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

GRUPO T1

TG3. WEB TESTING TOOLS: FUNCTIONAL TESTING

Desarrollo de tecnologías emergentes

Contenido

[1. Autores del trabajo, planificación y entrega 3](#_Toc482030179)

[1.1 Autores 3](#_Toc482030180)

[1.2 Planificación 4](#_Toc482030181)

[1.3 Entrega 5](#_Toc482030182)

[2. Requisitos del prototipo a implementar 6](#_Toc482030183)

[2.1 Requisitos funcionales 6](#_Toc482030184)

[2.2 Otros requisitos 6](#_Toc482030185)

[3. Criterios de comparación en la implementación 7](#_Toc482030186)

[3.1 Criterio 1: Tiempo desarrollo 7](#_Toc482030187)

[3.2 Criterio 2: Recursos necesarios 7](#_Toc482030188)

[3.3 Criterio 3: Dificultad desarrollo 7](#_Toc482030189)

[3.4 Criterio 4: Tiempo de uso 7](#_Toc482030190)

[3.5 Criterio 5: Facilidad de uso 7](#_Toc482030191)

[3.6 Criterio 6: Número de iteraciones 7](#_Toc482030192)

[3.7 Criterio 7: Forma de mostrar el proceso 7](#_Toc482030193)

[3.8 Criterio 8: Problemas durante el desarrollo 7](#_Toc482030194)

[3.9 Criterio 9: Experiencia requerida 7](#_Toc482030195)

[4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología SELENIUM 8](#_Toc482030196)

[4.1 Documentación de diseño 8](#_Toc482030197)

[4.2 Documentación de construcción 8](#_Toc482030198)

[4.3 Documentación de pruebas 9](#_Toc482030199)

[4.4 Documentación de instalación 10](#_Toc482030200)

[4.5 Manual de usuario 10](#_Toc482030201)

[5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología TELERIK TEST STUDIO 11](#_Toc482030202)

[5.1 Documentación de diseño 11](#_Toc482030203)

[5.2 Documentación de construcción 14](#_Toc482030204)

[5.3 Documentación de pruebas 20](#_Toc482030205)

[5.4 Documentación de instalación 21](#_Toc482030206)

[5.5 Manual de usuario 24](#_Toc482030207)

[6. Comparación de las dos implementaciones 25](#_Toc482030208)

[6.1 Evaluación de los criterios en la implementación usando la tecnología SELENIUM 25](#_Toc482030209)

[6.2 Evaluación de los criterios en la implementación usando la tecnología TELERIK TEST STUDIO 25](#_Toc482030210)

[7. Comparación de la implementación de las tecnologías 26](#_Toc482030211)

[8. Conclusiones 29](#_Toc482030212)

# 1. Autores del trabajo, planificación y entrega

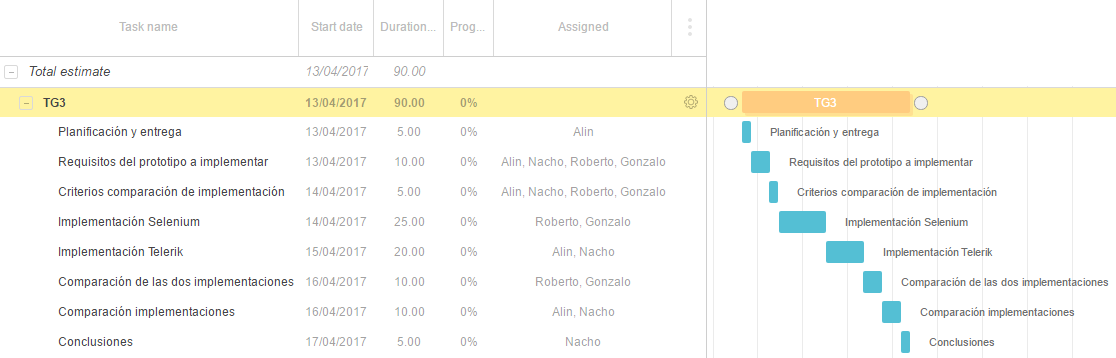
## 1.1 Autores

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/TG3>

# 2. Requisitos del prototipo a implementar

## 2.1 Requisitos funcionales

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

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| RF01 | El sistema deberá poder acceder a la página de registro de Facebook. |
| RF02 | El sistema deberá poder conectar con los navegadores web. |
| RF03 | El sistema deberá comprobar la pantalla de registro de Facebook. |
| RF04 | El sistema deberá introducir todos los datos que se pide en el proceso de registro. |
| RF05 | El sistema deberá introducir parámetros aleatorios para comprobar que no se puede introducir datos erróneos en los campos de registro. |
| RF06 | El sistema deberá comprobar los mensajes de la página al introducir datos erróneos en los campos del registro. |
| RF07 | El sistema deberá intentar saturar el registro de Facebook. |
| RF08 | El sistema no debería abortar, aunque el registro sea incorrecto. |
| RF09 | El sistema debe mostrar gráficamente como se lleva a cabo el proceso. |

RF: REQUISITO FUNCIONAL.

## 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** |
| --- | --- |
| RNFS01 | El sistema debe ser inalterable por agentes externos. |
| RNFR02 | El sistema debe tardar menos de 10 segundos en llevar a cabo el registro. |

RNFS: REQUISITO NO FUNCIONAL DE SEGURIDAD

RNFR: REQUISITO NO FUNCIONAL DE RENDIMIENTO

# 3. Criterios de comparación en la implementación

## 3.1 Criterio 1: Tiempo desarrollo

Descripción: Horas invertidas en la implementación de los sistemas para la creación de los test.

Tipo de valor: Numérico (horas).

## 3.2 Criterio 2: Recursos necesarios

Descripción: El material y las herramientas necesarias para la creación de los sistemas.

Tipo de valor: texto libre.

## 3.3 Criterio 3: Dificultad desarrollo

Descripción: Grado de dificultad para la creación de los sistemas.

Tipo de valor: texto libre.

## 3.4 Criterio 4: Tiempo de uso

Descripción: Tiempo que se tarda en la utilización de los sistemas.

Tipo de valor: Numérico (segundos).

## 3.5 Criterio 5: Facilidad de uso

Descripción: experiencia requerida por el usuario para la correcta manipulación del programa.

Tipo: texto libre.

## 3.6 Criterio 6: Número de iteraciones

Descripción: número de veces que es capaz de hacer el registro sin que Facebook lo aborte.

Tipo: Numérico.

## 3.7 Criterio 7: Forma de mostrar el proceso

Descripción: Forma de mostrar gráficamente como se está realizando el proceso.

Tipo: texto libre.

## 3.8 Criterio 8: Problemas durante el desarrollo

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

Tipo de valor: Numérico.

## 3.9 Criterio 9: Experiencia requerida

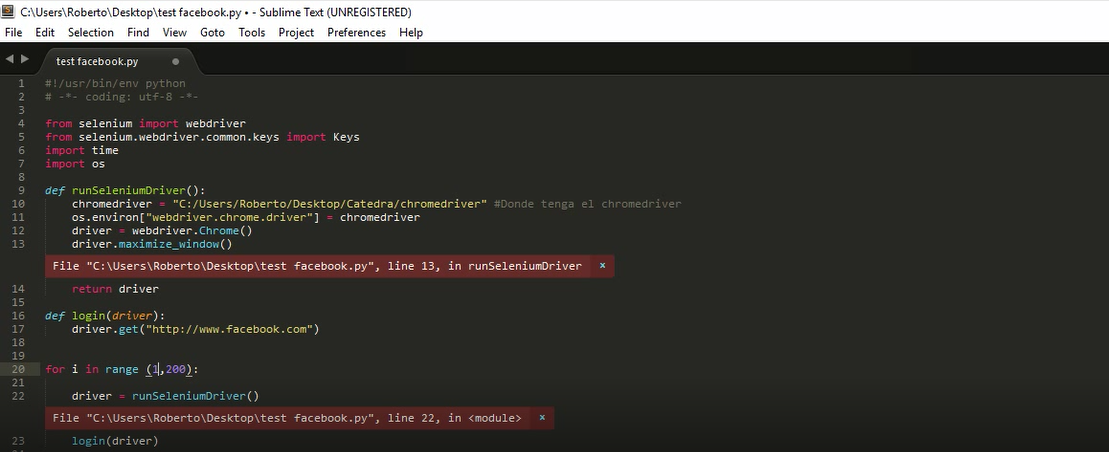
Descripción: cantidad de experiencia requerida para realizar el desarrollo correctamente.

Tipo de valor: texto.

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

## 4.1 Documentación de diseño

La interfaz de Selenium tiene el siguiente aspecto:



La interfaz va a depender del entorno de desarrollo que utilicemos, en este caso resulta bastante menos intuitiva que la de Telerik Test Studio, puesto que es realmente un entorno de programación y depende todo de lo que el usuario incluya como código fuente.

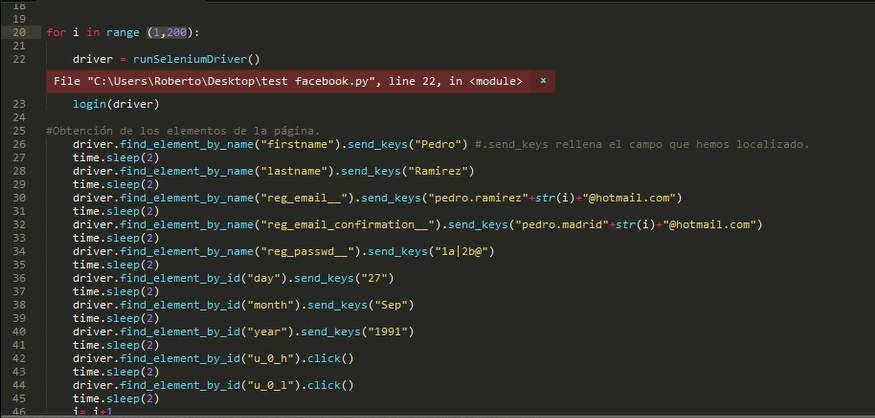
Aun así, nos ofrece una barra de herramientas en la parte superior donde podremos abrir proyectos ya existentes, guardar el proyecto actual o crear uno nuevo. También nos ofrece opciones de edición, vista, diferentes herramientas, preferencias, herramientas de proyecto y una ventana de ayuda.

## 4.2 Documentación de construcción

A continuación, se muestra el código fuente incluido en entorno de desarrollo con la librería Selenium, y una breve explicación de cada uno de los métodos:



* De la línea 4 a la 7 se encuentran los diferentes importes necesarios para el funcionamiento del proyecto, es decir, los drivers.
* De la línea 9 a la 13, la función runSeleniumDriver() se encarga de elegir el driver correspondiente del directorio donde se encuentre en nuestro equipo y almacenarlo en la variable driver, que es devuelta al finalizar el método.
* De la línea 16 a la 17, en la función login(driver) elegimos el sitio web sobre el que vamos a realizar la prueba, en este caso Facebook.



* De la línea 20 a la 46, el bucle for se encarga de hacer las llamadas a los métodos mencionados anteriormente para comenzar a pasar valores que se utilizarán durante la prueba.

Como es el caso de Facebook, se introducen valores como nombre, apellido, correo electrónico, contraseña, y fecha de nacimiento. Estos valores se repetirán el número de iteraciones indicadas en el bucle for, y se irán repitiendo los registros en Facebook con ellos.

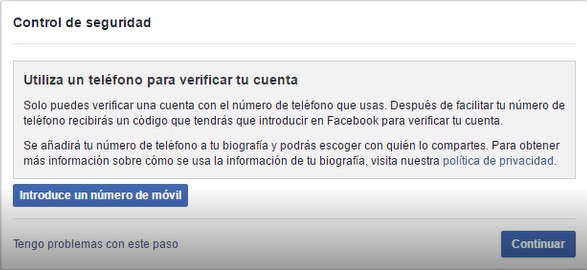
## 4.3 Documentación de pruebas

En un principio probamos la ejecución de la prueba con el bucle for (1,200), por lo que al ejecutarlo de nuevo se repetían los correos, asi que cambiamos el número de iteraciones a (5,200).

También en la anterior prueba cambiamos el correo de confirmación, por lo que al realizar el registro no salta el error por parte de Facebook, como se observa en la imagen siguiente:



Por último, con todos los valores correctos se produce el registro y Facebook nos reclama número de móvil para verificar la cuenta:



## 4.4 Documentación de instalación

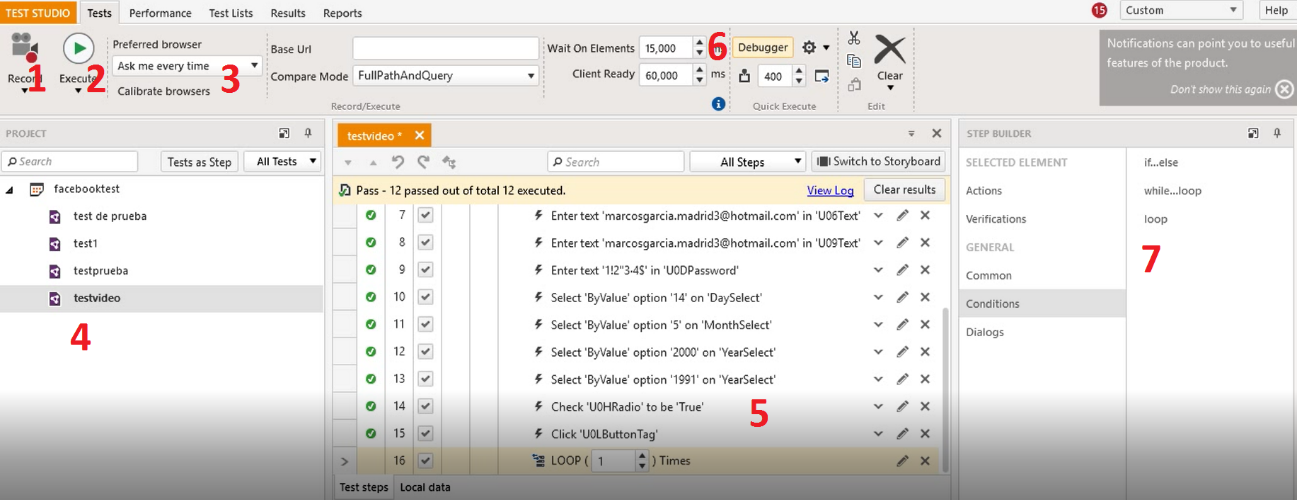
## 4.5 Manual de usuario

Para poder llevar a cabo la ejecución de este prototipo es necesario tener conocimientos previos de Python, partiendo de ahí en los apartados ***4.1 Documentación de diseño*** y ***4.2 Documentación de construcción*** se explica toda la información necesaria para la implementación de este prototipo.

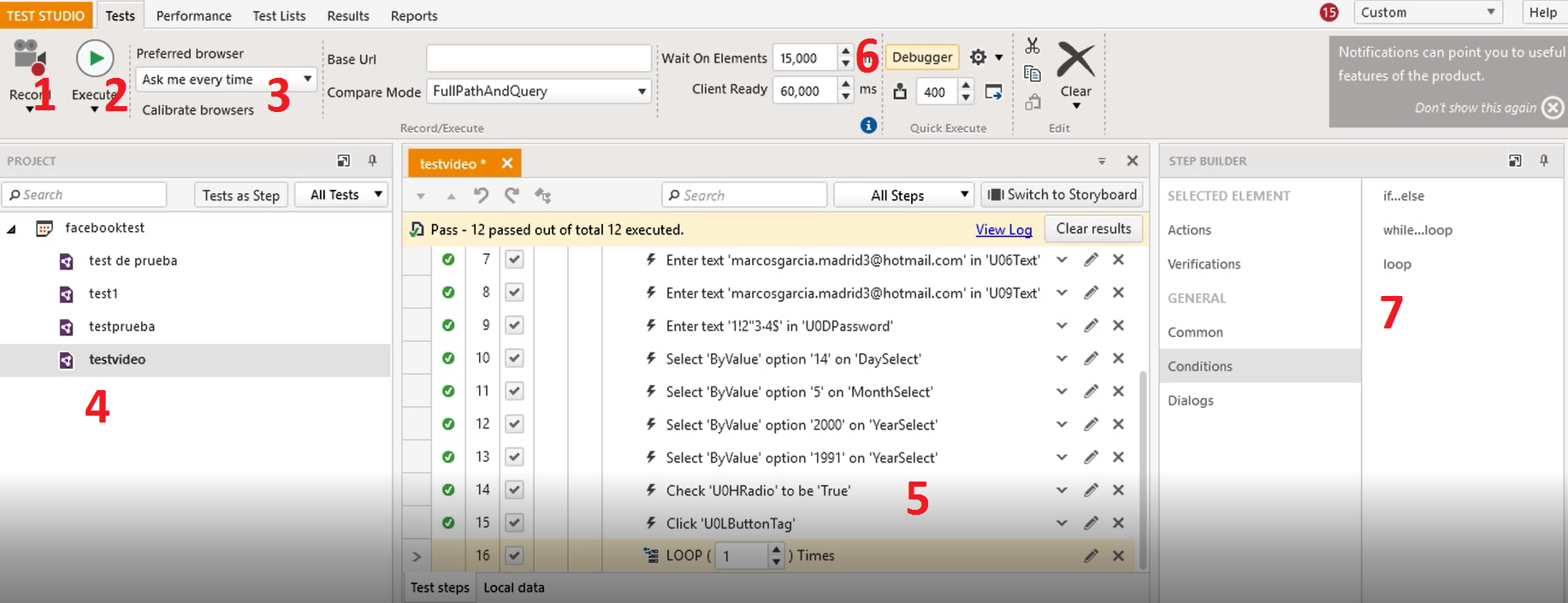
# 5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología TELERIK TEST STUDIO

## 5.1 Documentación de diseño

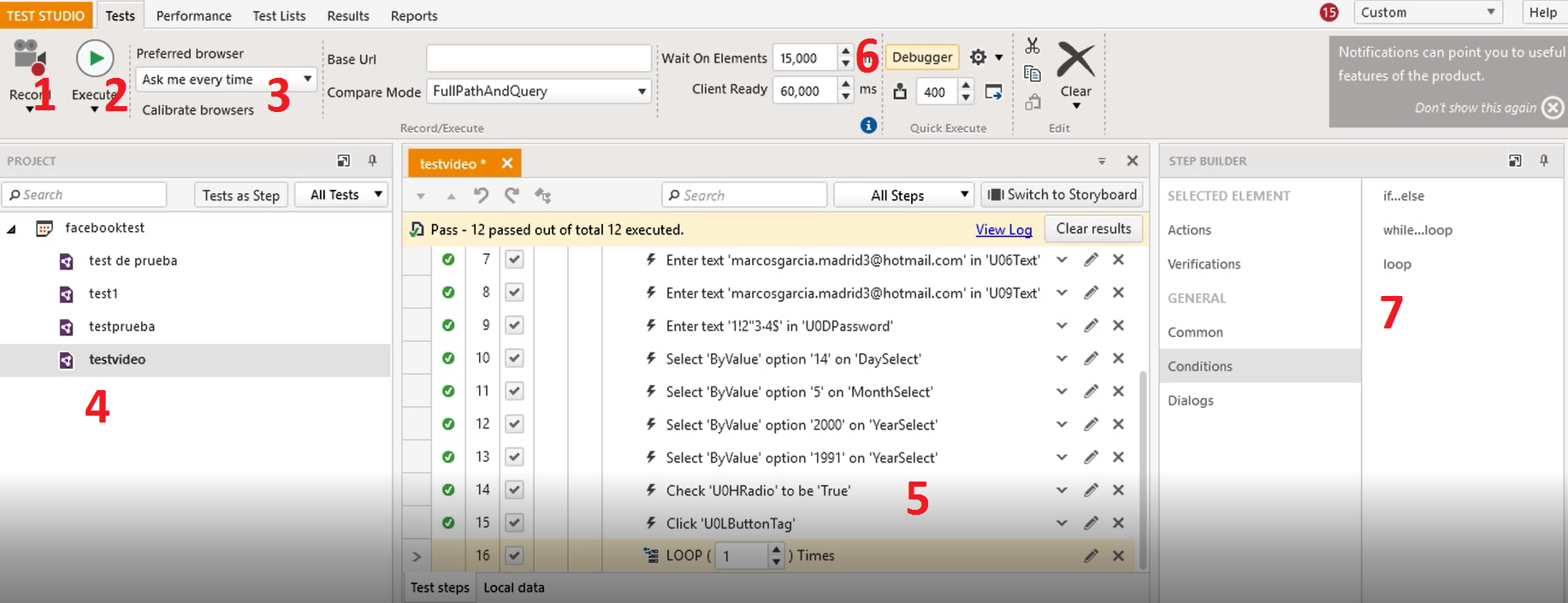
La interfaz principal de Telerik tiene el siguiente diseño:



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.

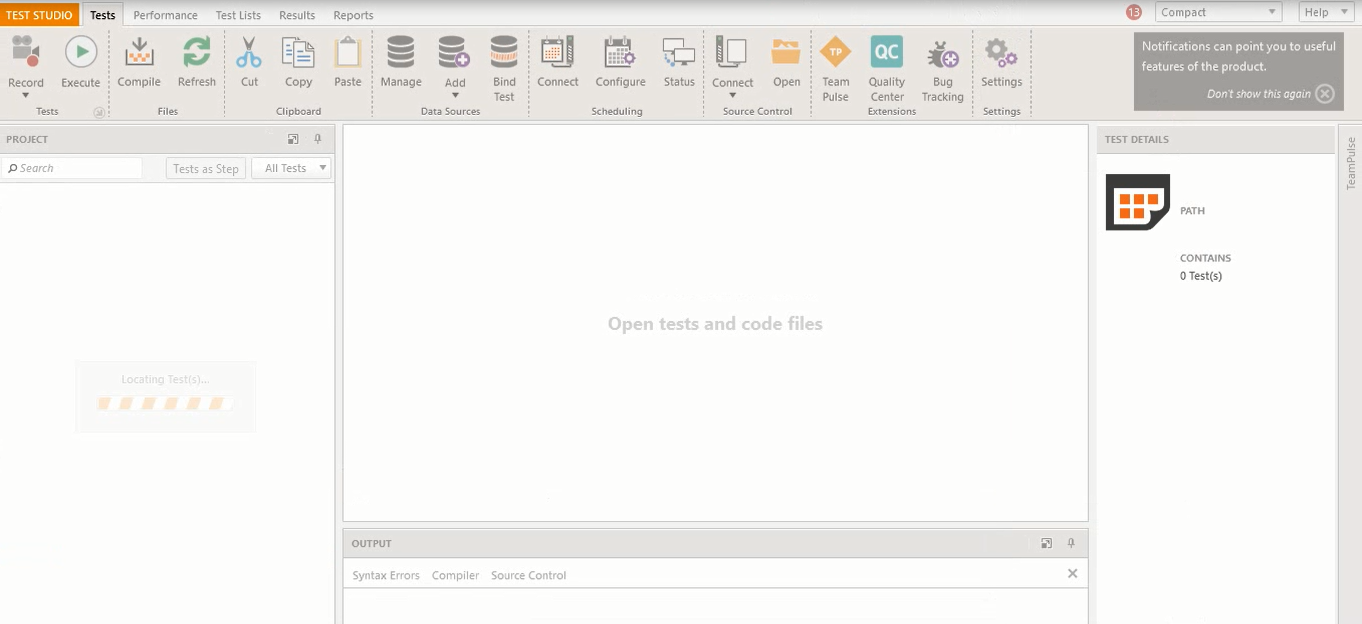


1. Es el panel principal de Telerik Test Studio. En él, se muestran todas las acciones que han sido grabadas. Por ejemplo, en este caso, se puede ver como la séptima acción ha sido introducir el texto [marcosgarcia.madrid3@hotmail.com](mailto:marcosgarcia.madrid3@hotmail.com) en el campo de nombre ‘U06Text’ 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.

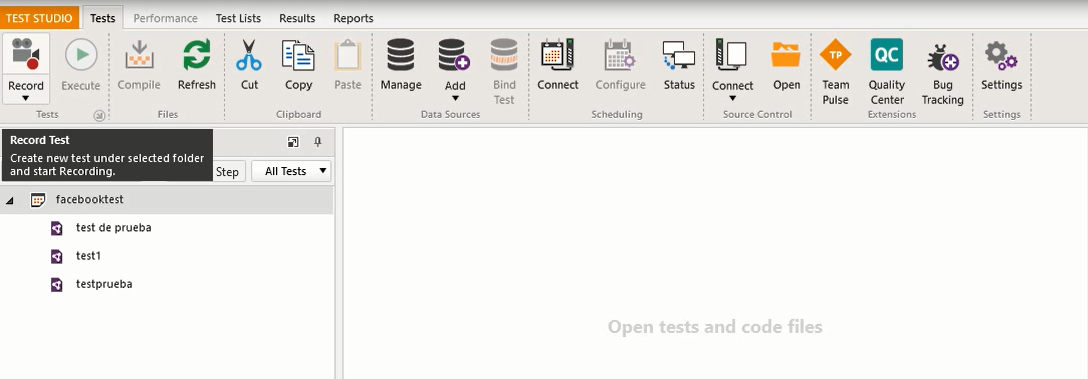
## 5.2 Documentación de construcción

Los pasos para construir el prototipo mediante la tecnología TELERIK TEST STUDIO son los siguientes:

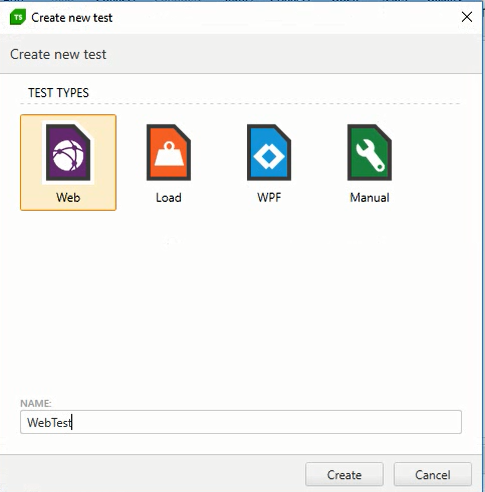
1. Acceder a la web de Telerik (<http://www.telerik.com/>) y descargar la aplicación.
2. Una vez descargada, tendremos que instalarla. *Esto se detallará en el punto 5.4 Documentación de instalación.*
3. Cuando accedemos a la aplicación nos aparecerá la siguiente interfaz:



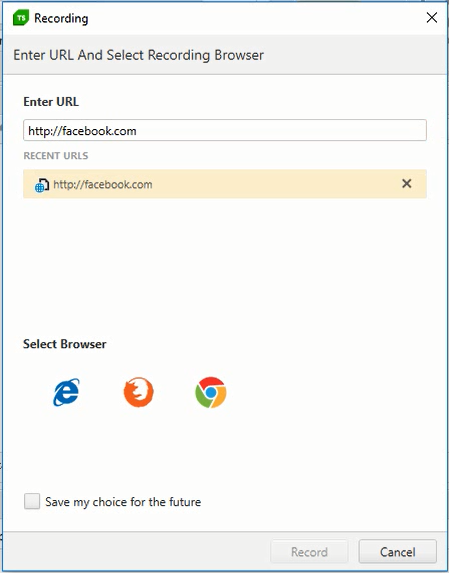
1. Para crear un nuevo test tenemos que hacer click en el botón Record.



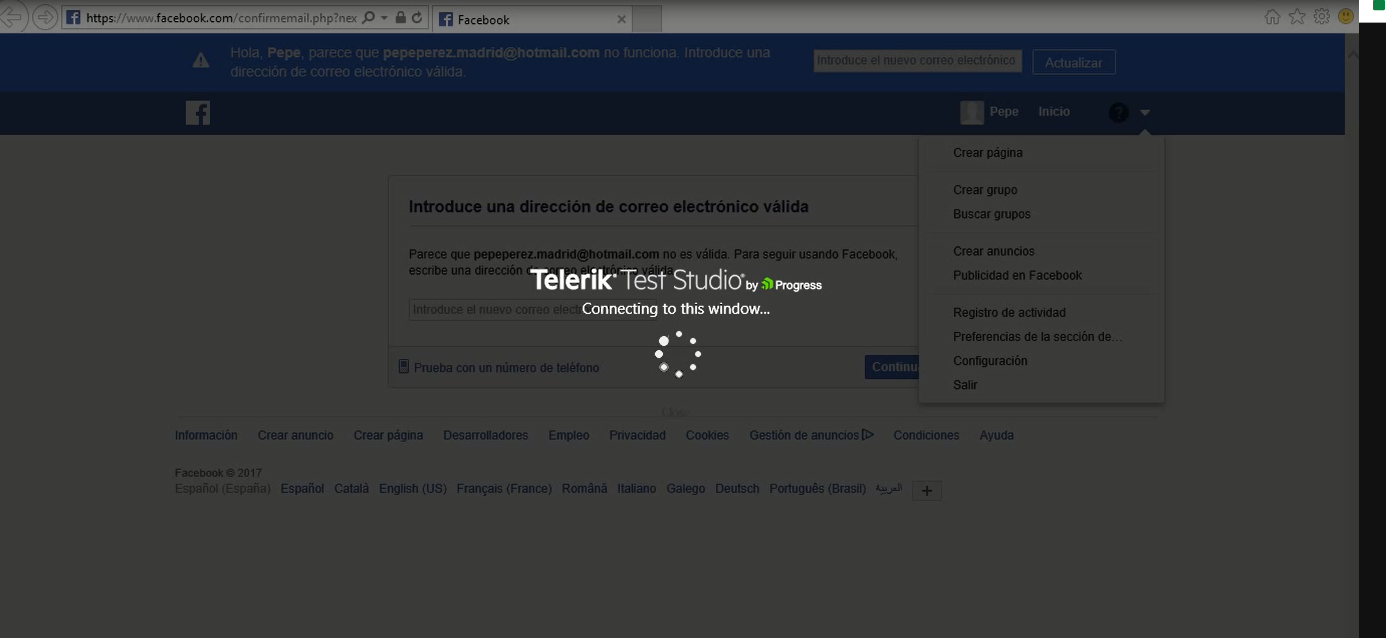
1. Hacemos click en la pestaña de Web, le damos un nombre al test y pinchamos en Create:



1. Introducimos la URL de la web donde queremos realizar el test y seleccionamos el navegador que se va a utilizar para realizar el mismo:



1. Ahora podemos observar cómo se está conectando el navegador elegido (Internet Explorer) con Telerik Test Studio.



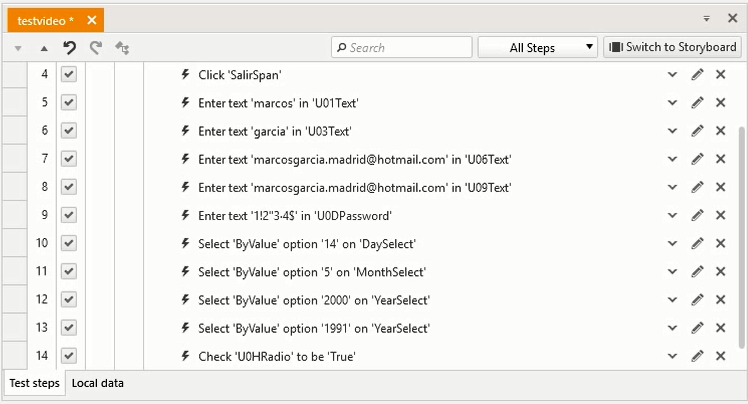
1. Una vez se ha realizado la conexión correctamente entrará en la web que queramos analizar, en este caso, Facebook:



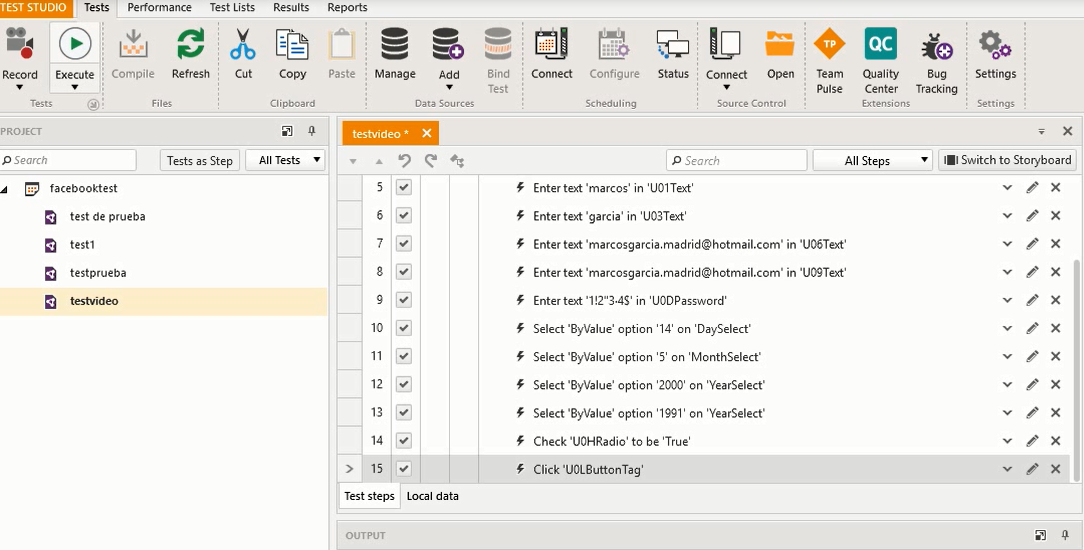
1. Ahora procederemos a introducir la información en los campos que nos piden y haremos click en registrarte. Hay que recordar que Telerik está constantemente grabando los pasos que realiza el usuario para luego reproducirlo automáticamente.



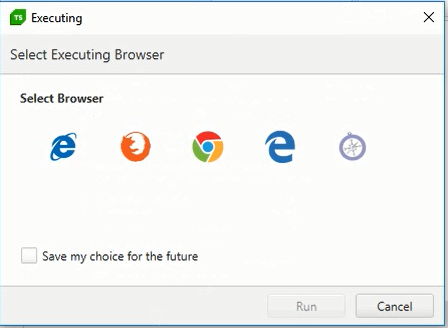
1. Una vez que hemos introducido todos los pasos que queremos reproducir posteriormente, volvemos a la aplicación de Telerik y podemos observar que se han guardado cada una de las interacciones que hemos realizado anteriormente.



1. A continuación, vamos a proceder a reproducir los pasos que hemos grabado anteriormente. Para ello, seleccionamos el test que queremos reproducir y haremos click en Execute:



1. Elegimos el navegador donde vamos a reproducir el test:



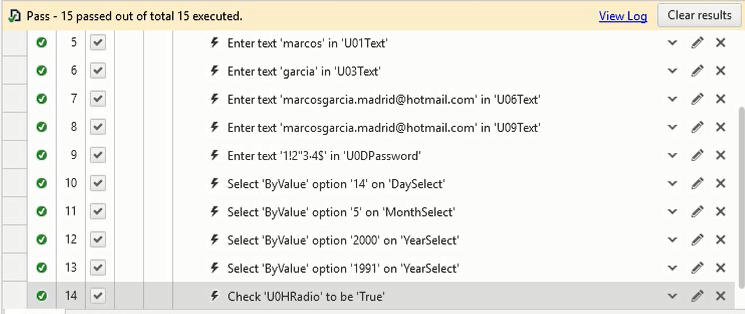
1. Esperaremos unos segundos a que se nos conecte automáticamente al navegador:



1. Podemos observar cómo se realizan los mismos movimientos que hemos grabado antes de forma automática:



1. Si volvemos a la interfaz de Telerik podemos ver que se han realizado todos los movimientos que hemos creado anteriormente de manera satisfactoria:



## 5.3 Documentación de pruebas

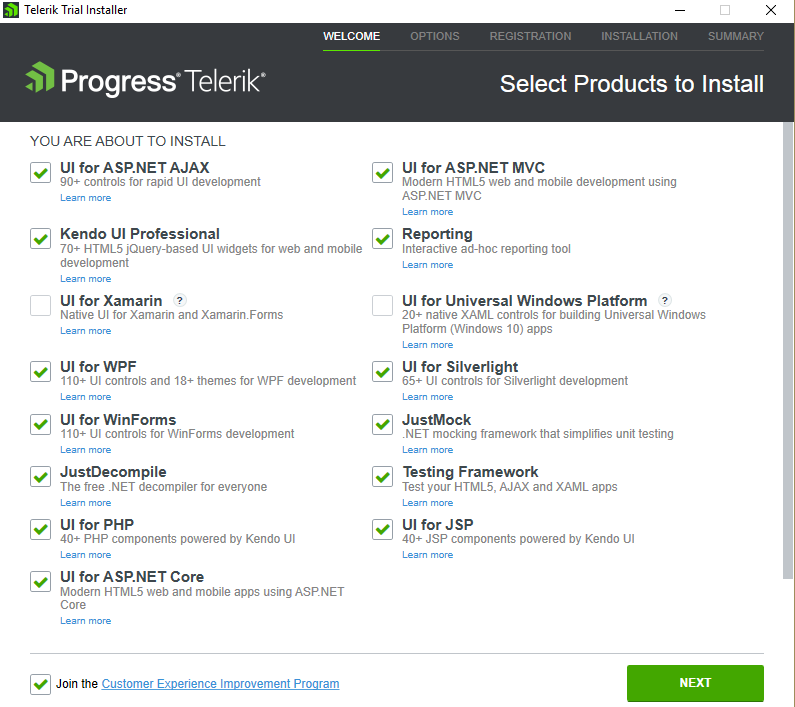
Se han realizado casos de prueba distintos en base a tres modificaciones sobre el caso inicial: el navegador sobre el que se realiza el test, la presencia o no de cookies de la web a probar almacenadas en el navegador antes del test y la modificación de los campos introducidos en el test. Los resultados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

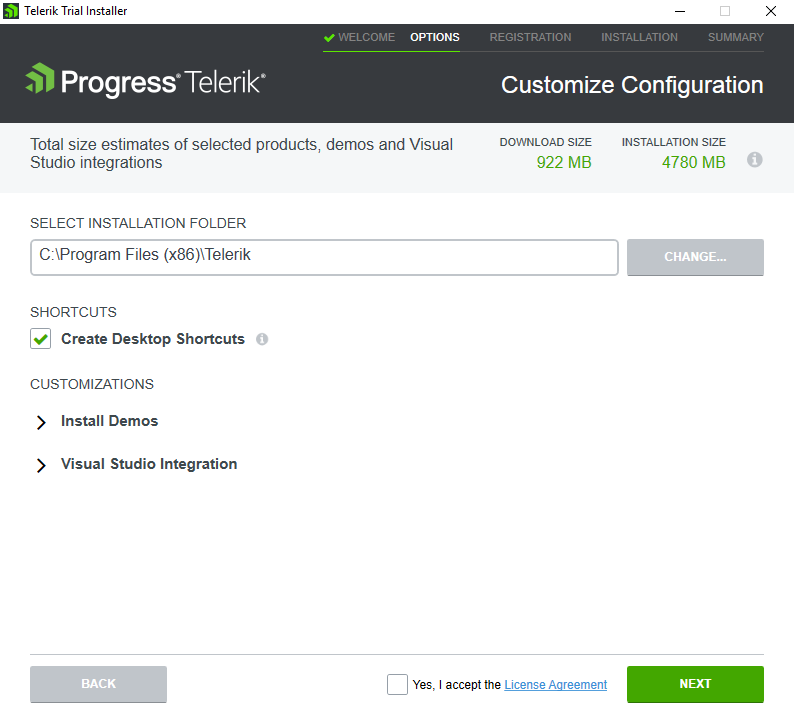
## 5.4 Documentación de instalación

Para poder realizar pruebas con TELERIK TEST STUDIO, simplemente nos tendremos que descargar la aplicación e instalarla en nuestro dispositivo:

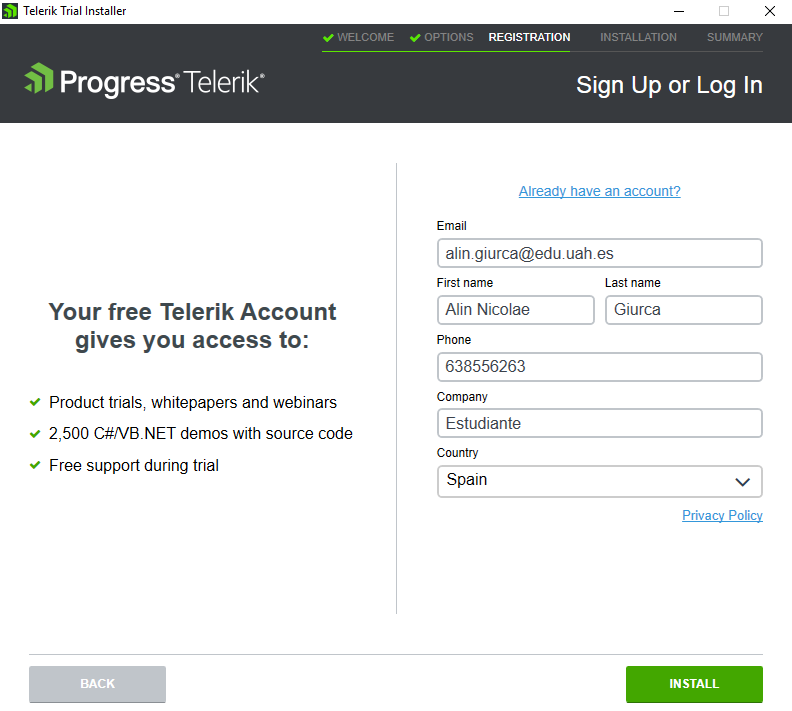
1. Acceder a la web de Telerik (<http://www.telerik.com/>) y descargar la aplicación.
2. Ejecutar el archivo de instalación.
3. Instalación:
   1. Una vez que hemos ejecutado el archivo de instalación nos aparecerá la siguiente pantalla, donde elegiremos los productos que queremos instalar y haremos click en NEXT:



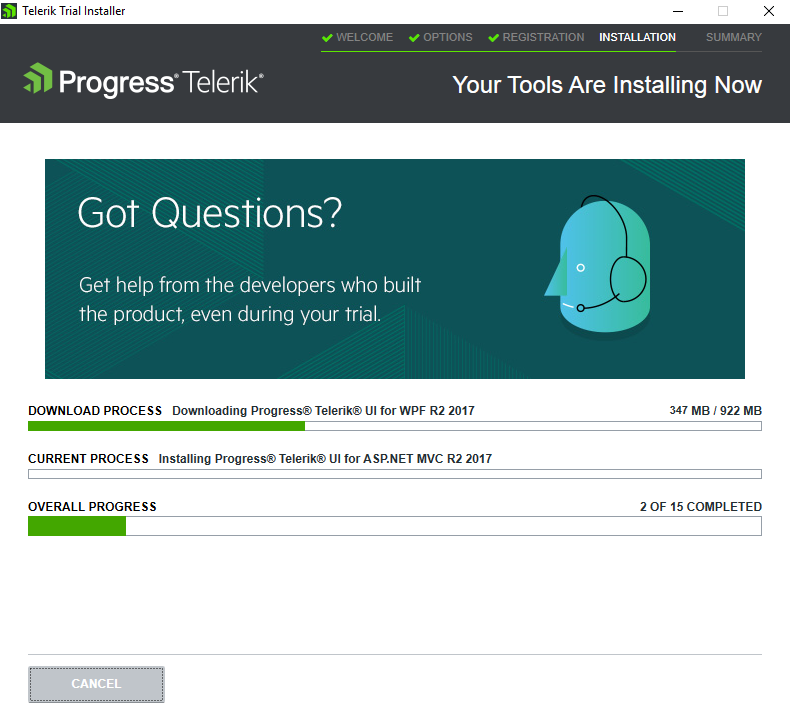
* 1. Seleccionamos la carpeta de nuestro dispositivo donde se va a instalar TELERIK y podemos customizar nuestra instalación añadiendo paquetes o extensiones a la misma. Una vez que hemos seleccionado los parámetros de instalación y aceptado las condiciones de uso hacemos click en NEXT:



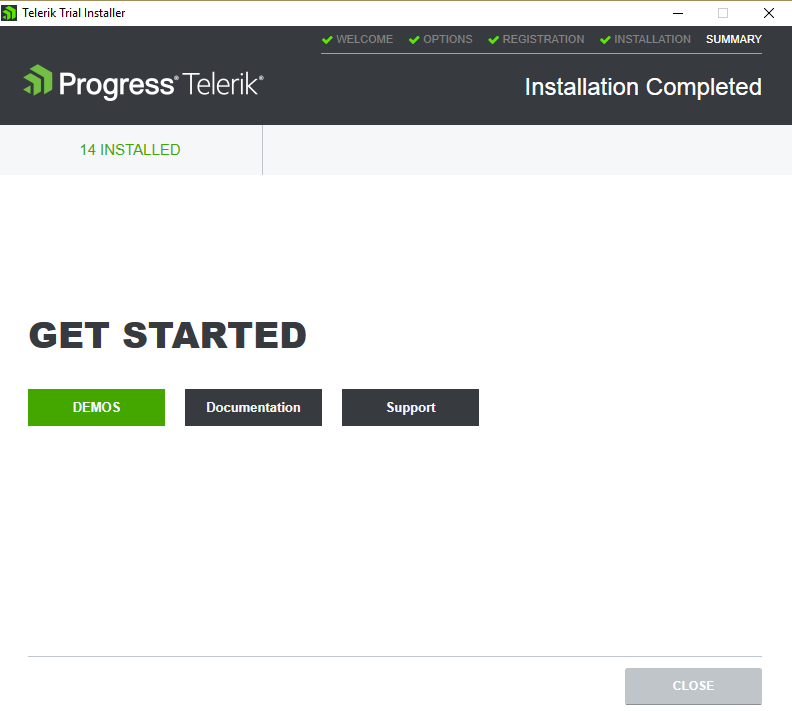
* 1. Lo siguiente que tenemos que hacer es iniciar sesión si tenemos una cuenta, y si no, tendremos que introducir nuestros datos y darnos de alta en el sistema:



* 1. Finalmente, tenemos que esperar a que finalice el proceso de instalación de TELERIK:



* 1. Una vez que ha finalizado podemos comenzar a realizar el test que deseemos. Para realizar los test tenemos que seguir los pasos descritos en el punto *5.2 Documentación de construcción.*



## 5.5 Manual de usuario

Para que cualquier persona pueda realizar un Test de Prueba de cualquier web mediante la herramienta TELERIK TEST STUDIO, simplemente tendrá que instalar la aplicación siguiendo los pasos indicados en el punto ***5.4 Documentación de instalación*** y para poder realizar el Test de Prueba deberá seguir los pasos del punto ***5.2 Documentación de construcción.***

# 6. Comparación de las dos implementaciones

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

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Tiempo de desarrollo | 35 horas |
| Recursos necesarios | Librería Selenium para Python, Browsers y drivers, Compiladores Python y Requisitos de Hardware. |
| Dificultad de desarrollo | Alto |
| Tiempo de uso | 10 segundos |
| Facilidad de uso | Media |
| Número de iteraciones | A elegir. |
| Tiempo de verificación | Medio |
| Forma de mostrar el proceso | Gráfica, muestra la web. |
| Tiempo registro | Bajo |
| 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, Dificultad a la hora de verificar captcha. |
| Experiencia requerida | Alta |

## 6.2 Evaluación de los criterios en la implementación usando la tecnología TELERIK TEST STUDIO

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Tiempo de desarrollo | 28 horas |
| Recursos necesarios | Bajo |
| Dificultad de desarrollo | Medio |
| Tiempo de uso | 5 segundos |
| Facilidad de uso | Baja |
| Número de iteraciones | A elegir. |
| Tiempo de verificación | Medio |
| Forma de mostrar el proceso | Gráfica, muestra la web. |
| Problemas durante el desarrollo | Cookies en los navegadores, Problemas de compatibilidad al realizar un test creado en un navegador en otro, Dificultad a la hora de verificar captcha. |
| Experiencia requerida | Media |

# 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 | 10 segundos | 5 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 * Dificultad a la hora de verificar captcha. | * Cookies en los navegadores * Problemas de compatibilidad al realizar un test creado en un navegador en otro * Dificultad a la hora de verificar captcha. | 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.