**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 3](#_Toc478573658)

[1.1 Autores 3](#_Toc478573659)

[1.2 Planificación 4](#_Toc478573660)

[1.3 Entrega 5](#_Toc478573661)

[2. Descripción de las tecnologías 5](#_Toc478573662)

[2.1 Descripción de la tecnología 1 5](#_Toc478573663)

[2.2 Descripción de la tecnología 2 6](#_Toc478573664)

[3. Criterios de comparación 7](#_Toc478573665)

[3.1 Categoría A: Generales 7](#_Toc478573666)

[3.1.1 Criterio A.1: Precio 7](#_Toc478573667)

[3.1.2 Criterio A.2: Tipo de licencia 7](#_Toc478573668)

[3.1.3 Criterio A.3: Popularidad 7](#_Toc478573669)

[3.1.4 Criterio A.3: Soporte y mantenimiento 7](#_Toc478573670)

[3.1.5 Criterio A.3: Frecuencia de las actualizaciones 7](#_Toc478573671)

[3.1.6 Criterio A.3: Exigencias técnicas. 7](#_Toc478573672)

[3.2 Categoría B: Software 7](#_Toc478573673)

[3.2.1 Criterio B.1: Multiplataforma. 7](#_Toc478573674)

[3.2.2 Criterio B.2: Robustez 7](#_Toc478573675)

[3.2.3 Criterio B.3: Madurez 7](#_Toc478573676)

[3.2.4 Criterio B.4: Multilenguaje 8](#_Toc478573677)

[3.2.5 Criterio B.5: Escalabilidad 8](#_Toc478573678)

[3.2.6 Criterio B.6: Profundidad de análisis 8](#_Toc478573679)

[3.2.7 Criterio B.7: Migración de datos 8](#_Toc478573680)

[3.2.8 Criterio B.8: Seguridad de la herramienta. 8](#_Toc478573681)

[3.2.9 Criterio B.9: Trabajo colaborativo 8](#_Toc478573682)

[3.3 Categoría C: Facilidad de uso 8](#_Toc478573683)

[3.3.1 Criterio C.1: Usabilidad 8](#_Toc478573684)

[3.3.2 Criterio C.2: Interfaz 8](#_Toc478573685)

[3.3.3 Criterio C.3: Curva de aprendizaje 9](#_Toc478573686)

[3.3.4 Criterio C.4: Documentación 9](#_Toc478573687)

[4. Evaluación de los criterios por tecnología 10](#_Toc478573688)

[4.1 Evaluación de los criterios para la tecnología 1 10](#_Toc478573689)

[4.2 Evaluación de los criterios para la tecnología 2 10](#_Toc478573690)

[5. Comparación de las tecnologías 11](#_Toc478573691)

[6. Recomendaciones 13](#_Toc478573692)

[6.1 Situación 1 13](#_Toc478573693)

[6.1.1 Descripción de la situación 13](#_Toc478573694)

[6.1.2 Recomendación de tecnología a utilizar 13](#_Toc478573695)

[6.2 Situación 2 13](#_Toc478573696)

[6.2.1 Descripción de la situación 13](#_Toc478573697)

[6.2.2 Recomendación de tecnología a utilizar 13](#_Toc478573698)

[7. Encuesta 15](#_Toc478573699)

[8. Conclusiones 16](#_Toc478573700)

# 1. Autores del trabajo, planificación y entrega

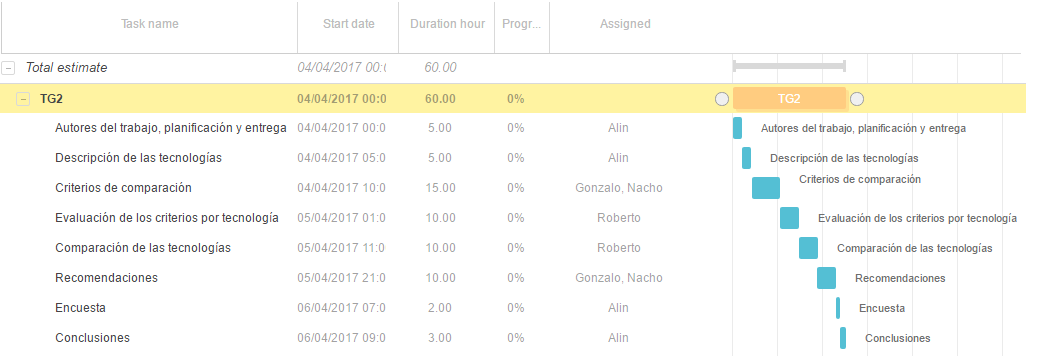
## 1.1 Autores

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

## 3.1 Categoría A: Generales

### 3.1.1 Criterio A.1: Precio

Descripción: coste del software.

Tipo de valor: euros.

### 3.1.2 Criterio A.2: Tipo de licencia

Descripción: tipo de licencia del software (freeware, open source, cerrado…)

Tipo: texto libre.

### 3.1.3 Criterio A.3: Popularidad

Descripción: número y reputación de los usuarios, respaldo de la comunidad.

Tipo: texto libre.

### 3.1.4 Criterio A.3: Soporte y mantenimiento

Descripción: calidad y tipo de soporte y mantenimiento ofrecido por la desarrolladora.

Tipo: texto libre.

### 3.1.5 Criterio A.3: Frecuencia de las actualizaciones

Descripción: frecuencia con la que liberan nuevas versiones del paquete software.

Tipo: escala de tiempo.

### 3.1.6 Criterio A.3: Exigencias técnicas.

Descripción: popularidad.

Tipo: texto libre.

## 3.2 Categoría B: Software

### 3.2.1 Criterio B.1: Multiplataforma.

Descripción: posibilidad de utilizar la herramienta en diferentes sistemas operativos.

Tipo de valor: Booleano (Si/No).

### 3.2.2 Criterio B.2: Robustez

Descripción: capacidad de reacción del software frente a situaciones excepcionales.

Tipo de valor: Texto libre.

### 3.2.3 Criterio B.3: Madurez

Descripción: tiempo de vida del software.

Tipo de valor: Texto libre.

### 3.2.4 Criterio B.4: Multilenguaje

Descripción: permite la implementación en diferentes lenguajes de programación.

Tipo de valor: Booleano (Si/No).

### 3.2.5 Criterio B.5: Escalabilidad

Descripción: habilidad de reaccionar y adaptarse a cambios de tamaño sin perder calidad.

Tipo de valor: Texto libre.

### 3.2.6 Criterio B.6: Profundidad de análisis

Descripción: complejidad de los análisis que permite realizar.

Tipo de valor: Texto libre.

### 3.2.7 Criterio B.7: Migración de datos

Descripción: capacidad de cambiar el trabajo a otro software correctamente y sin pérdida de información.

Tipo de valor: Texto libre.

### 3.2.8 Criterio B.8: Seguridad de la herramienta.

Descripción: protección frente a la pérdida de datos.

Tipo de valor: Texto libre

### 3.2.9 Criterio B.9: Trabajo colaborativo

Descripción: posibilidad de trabajar más de un usuario sobre un mismo script simultáneamente.

Tipo de valor: Booleano (Si/No).

## 3.3 Categoría C: Facilidad de uso

### 3.3.1 Criterio C.1: Usabilidad

Descripción: facilidad de uso y experiencia requerida por el usuario para la correcta manipulación del programa.

Tipo: texto libre.

### 3.3.2 Criterio C.2: Interfaz

Descripción: nivel de intuitividad de la interfaz de usuario.

Tipo: texto libre.

### 3.3.3 Criterio C.3: Curva de aprendizaje

Descripción: tiempo y esfuerzo requerido para aprender a aprovechar el software partiendo de cero.

Tipo: texto libre.

### 3.3.4 Criterio C.4: Documentación

* C4.1 Documentación Oficial

Descripción: calidad y cantidad de documentación ofrecida por la desarrolladora del software.

Tipo: texto abierto.

* C4.2 Documentación no oficial

Descripción: calidad y cantidad de documentación existente en internet (guías, tutoriales, cursos, etc.).

# 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

## 6.1 Situación 1

### Descripción de la situación

Soy un estudiante de periodismo. Estoy realizando una página web para crear, compartir y discutir contenidos con mis colegas y necesito testear que funciona correctamente la pantalla de Registro antes de lanzarla. Solo conozco programación web HTML, javascript, etc. a nivel amateur y nunca he trabajado con otros lenguajes.

### 6.1.2 Recomendación de tecnología a utilizar

En este caso no cabe lugar a dudas. Sin conocimientos de Python utilizar Selenium requeriría mucho tiempo de formación y pasar por una curva de aprendizaje muy pronunciada. Telerik Studio, dada su interfaz visual y su usabilidad sería la opción correcta para el estudiante. De este modo, pese a no permitirle realizar análisis profundos (que seguramente no los necesite), le ahorrará tiempo y cubrirá completamente sus expectativas.

|  |  |  |
| --- | --- | --- |
| Criterios relevantes para la decisión | Ventajas Telerik Test Studio | Ventajas Selenium |
| Precio | El free trial de 30 días será suficiente para el estudiante. | Selenium sería mejor opción puesto que es Open Source y tendría costo cero. |
| Curva de aprendizaje | Mucho más suave que la de Selenium. | Requerirá al estudiante aprender a programar y usar librerías de Python. |
| Usabilidad | Es más sencillo de utilizar que Selenium. | Requiere más experiencia y es más complicado sacarle partido. |
| Interfaz | Dashboard más intuitivo que un panel de programación común. | Mucho menos intiuitivo que Telerik Test Studio. |

## 6.2 Situación 2

### 6.2.1 Descripción de la situación

Somos un grupo de desarrolladores experimentados trabajando en una aplicación móvil, la cual está en fase final y necesitamos someterla a pruebas, para ello hemos estado investigando diferentes herramientas buscando una que nos ofrezca una gran profundidad a la hora de aplicar dichas pruebas sobre el software.

### 6.2.2 Recomendación de tecnología a utilizar

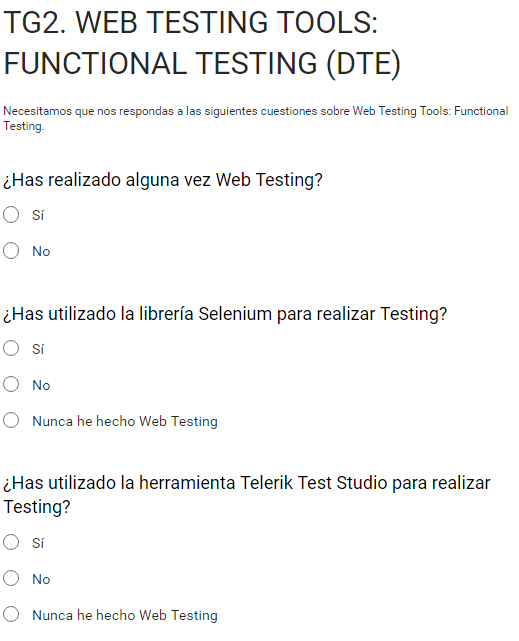
Puesto que el usuario es experimentado en el ámbito de la programación, y busca realizar un test profundo y complejo, la recomendación definitiva es el uso de la librería Selenium de Python.

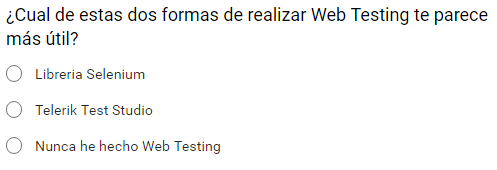
|  |  |  |
| --- | --- | --- |
| Criterios relevantes para la decisión | Ventajas Telerik Test Studio | Ventajas Selenium |
| Usabilidad | Bastante alta, mayormente para un usuario novel. | Complicada para un principiante pero buena para un usuario experimentado. |
| Profundidad de análisis | Superficial, permite hacer test sencillos. | Grande, permite realizar test más profundos y complejos que Telerik. |
| Interfaz | Sencilla, más visual e intuitiva que Selenium | Compleja, puesto que es en entorno de desarrollo, no supondría problema para un desarrollador experto. |
| Escalabilidad | Disminuida, pensada para grupos más pequeños. | Buena, adecuada para un grupo más grande con posibilidad de desarrollar una aplicación grande. |

# 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:

(Faltan resultados)

# 8. Conclusiones