Contenido

[1. Autores del trabajo, planificación y entrega 2](#_Toc448254544)

[1.1 Autores 2](#_Toc448254545)

[1.2 Planificación 2](#_Toc448254546)

[1.3 Entrega 2](#_Toc448254547)

[2. Requisitos del prototipo a implementar 3](#_Toc448254548)

[2.1 Requisitos funcionales 3](#_Toc448254549)

[2.2 Otros requisitos 3](#_Toc448254550)

[3. Criterios de comparación en la implementación 4](#_Toc448254551)

[3.1 Criterio 1: Nombre del criterio 4](#_Toc448254552)

[3.2 Criterio 2: Nombre del criterio 4](#_Toc448254553)

[3.N Criterio N: Nombre del criterio 4](#_Toc448254554)

[4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología A 5](#_Toc448254555)

[4.1 Documentación de diseño 5](#_Toc448254556)

[4.2 Documentación de construcción 5](#_Toc448254557)

[4.3 Documentación de pruebas 5](#_Toc448254558)

[4.4 Documentación de instalación 5](#_Toc448254559)

[4.5 Manual de usuario 5](#_Toc448254560)

[5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B 6](#_Toc448254561)

[5.1 Documentación de diseño 6](#_Toc448254562)

[5.2 Documentación de construcción 6](#_Toc448254563)

[5.3 Documentación de pruebas 6](#_Toc448254564)

[5.4 Documentación de instalación 6](#_Toc448254565)

[5.5 Manual de usuario 6](#_Toc448254566)

[6. Comparación de las dos implementaciones 7](#_Toc448254567)

[6.1 Evaluación de los criterios en la implementación usando la tecnología A 7](#_Toc448254568)

[6.2 Evaluación de los criterios en la implementación usando la tecnología B 7](#_Toc448254569)

[7. Comparación de la implementación de las tecnologías 8](#_Toc448254570)

[8. Conclusiones 10](#_Toc448254571)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

En este apartado se debe indicar el número de grupo y los nombres de los autores, poniendo en primer lugar al coordinador del grupo.

## 1.2 Planificación

En este apartado se debe incluir un enlace (URL) compartido a la planificación del trabajo utilizando una herramienta online de diagramación Gantt (por ejemplo, GanttPro, versión gratuita).

Hay que tener en cuenta que cada participante del grupo debe tener asignadas tareas que sumen al menos 45 horas. El peso de este trabajo en la calificación total de la asignatura es de un 30%, por tanto requiere de una dedicación de 45 horas del total de 150 horas de la asignatura.

## 1.3 Entrega

En este apartado debe incluirse un enlace (URL) a un repositorio en GitHub o en BitBucket creado para el trabajo.

En dicho repositorio debe encontrarse, al menos los siguientes archivos en la rama máster:

* Informe del trabajo: con el nombre TG3\_final.docx
* Presentación del trabajo: TG3\_final.pptx
* Prototipos obtenidos implementando cada una de las tecnologías (deben incluir el código fuente y todos los archivos necesarios para la instalación y uso de cada prototipo):
  + PrototipoTecnologiaA\_final.zip (o .rar)
  + PrototipoTecnologiaB\_final.zip (o .rar).

Dichos archivos serán los que se tendrán en cuenta para la calificación del trabajo.

# 2. Requisitos del prototipo a implementar

El objetivo del proyecto es comparar la implementación de un mismo prototipo de sistema utilizando dos tecnologías diferentes (A y B).

Es importante cumplimentar este apartado antes de empezar a implementar el prototipo de cada tecnología, porque ambos prototipos deben cumplir los requisitos que se establezcan en este apartado. Si se van a crear dos equipos de trabajo, uno para cada prototipo, el contenido de este apartado es lo que han de compartir ambos equipos como punto de partida.

Cuanto más detallados sean los requisitos, mayor será la precisión en la comparación que se realizará al final del trabajo. Se trata de conseguir dos prototipos con igual funcionalidad, pero utilizando diferentes tecnologías.

Se puede dar libertad a los equipos de desarrollo en cuanto al diseño, pero la funcionalidad debe ser lo más parecida posible. Por ejemplo, no es necesario que los colores utilizados en las pantallas sean exactamente los mismos en ambos prototipos, a no ser que los miembros del grupo lo hayan decidido así, en cuyo caso, esos detalles de colores deben incluirse en el catálogo de requisitos, para que ambos equipos los cumplan.

## 2.1 Requisitos funcionales

Los requisitos funcionales deben ser los mismos para las dos implementaciones.

En la siguiente tabla se indicará el catálogo de requisitos funcionales del sistema.

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| RF01 | El sistema deberá comprobar la pantalla de registro de Facebook |
| RF02 | El sistema deberá intentar saturar el registro de Facebook |
| RF03 | El sistema deberá comprobar los mensajes de la página al introducir datos erróneos en los campos del registro |
| RF04 | El sistema no debería abortar, aunque el registro sea incorrecto |
| RF05 | El sistema debe mostrar gráficamente como se lleva a cabo el proceso. |
| RF06 |  |
|  |  |

## 2.2 Otros requisitos

Se pueden incluir aquí otros requisitos para el prototipo que no puedan considerarse como funcionales. Por ejemplo, requisitos de datos, de seguridad, de interfaz de usuario, de rendimientos, etc.

Se puede dejar libertad

En la siguiente tabla se indicará el catálogo de requisitos no funcionales del sistema.

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| R01 | El sistema debe ser inalterable por agentes externos. |
| R02 | El sistema debe tardar menos de 10 segundos en llevar a cabo el registro. |
|  |  |

# 3. Criterios de comparación en la implementación

En el trabajo TG2 se definieron criterios de comparación de las dos tecnologías a nivel teórico.

En este trabajo hay que definir criterios para la comparación de la implementación de las tecnologías en la construcción del prototipo de sistema de ejemplo, cuyos requisitos son los establecidos en el apartado 2.

Se trata de criterios del tipo” “horas empleadas en el desarrollo del sistema”, “velocidad de funcionamiento del sistema”, “recursos necesarios”, etc.

## 3.1 Criterio 1: Tiempo necesario para desarrollar el sistema

Descripción: tiempo empleado en tener el sistema listo para comprobar el registro de Facebook.

Tipo de valor: horas.

## 3.2 Criterio 2: Velocidad del test

Descripción: tiempo que tarda el sistema en analizar eficientemente el registro de Facebook.

Tipo de valor: segundos.

## 3.3 Criterio 3: Recursos necesarios

Descripción: recursos hardware, software y web necesarios para realizar eficientemente el testeo.

Tipo de valor: texto.

## 3.4 Criterio 4: Problemas durante el desarrollo

Descripción: número de problemas encontrados durante el desarrollo del sistema de testeo.

Tipo de valor: texto.

## 3.5 Criterio 5: Experiencia requerida

Descripción: cantidad de experiencia requerida para realizar el desarrollo correctamente.

Tipo de valor: texto.

## 3.N Criterio N: Nombre del criterio

Descripción:

Tipo de valor:

# 4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología A

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología A, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

## 4.1 Documentación de diseño

Hay que incluir la descripción del diseño del prototipo, incluyendo diagramas, y el diseño de la interfaz de usuario.

## 4.2 Documentación de construcción

Hay que incluir una descripción de la construcción del prototipo, incluyendo algún extracto de código fuente. No es necesario todo el código. Sólo algún extracto para ver cómo se ha comentado.

## 4.3 Documentación de pruebas

Casos de prueba establecidos y resultados de las pruebas y acciones de corrección. No es creíble que no hayan aparecido errores en los caso de prueba.

## 4.4 Documentación de instalación

Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.

## 4.5 Manual de usuario

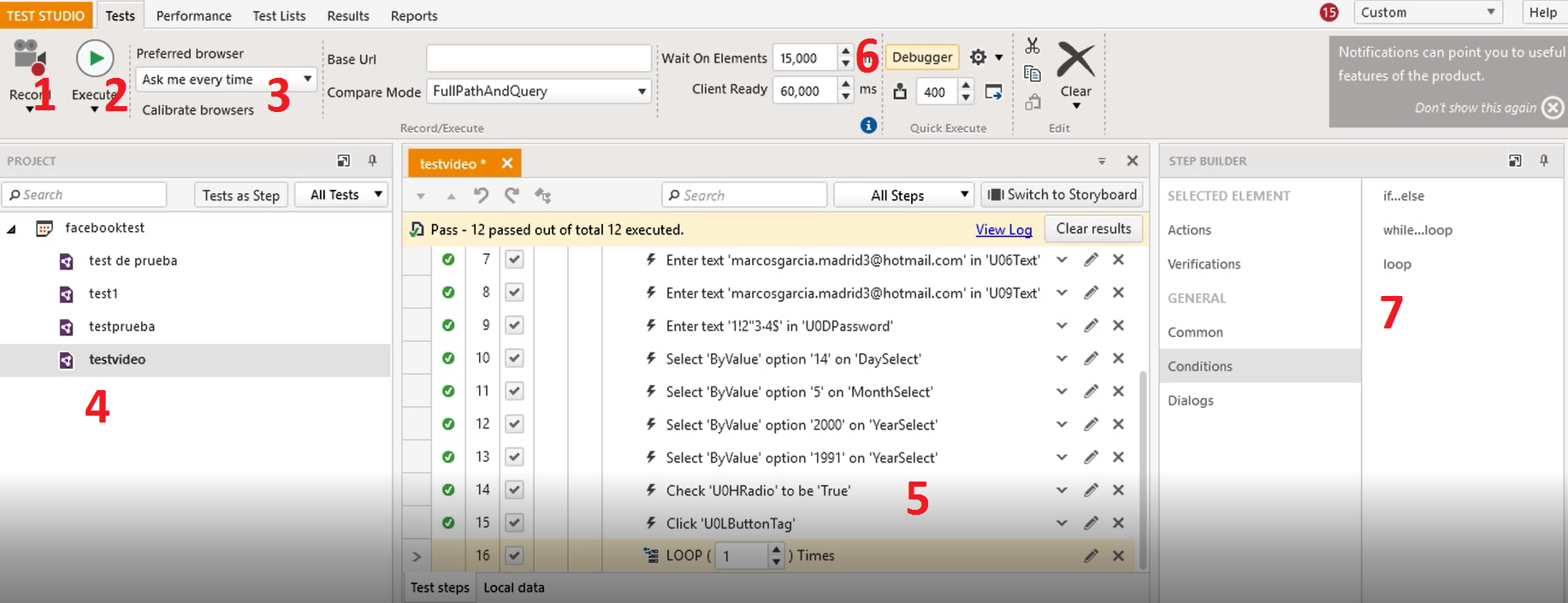
Descripción suficiente para que una persona que no ha participado en el proyecto pueda utilizar toda la funcionalidad que ofrece el prototipo. Que debe coincidir con los requisitos funcionales incluidos en el apartado 2.

# 5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B

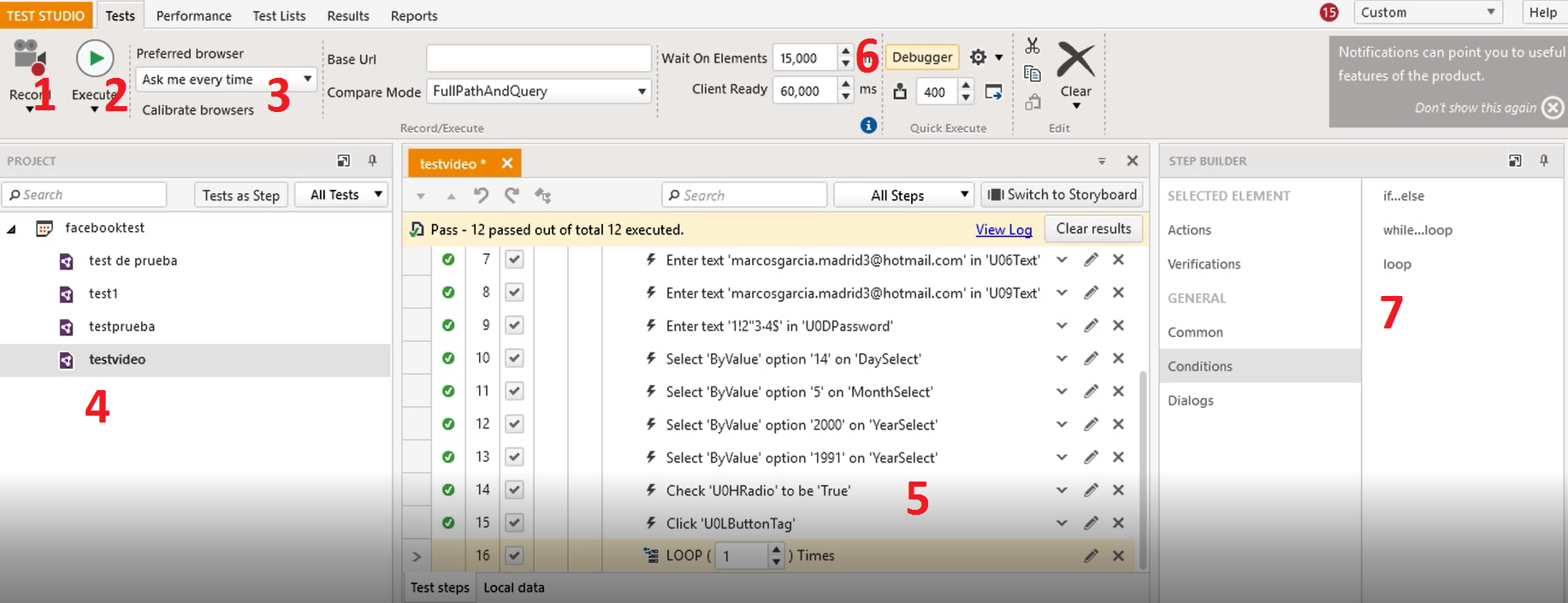
Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología B, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

## 5.1 Documentación de diseño

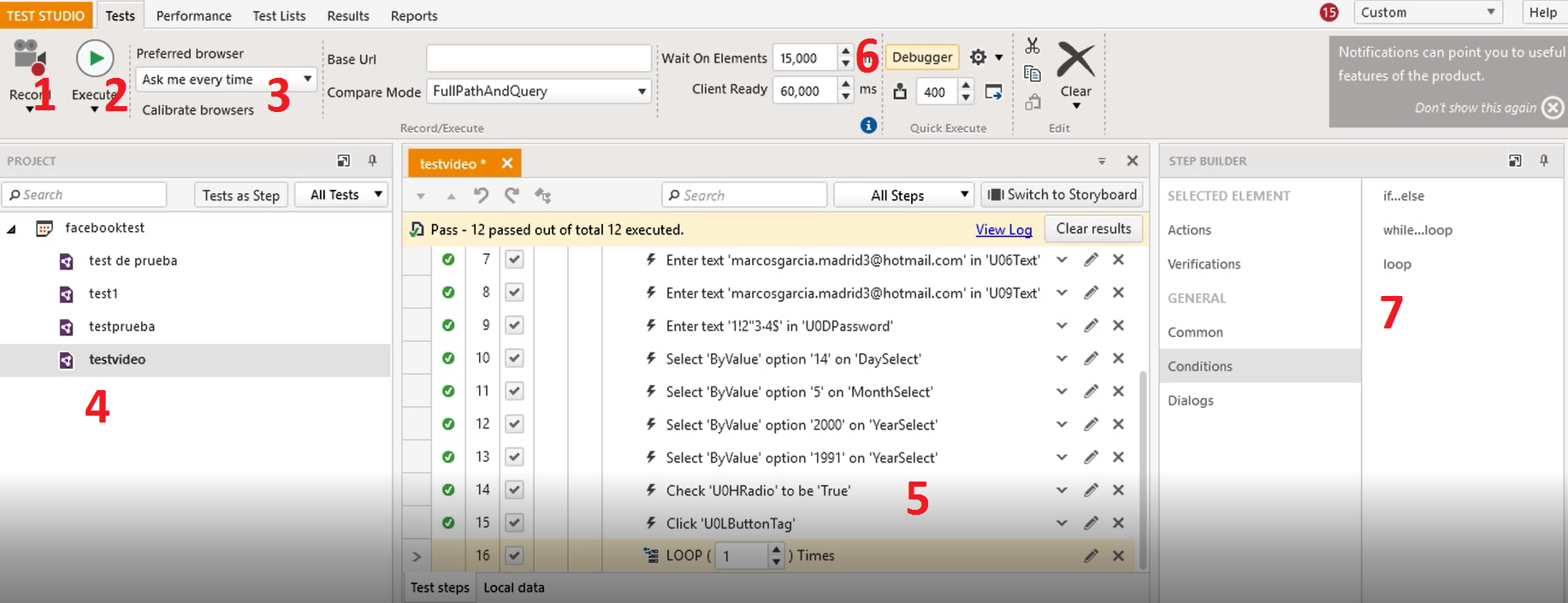
La interfaz principal de Telerik tiene el siguiente diseño:



Tal y como se ha explicado en ocasiones anteriores, una vez que se ha seleccionado el sitio web o aplicación sobre el que se va a realizar el test, Telerik funciona grabando todos los movimientos que se realizan para repetirlos después. A continuación pasamos a explicar las funciones principales e imprescindibles para realizar un test.



1. Con el botón Record grabará nuestros movimientos (clicks, entrada por teclado, etc.) hasta que seleccionemos Stop.
2. Con el botón Execute ejecutará todos los movimientos grabados previamente.
3. En el apartado Preferred Browser nos permite elegir con qué navegador queremos abrir la web sobre la que se va a realizar el test.
4. El panel Project muestra todos los tests que se han guardado y permite su clasificación por carpetas y categorías.



1. Es el panel principal de Telerik Test Studio. En él, se muestran todas las acciones que han sido grabadas. Por ejemplo, en este caso, se puede ver como la séptima acción ha sido introducir el texto [marcosgarcia.madrid3@hotmail.com](mailto:marcosgarcia.madrid3@hotmail.com) en el campo de nombre ‘U06Text’ de la web. En este panel se nos mostrarán las acciones que se han podido llevar a cabo y las que no en forma de ticks verdes o cruces rojas a la izquierda de cada acción. Además, podremos deseleccionar las acciones que han sido grabadas y no queremos que se lleven a cabo de forma que si deseleccionamos la acción 8 dejará de introducir el correo en el campo ‘U09Text’ y pasará directamente a la acción 9, escribir la contraseña en el campo ‘U0DPassword’. Del mismo modo, podemos cambiar las acciones haciendo click en el lápiz a la derecha cada una. En este ejemplo podríamos editar la acción siete para hacer que introduzca un texto distinto en el campo ‘U06Text’.
2. En Wait On Elements podemos elegir el tiempo que queremos que el test espere entre acción y acción.
3. En el panel Step Builder se pueden elegir opciones adicionales como por ejemplo hacer el test en loop (bucle for) tantas veces como queramos o hacer que se salte un paso si se da una condición determinada con la opción if…else.

## 5.2 Documentación de construcción

*Paso por paso explicar como ha hecho rober el prototipo anal*

Hay que incluir una descripción de la construcción del prototipo, incluyendo algún extracto de código fuente. No es necesario todo el código. Sólo algún extracto para ver cómo se ha comentado.

## 5.3 Documentación de pruebas

Se han realizado casos de prueba distintos en base a tres modificaciones sobre el caso inicial: el navegador sobre el que se realiza el test, la presencia o no de cookies de la web a probar almacenadas en el navegador antes del test y la modificación de los campos introducidos en el test. Los resultados han sido los siguientes.

* El caso inicial funciona perfectamente en el navegador Internet Explorer.
* Al modificar los campos introducidos (en nuestro caso hemos cambiado la dirección de correo electrónico y el nombre) encontramos que funciona sin problema. Sin embargo, rechaza al encontrarse con que los campos ‘Correo electrónico’ y ‘Repita correo electrónico’ son distintos, cosa que en ningún caso es un error.
* Al realizar el test sobre otros navegadores (en concreto Chrome y Mozilla Firefox) Telerik Test Studio no es capaz de arrancar. Pensamos que esto se debe a que el test inicial se realizó sobre Internet Explorer.
* También nos encontramos problemas a la hora de realizar el mismo test dos veces sobre el mismo navegador. Esto es porque el navegador ya ha guardado en forma de cookies los datos introducidos anteriormente. Para suprimir este obstáculo se ha realizado un test más largo en el que se graba la solución al problema de las cookies y en el que únicamente hay que desactivar tres acciones en caso de que se realice el test por primera vez.

## 5.4 Documentación de instalación

*Explicar paso por paso como se instala Telerik anal studio test anal anal anakllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllll vaginas.*

*Para hacerlo con Facebook mirar 5.2*

Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.

## 5.5 Manual de usuario

*Demostrar que el prototipo cumple los requisitos.*

Descripción suficiente para que una persona que no ha participado en el proyecto pueda utilizar toda la funcionalidad que ofrece el prototipo. Que debe coincidir con los requisitos funcionales incluidos en el apartado 2.

# 6. Comparación de las dos implementaciones

Se trata de dar valores a los criterios de comparación definidos en el apartado 3 sobre la implementación de cada uno de los prototipos.

## 6.1 Evaluación de los criterios en la implementación usando la tecnología A

Debe incluir al menos una tabla con la siguiente estructura.

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Criterio 1 |  |
| Criterio 2 |  |
| … |  |
| Criterio N |  |

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

## 6.2 Evaluación de los criterios en la implementación usando la tecnología B

# 7. Comparación de la implementación 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

|  |  |  |  |
| --- | --- | --- | --- |
| **CRITERIOS** | **TECNOLOGÍA Selenium** | **TECNOLOGÍA Telerik** | **COMENTARIOS** |
| 1: Tiempo desarrollo | 35 horas | 28 horas | El tiempo de desarrollo de Selenium es más elevado debido a que se realiza mediante programación. |
| 2: Recursos necesarios | * Librería Selenium para Python * Browsers y drivers   + Chromedriver para Chrome   + Firefox Geckodriver para Mozilla   + Microsoft Edge driver para Internet Explorer   + Safari driver para Safari * Compiladores Python   + Sublime text 3 3 trial version   + Pycharm student version   + Python oficial * Requisitos de Hardware: dependerán del compilador que se elija | * Telerik Test Studio Free Trial * Browser (Internet Explorer, Chrome, Safari, etc.) * Entornos:   + .Net Framework   + Visual estudio 2012 o superior   + Sharepoint   + C# / VB.NET * Requisitos mínimos de hardware/SO:   + Windows 10, 8, 7 o Vista   + Windows server 2012/2008   + IIS 5+ | Hardware y software necesario para realizar el test. |
| 3: Dificultad desarrollo | Alta | Media | La dificultad del desarrollo mediante Selenium es más elevado debido a que se realiza mediante programación. |
| 4: Tiempo de uso | 7 segundos | 9 segundos | Tiempo que tarda el software en hacer un test completo |
| 5: Facilidad de uso | Media | Baja | Facilidad para llevar a cabo el test una vez están los prototipos preparados |
| 6: Número de iteraciones | A elegir. | A elegir. | Número de veces que se lleva a cabo el test. |
| 9: Forma de mostrar el proceso | Gráfica, muestra la web. | Gráfica, muestra la web. | Forma en la que el prototipo muestra los resultados. |
| 10: Problemas durante el desarrollo | * Dificultad con el código a la hora de hacer lo que se espera de él * Problemas de conexión inicial al navegador con Chrome driver | * Cookies en los navegadores * Problemas de compatibilidad al realizar un test creado en un navegador en otro | Dificultades encontradas durante el desarrollo del protorio. |
| 11: Experiencia requerida | Alta | Media | Es necesario disponer de mayor experiencia para realizar la implementación con Selenium debido a que es necesario saber programar en Python. |

# 8. Conclusiones

El functional testing es completamente imprescindible para el lanzamiento de sitios web. No se puede concebir una página web en la que la mitad de los botones no funcionan, hay campos que no se pueden rellenar o permiten al usuario rellenarlos con información incorrecta o con enlaces rotos que llevan a donde no corresponde.

Es por esto que es fundamental comprobar que la web funciona tal y como se espera de ella antes de lanzarla al usuario y una forma de hacerlo es mediante las herramientas expuestas en este documento. Telerik Test Studio y Selenium ofrecen formas muy distintas de hacer estas comprobaciones. El primero, mediante una interfaz muy visual y el segundo mediante el uso de Python. Si bien es cierto que Telerik es más intuitivo y requiere menos conocimientos de programación e informática en general, con Selenium podremos afinar mucho más el objetivo de nuestros tests. 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.

El desarrollo del prototipo sigue en la línea de lo expuesto anteriormente. A la hora de instalar los programas necesarios Telerik es mucho más sencillo: un setup .exe que se ejecuta e instala el software. Para utilizar Selenium tendremos que elegir entre distintos compiladores (en nuestro caso Sublime), cada uno con sus ventajas, inconvenientes y dificultades en la instalación. Como era previsible, el desarrollo en Selenium es más pesado pues requiere de conocimientos de codificación y un largo proceso de trial and error hasta conseguir que el prototipo realice la secuencia deseada de acciones. El desarrollo utilizando Telerik es notablemente más sencillo e intuitivo: aunque sea la primera vez que utilizas el programa no hay mayor problema a la hora de indicar lo que quieres que haga y hacer las modificaciones necesarias para que el resultado sea satisfactorio. Sin embargo y como ya se ha mencionado, es bastante más complicado e incluso imposible realizar tests complejos con Telerik, cosa que Selenium si permitiría una vez se tienen los conocimientos necesarios.