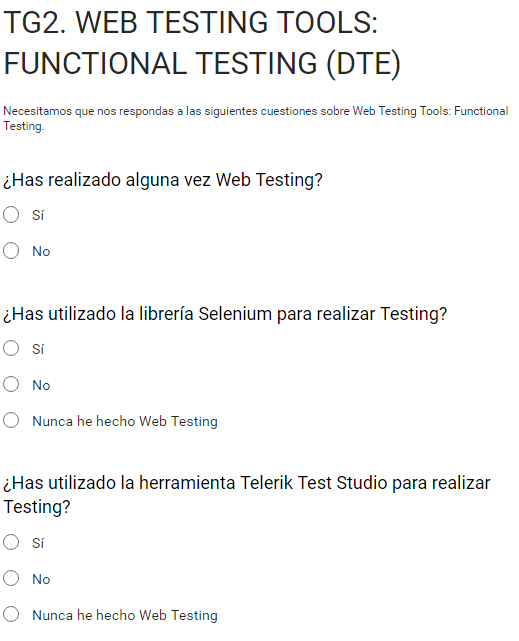
---------------------------

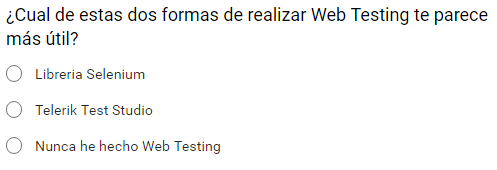
(Hay que cumplir la estructura básica indicada de secciones. Pero si se desea se pueden añadir otras secciones como anexos. Por ejemplo, alguna encuesta de opinión realizada sobre las tecnologías, etc.)

# 7. Encuesta

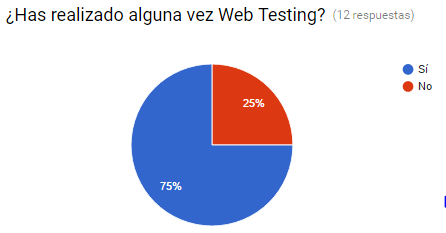
Hemos realizado una encuesta mediante los formularios de Google para ver cuanta gente realiza o ha realizado alguna vez WEB TESTING: FUNCTIONAL TESTING. Además, queremos realizar un análisis sobre cuál de las dos herramientas que estamos investigando son más utilizadas en nuestro entorno y cuál es la que consideran más útil para realizar Testing.

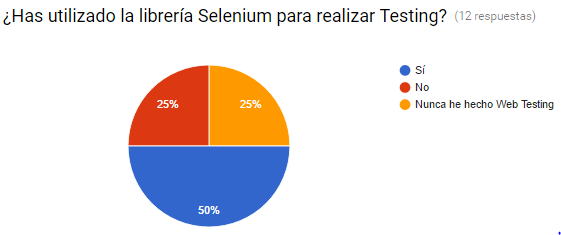
El test es el siguiente (<https://goo.gl/forms/OBF9AWVi0PRd3qOa2>):

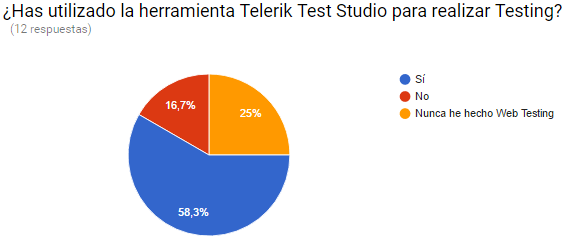


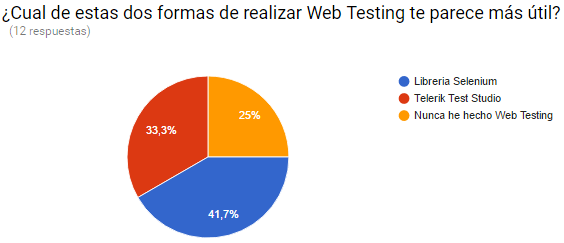


Los resultados obtenidos son los siguientes:









De lo resultados que hemos obtenido en esta encuesta podemos sacar diferentes conclusiones acerca de la misma:

En primer lugar, un alto porcentaje de las personas que han hecho la encuesta han realizado alguna vez algún tipo de web testing. Esto es debido a que la encuesta ha sido lanzada a compañeros de nuestro entorno y a miembros de cátedras de investigación de la Universidad de Alcalá.

En segundo lugar, preguntamos a los encuestados que cuántos de ellos habían utilizado las herramientas Telerik Test Studio y Selenium. Para nuestra sorpresa la mayoría de las personas habían utilizado más Telerik Test Studio que Selenium. Esto creemos que es debido a que con Telerik el testing es mucho más intuitivo y mucho más rápido de realizar que con Selenium, debido a que esta última tiene más complejidad y muchos de los encuestados habían realizado pequeños test para verificar el adecuado funcionamiento de sus aplicaciones.

Finalmente, preguntamos cuál de las dos herramientas les resultaba más útil para realizar testing. La respuesta fue la esperada, Selenium les resulta más útil que Telerik. Pensamos que esto es debido a que algunos de los encuestados tenían más experiencia a la hora de realizar aplicaciones y le daban mucha más importancia al testing más invasivo, en cambio, otras personas que habían implementado aplicaciones mucho menos complejas que no requerían un testing tan sofisticado.

# 8. Conclusiones

Una vez hemos realizado una comparación entre ambas herramientas podemos concluir que son dos herramientas que realizan la misma tarea, pero de forma diferente. Telerik Test Studio tiene una interfaz que hace muy sencillo la realización de test básicos y Selenium requiere de un alto entendimiento del lenguaje de programación Python.

Si hacemos un resumen sobre los criterios de comparación que hemos analizado en los apartados anteriores podemos observar que Selenium es una librería gratuita de Python, Open Source, sin mantenimiento, mucho más desarrollada, sin interfaz gráfica y permite hacer test mucho más específicos, pero exige que el usuario tenga altos conocimientos del lenguaje de programación. En cambio, Telerik es una herramienta de pago, con mantenimiento, menos desarrollada, con interfaz gráfica y permite hacer cualquier tipo test sin poseer grandes habilidades a la hora de programar.

Si tuviéramos que recomendar estas dos herramientas, propondríamos Telerik a un usuario que vaya a realizar test básicos y muy superficiales, y Selenium a aplicaciones que requieran test profundos y complejos.